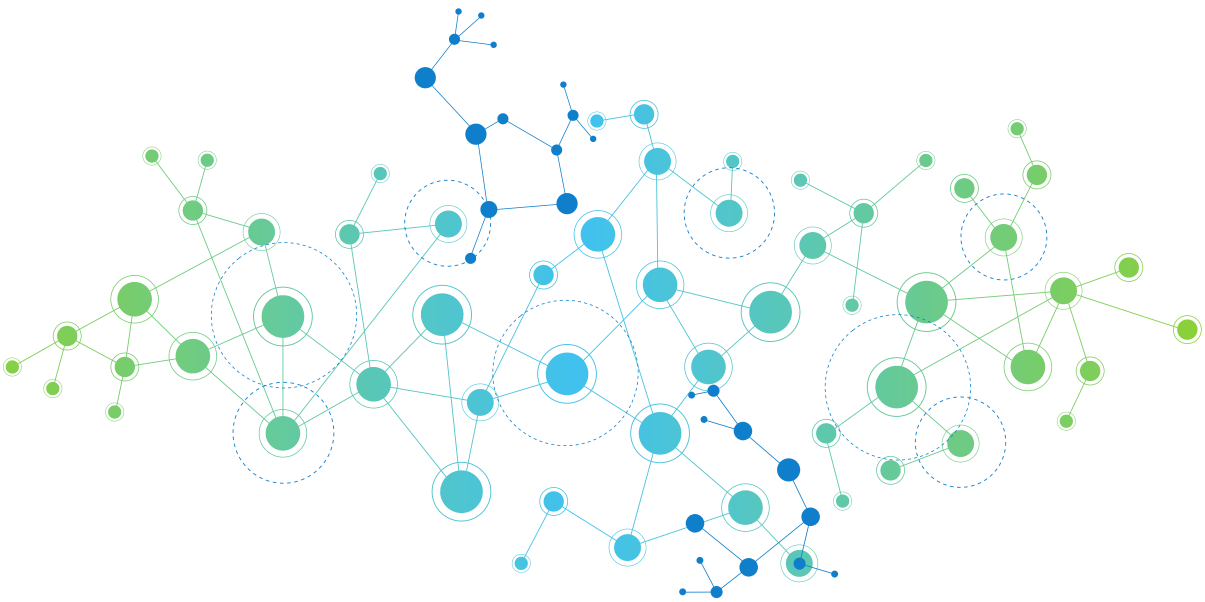
Enabling interoperability between MAC-heterogeneous sensor networks

Proefschrift voorgelegd tot het behalen van de graad van doctor in
de wetenschappen: informatica aan de Universiteit Antwerpen te verdedigen door

Daniel van den Akker



Promotor
Prof. dr. C. Blondia

Faculteit Wetenschappen
Departement Informatica

Antwerpen 2020

Universiteit Antwerpen

Universiteit Antwerpen
Faculteit Wetenschappen
Departement Informatica

Enabling interoperability between MAC-heterogeneous sensor networks

Proefschrift voorgelegd tot het behalen van
de graad van doctor in de Wetenschappen:
Informatica aan de Universiteit Antwerpen
te verdedigen door

Daniel van den Akker

Promotor: Prof. dr. C. Blondia

IDLab
INTERNET & DATA LAB

Antwerpen, 2020

Acknowledgements

This work has been supported by an Aspirant Grant of the from the Fund for Scientific Research Flanders (FWO) and part of the research leading to these results has also received funding from the Agency for Innovation by Science and Technology (IWT) as part of the Symbionets project.

While working on this PhD, I have received help and guidance from many, so before moving on to more technical matters I would first like to extend a word of gratitude to all of those who helped make it possible.

First of all, I would like to thank my parents Jenny and Arie and my sister Suzanne for all that they have done for me over the years. I could always count on their support and advice, as well as on their limitless patience and understanding during the, more numerous that I would have liked, bouts of PhD-induced grumpiness. In addition, I consider myself lucky to have been able to benefit from the many insights into both the research and non-research aspects of working in an academic environment that Suzanne was able to provide and I am thankful for her valiant efforts to get that stubborn brother of hers away from his keyboard and out of the house once in a while. Likewise, I am grateful to my parents for keeping track of, and on many occasions taking over, the everyday tasks and other projects that have so often been pushed to the background while I was otherwise engaged. Without their assistance my academic progress, as well as a certain other non-academic project, would never have come anywhere as close as to where they are today.

In addition I would also like to thank

> my family and friends for their continued support and encouragement over the years. The appreciation and interest they all regularly showed was a great help in keeping me focussed on the task at hand.
>
> the many wonderful colleagues, both current and former, that I have had the fortune to work with over the years. Apart from doing their absolute best to create a pleasant and supportive work environment, they have always been willing to assist

me in the technical and non-technical aspects of both my PhD and of the various research projects that I have been involved in.

the honourable members of the jury: Prof. Dr. Hans Vangheluwe, Prof. Dr. Sofie Pollin, Prof. Dr. Ingrid Moerman, Dr. Kathleen Spaey and Prof. Dr. Steven Latré, for their helpful feedback, for making time to help me finalise my PhD and for making sure that the remainder of this document makes sense to a wider audience than just the guy who wrote it. I would also like to thank Dr. Kathleen Spaey for going through this thesis with a fine-tooth comb in her spare time to help me fix the many 'little mistakes' that would otherwise have gone unnoticed until *after* the book had been printed.

Last but not least, I would like to thank my promotor Prof. Dr. Chris Blondia who not only provided me with the opportunity to start this epic journey and advised me every step of the way, but who also, when I has just about thrown in the towel, told me to pick it up again and keep on going.

Abstract

Sensor networks are wireless networks composed of small, cheap, battery powered nodes that are deployed either inside or close by a phenomenon of interest. These nodes are equipped with one or more sensors and communicate with each other through a wireless communication network to provide a combined view of the area in which the sensors are deployed. Today sensor networks are used for a wide range of applications ranging from environmental monitoring (e.g., monitoring air and water quality) to industrial process control and building automation. One key aspect of these sensor networks is that, in order to keep both cost and energy consumption to a minimum, the nodes in these networks tend to be quite resource constrained. Over the years, this has caused sensor network developers to sacrifice interoperability for increased performance and/or longevity by optimising not only the application-code, but also the protocols used to organise communication between the different nodes, to the specific requirements of the use case for which the sensor network is being deployed.

Over the last decade however, there has been a substantial paradigm-shift, not only in the way we look at sensor networks but also in the way we look at electronic devices in general. Under the title 'Internet-of-Things' (IoT), there has been a continuous push to make not only the nodes of a sensor network but also everyday devices and physical objects accessible through the internet. While the incorporation of sensor networks into the broader family of 'Internet-of-Things' networks has not altered any of the underlying principles behind these networks, it does mean that today interconnectivity and interoperability are more important than ever in the design of sensor networks.

This thesis therefore considers on the problem of enabling link-level interoperability between sensor networks using different (incompatible) MAC protocols. The reason for focussing on the MAC layer is that, due to its impact on the overall energy consumption of the node, this layer has been the most extensively customised by sensor network developers (and researchers) and therefore presents one of the largest obstacles to enabling inter-network connectivity.

To do so, this thesis first investigates how the presence of multiple MAC-heterogeneous sensor networks in a single wireless environment affects the performance of these networks. The reason for doing so is that there is little point in attempting to enable interoperability between these networks if the interference that exists between them prevents them from (feasibly) co-existing with one another. The results of this performance-study show that although in extreme circumstances (high traffic load, large number of nodes) the performance can be significantly affected by inter-MAC interference, under less strenuous conditions the effect of this interference on the overall network performance is small enough to allow these networks to co-exist in the same wireless environment without much issue.

Next, this thesis turns to the problem of actually enabling communication between such MAC-heterogeneous sensor networks. To do so, it proposes to use so-called *virtual gateways*: regular sensor nodes that run multiple MAC protocols simultaneously on top of a single radio interface. To demonstrate the feasibility of this approach, the software architecture needed to simultaneously run multiple MAC protocols is implemented for the extremely resource constrained Tmote Sky sensor node platform. It is shown that the proposed architecture is flexible and extensible enough to support a wide variety of MAC protocols and that the overhead of the Tmote Sky-implementation is minimal.

The remainder of the thesis focusses on the problem of deciding which sensor nodes to use as a virtual gateway. This is a non-trivial problem given that the optimal selection of virtual gateways not only depends on the route-topology of the networks, which is in turn affected by the specific virtual gateways used, but also on the specific performance-requirements imposed by the administrators of the individual networks. This thesis therefore introduces the *IRVG*-algorithm (Iterative Removal of Virtual Gateways) as a means to automatically select the virtual gateway nodes to use. As the name implies, this algorithm operates by first configuring all sensor nodes as a virtual gateway and then iteratively disabling those virtual gateways that are unnecessary. Through a series of performance tests, this algorithm is shown to be able to select virtual gateways in such a manner that the resulting performance of the networks nears or even exceeds the performance of the ideal 'Same-MAC' scenario. Moreover, this algorithm is also shown to be capable of balancing between the different (and possibly conflicting) performance requirements imposed by the administrators of the individual networks.

Nederlandstalige Samenvatting

Sensornetwerken zijn draadloze netwerken die uit kleine, goedkope nodes bestaan. Deze zijn uitgerust zijn met één of meerdere sensoren en worden meestal door een batterij gevoed. Door (draadloos) met elkaar te communiceren geven ze een gezamenlijk beeld van de omgeving waarin ze zich bevinden. Sensornetwerken worden tegenwoordig ingezet voor een brede waaier aan toepassingen. Typische voorbeelden hiervan zijn onder andere milieubeheer, zoals bijvoorbeeld het meten van lucht- en waterkwaliteit, controle van industriële processen en domotica. Deze netwerken hebben als beperkende eigenschap dat, om de kostprijs te drukken en de levensduur van de batterijen van de sensornodes te verlengen, deze doorgaans slechts een zeer beperkte processor- en geheugencapaciteit hebben. Dit heeft ertoe geleid dat ontwikkelaars, gedurende jaren, ervoor hebben gekozen om, niet alleen de applicatiecode maar ook de protocollen die de communicatie tussen de verschillende nodes in goede banen leiden, zo veel mogelijk af te stemmen op de specifieke vereisten van de applicatie waarvoor het sensornetwerk geïnstalleerd wordt. Dat er hierdoor in het algemeen geen interoperabiliteit tussen verschillende sensornetwerken is, werd hierbij lange tijd als een aanvaardbare trade-off beschouwd.

Gedurende de laatste tien jaar is echter de wijze, waarop we met zowel sensornetwerken als met elektronische apparaten in het algemeen omgaan, sterk veranderd. Onder de noemer 'Internet-of-Things' (IoT), is er een trend ontstaan om niet alleen de nodes van een sensornetwerk maar ook dagdagelijkse apparaten en objecten toegankelijk te maken via het internet. Hoewel de integratie van sensornetwerken in de bredere familie van 'Internet-of-Things'-netwerken de onderliggende principes van deze sensornetwerken niet fundamenteel veranderd heeft, heeft dit er wél voor gezorgd dat vandaag de dag het voorzien van interoperabiliteit een steeds belangrijkere factor is in het ontwerp van sensornetwerken.

Deze thesis heeft daarom tot doel om communicatie op linkniveau mogelijk te maken tussen sensornetwerken die gebruik maken van verschillende (incompatibele) MAC-protocollen. De MAC-laag is de voorbije jaren immers het sterkst onder handen genomen

door sensornetwerkontwikkelaars en vormt daarom één van de grootste obstakels voor interoperabiliteit tussen verschillende sensornetwerken.

Hiervoor wordt in deze thesis eerst onderzocht hoe de performantie van een sensornetwerk beïnvloed wordt door de aanwezigheid van een ander sensornetwerk, dat gebruik maakt van een ander MAC-protocol. Dit wordt eerst onderzocht omdat het weinig zin zou hebben om verschillende MAC-heterogene sensornetwerken met elkaar te laten communiceren als de interferentie, tussen de verschillende MAC-protocollen, communicatie in de afzonderlijke netwerken onmogelijk zou maken. De resultaten van dit onderzoek tonen aan dat in extreme omstandigheden (veel trafiek, aanwezigheid van veel nodes in een kleine ruimte) de desbetreffende netwerken sterke hinder kunnen ondervinden van de interferentie tussen hun respectievelijke MAC-protocollen. Onder minder veeleisende omstandigheden is het effect van deze inter-MAC-interferentie echter klein genoeg om deze netwerken zonder al te veel problemen naast elkaar te laten werken in dezelfde omgeving.

Deze thesis bestudeert vervolgens hoe communicatie mogelijk kan worden gemaakt tussen MAC-heterogene sensornetwerken. Hiervoor wordt gebruik gemaakt van zogenaamde *virtuele gateways*. Dit zijn reguliere sensornodes die tegelijkertijd meerdere MAC-protocollen kunnen gebruiken in combinatie met één enkele radio-interface. Om de haalbaarheid van deze aanpak aan te tonen, werd de software-architectuur die hiervoor nodig is, geïmplementeerd voor het Tmote Sky sensornode-platform. Er wordt aangetoond dat deze architectuur flexibel en uitbreidbaar genoeg is om een breed scala aan MAC-protocollen te ondersteunen. Verder wordt ook aangetoond dat de performantiekost van de Tmote Sky-implementatie van deze architectuur minimaal is.

Verder richt de thesis zich op de vraag welke specifieke sensornodes best gebruikt kunnen worden als virtuele gateway. Dit is een vrij complex probleem vermits, bij de keuze van de verschillende gateways, rekening moet worden gehouden met de routes die in en tussen de verschillende netwerken gebruikt worden en dat deze routes op hun beurt beïnvloed worden door de locatie van de virtuele gateways. Daarnaast moet hierbij ook rekening worden gehouden met de performantievereisten die gelden binnen elk van deze netwerken. Deze thesis introduceert daarom het *IRVG*-algoritme om automatisch de te gebruiken virtuele gateways te selecteren. Om dit te doen worden éérst alle mogelijke virtuele gateways ingeschakeld waarna alle onnodige gateways iteratief worden uitgeschakeld. Door middel van een aantal performantietesten wordt aangetoond dat dit algoritme in staat is om de virtuele gateways op zo'n manier te kiezen dat, bij gebruik van de geselecteerde gateways, de performantie van de netwerken die van het ideale 'Same-MAC'-scenario benadert of zelfs overtreft. Verder wordt eveneens aangetoond dat dit algoritme in staat is om een afweging te maken tussen de verschillende en mogelijk conflicterende performantievereisten van de verschillende netwerken.

Publications

1. Daniel van den Akker, Kurt Smolderen, Peter De Cleyn, Bart Braem and Chris Blondia. *TinySPOTComm: Facilitating Communication over IEEE 802.15.4 between Sun SPOTs and TinyOS-Based Motes*, Sensor Applications, Experimentation, and Logistics, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 29, pp. 177-194, Springer Berlin Heidelberg, 2010.

2. Daniel van den Akker, Bart Braem and Chris Blondia. *anyMAC: adapting the Sun SPOT architecture for MAC development*, 16th Annual Symposium on Communications and Vehicular Technology in the Benelux, Louvain-la-Neuve, Belgium, 2009.

3. Daniel van den Akker, Bart Braem and Chris Blondia. *On the Effects of Interference between Heterogeneous Sensor Network MAC Protocols*, 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, Valencia, 2011, pp. 560-569.

4. Daniel van den Akker and Chris Blondia. *MultiMAC: A Multiple MAC Network Stack Architecture for TinyOS*, 2012 21st International Conference on Computer Communications and Networks (ICCCN), Munich, 2012, pp. 1-5.

5. Daniel van den Akker and Chris Blondia. *Virtual gateways: enabling connectivity between MAC heterogeneous sensor networks*, International Journal of sensor networks, Vol. 14:3 (2013): 133-143.

6. Milos Rovcanin, Eli De Poorter, Daniel van den Akker, Ingrid Moerman, Piet Demeester, Chris Blondia. *Experimental validation of a reinforcement learning based approach for a service-wise optimisation of heterogeneous wireless sensor networks*, Wireless networks: the journal of mobile communication, computation and information, Vol 21:3 (2015): 931-948.

Contents

CHAPTER 1

Introduction

In the 1980s the Defence Advanced Research Projects Agency (DARPA) ran a research program to develop what they at that point referred to as "Distributed Sensor Networks" [1]. The main goal was to develop surveillance and tracking systems in which not only the collection of sensor data, but also the processing required to transform this data into useful target-tracking information, is distributed over multiple geographically separated "sensor/processor" nodes (which thus avoids a 'single point of failure' in the system). To this end they envisioned a system in which "*many spatially distributed low-cost sensing nodes ... collaborate with each other but* [still] *operate autonomously, with information being routed to whichever node can best use the information*" [2].

Although the project was successful in proving the feasibility of this distributed approach, their initial 'sensor network' would not be considered practical by modern day standards. Each 'node' consisted of an equipment rack filled with electronics which had to be built into a truck to allow it to be moved into position. In contrast, modern sensor networks are composed of small, cheap and (usually) battery powered nodes that are equipped with a number of sensors (and sometimes actuators) that allow them to sense (and sometimes interact with) the environment in which they have been deployed. These nodes are, depending on the sensors used, typically [3] around the size of a mobile phone and can even be as small as a grain of rice[1] if needs be. Although the technology powering modern sensor networks is thus very different from that used to build what is now considered [2] to be the very first distributed sensor network, the vision behind these networks is basically the same: multiple nodes (equipped with sensors) that communicate with each other through a (wireless) communication network to provide a combined view of the area in which the sensors are deployed.

[1] `http://www.electronicsweekly.com/news/research/device-rd/isscc-rice-grain-solar-sensor-nodes-uses-cortex-m3-2010-02/`

As the technology powering these networks has evolved over the years, so too have the applications for which these sensor networks are deployed. It should therefore not come as a surprise that today, sensor networks are used for a much wider range of applications than the military ones for they were originally conceived. They are for instance also used in environmental monitoring applications (e.g., tracking of air[2] and water[3] quality), early warning systems for natural disasters (forest fires [4], flash floods [5], ...), agriculture [6], wildlife monitoring [7], factory and warehouse automation [8], and logistics [9] (e.g., monitoring cargo while in transit).

This wide applicability however also poses a challenge to sensor network developers. After all: each application tends to impose its own set of requirements on the sensor network and given the diverse range of applications for which sensor networks are being used, these requirements can vary quite significantly from one application to the next. As an example: supporting node mobility will for instance be an important requirement for sensor networks used in wildlife monitoring, whereas for environmental monitoring applications this is unlikely to be an issue. As a result, network protocols which work very well for one application may be completely unsuitable for another one. In addition sensor network developers also have to keep energy usage to a minimum (batteries should last as long as possible) and allow for the fact that, to keep both cost and energy consumption to a minimum, the sensor nodes themselves tend to be quite resource constrained.

Because of these limitations, it has long been considered standard practice to sacrifice interoperability in favour of performance by heavily optimising not only the application running on the sensor nodes but also the different layers of the network stack to the specific requirements imposed on the sensor network. The rationale for doing so was that since sensor networks were initially mostly deployed in remote, isolated, environments there was no need to account for the possibility of another network being present in the wireless environment of the sensor network itself. It should thus not come as a surprise to find that the majority of the communication protocols that have been proposed by the research community over the years have, for instance, been specifically designed for a particular type of traffic flow or for a specific type of node deployment. Likewise, this preference for highly specialised solutions is also illustrated by the fact that, although a significant number of standards [10, 11, 12, 13, 14] for communication in low power wireless sensor have been established over the years, these standards have not been nearly as well adopted as the standards that exist for other types of wireless networks (such as WiFi or cellular networks).

Over the last decade however, the popularity of sensor networks has grown tremendously[4] and as a result the assumptions underlying the 'traditional' sensor network design principles have increasingly come under pressure. Today it is increasingly common, especially in urban environments, to find multiple independent sensor networks to be deployed in the same area, which means that they are no longer the 'sole user' of the wireless environment in which they are deployed. This means that, even from purely an interference point of view, these networks must either find a way to co-exist peacefully or, as further discussed in chapter 2, suffer the performance-wise consequences of ignoring one another's presence.

[2] https://www.imeccityofthings.be/en/projecten/bel-air

[3] https://vito.be/en/news/internet-water-protects-flanders-against-water-risks

[4] https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

More importantly, the rise of the *Internet-of-Things* (which is "*the extension of Internet connectivity into physical devices and everyday objects*"[5]), means that today, sensor networks are no longer regarded as stand-alone systems. Instead, both sensor networks and the nodes therein are expected to integrate with an increasingly wide range of (other) networks and cloud-based services, which means that for sensor networks interoperability with other networks is becoming increasingly important.

The work presented in this thesis is therefore concerned with enabling interconnectivity (and thus interoperability) between heterogeneous sensor networks. The focus is specifically on the MAC layer of the network stack as the design of this layer has the largest influence on the overall energy consumption of the nodes, has thus been the most extensively customised by sensor network developers (and researchers) and therefore presents one of the largest obstacles to enabling inter-network connectivity.

1.1 Sensor Networks

As briefly discussed above a sensor network is, at its core, a wireless network composed of a (large) number of sensor nodes which are deployed either inside or very near a phenomenon of interest. While in most cases these sensor networks are only used to *monitor* the environment in which there are deployed, there are a number of applications that require the nodes of the sensor network to *interact* with their environment as well. In a home-automation scenario for instance, a 'smart' light may be required to turn on or off based on sensor readings from other nodes in the network. Although the specific properties and requirements of these networks tend to depend for a significant portion on the application for which they are developed, there are also a number of common traits and properties which set these networks apart from other types of wireless networks. The most important of these are briefly discussed below:

- *Limited Energy Reserves*: In most cases, sensor nodes are powered by batteries which means that they only have a limited amount of energy at their disposal. Given moreover that they are often deployed in remote or inaccessible locations, it is in most cases not feasible to replace their batteries once they have been deployed. Moreover, even in those cases where nodes are more readily accessible (e.g., in logistics or building automation scenarios), there is still a significant equipment and labor cost associated with replacing the batteries of the nodes which means that even in those cases the sensor nodes should be made to last as long as possible on a single charge. Since these sensor nodes are, depending on the application, thus expected to remain operational from anywhere between a few weeks to a few years it is clear that energy consumption is by far the most import factor in the design of these networks.
- *Resource constrained hardware:* In addition to being as energy efficient as possible, the nodes in a sensor network also need to be (relatively) cheap. Otherwise, the cost of the hardware alone would make the large-scale deployment of these nodes infeasible. Given that these nodes thus need to be both cheap and energy efficient, they tend to be extremely resource constrained. As an example of the typical hardware capabilities of a sensor node: the widely used *Zolertia Remote* [3] is based

[5] `https://en.wikipedia.org/wiki/Internet_of_things`

on a 32MHz microcontroller with 32K RAM and 512K flash memory. This not only imposes severe limitations on the software running on these sensor nodes but, since these nodes are not powerful enough to run a full-fledged operating system, this also means that both the application(s) as well as the network stack and the drivers usually run directly on the 'bare metal'.

- *Low data rate:* Sensor networks typically exhibit an extremely low data rate compared to other types of wireless networks. Data packets tend to be quite small (10-100 bytes) and are usually generated somewhere between once every few seconds to a few times a day depending on the application. While there may also be occasional traffic bursts (for instance when an event is detected or a software update is applied) these events are considered to be quite rare for most applications. Because of these low data rates, most sensor networks are based on a radio transmitter that is optimised for low energy usage rather than a high data rate. To give an idea of the data rates involved: the IEEE 802.15.4 [10] radios considered in this thesis have a bitrate of between 20 and 250 kbps.
- *Ad-hoc deployment:* Sensor networks are often deployed in environments where there is no usable pre-existing infrastructure available (especially in rural areas). Since there is thus no 'backbone'-network the sensor nodes can connect to this means that, in addition to sensing their environment, nodes may also have to act as an intermediate router for data transmissions from other nodes. Since it is usually not feasible to decide the routing paths and thus the network topology prior to deployment, this means that after deployment, the sensor nodes must be able to self-organise into a suitable network structure. In addition, they also have to be able to adapt to changes in the route topology as the availability of nodes and links may change over time. Sensor networks are thus quite similar to both wireless ad-hoc and wireless mesh networks in this respect given that these types of networks also have to deal with these kinds of problems. That being said, there is a noticeable difference in the scale of the networks involved. Wireless ad-hoc and mesh networks typically consist of a few tens of nodes whereas sensor networks can, depending on the application, consist of multiple hundreds of nodes.

In addition to the properties listed above, sensor network developers may also have to consider a number of limitations and requirements that are more specific to the application for which the network is being developed. Sensor networks used in target-tracking and fire detection applications will for instance be subject to strict *Quality-of-Service* requirements whereas this is less likely to be the case for those used in environmental monitoring applications. Likewise sensor networks may also have to meet certain *Security and Privacy* requirements (such as using MAC-layer encryption) but the exact measures that need to be taken to guard against assailants tend to depend on how privacy sensitive the information collected by the sensor network is, and thus also on the specific application for which the sensor network is being developed.

1.1.1 Applications

Sensor networks are used for a wide variety of applications ranging from environmental monitoring to industrial process automation. Although each of these tends to exact its own set of requirements on the sensor network there are, at least from a network perspec-

tive, also a number of similarities between them. More specifically, most sensor network applications can be divided into one of three categories based on the 'communication patterns' that are generated in the network by the application running on the individual sensor nodes.

Of these three categories, *monitoring applications* are perhaps the most well-known and the most pervasive. Typical examples include for instance environmental monitoring and agriculture, but many 'smart city'[6] applications also fall under this category. In all these applications, the different nodes of the network periodically collect measurements from their immediate surroundings which are then communicated through the network to a 'gateway' or 'sink' (usually located at the edge of the network) for further processing. Given that typically very little processing is done in the network itself, the 'traffic pattern' of these networks is highly directional with packets mostly flowing from the different sensor nodes to the sink but rarely in the opposite direction or from one node to another.

In contrast, sensor networks used for *target tracking* applications are more prone to also process data in the network itself and are thus more closely related to type of sensor network originally envisioned by DARPA. Although target tracking sensor networks are, unsurprisingly, widely used in a military context (e.g., tracking of aircrafts and ground-based weapons, perimeter control, ...) there are also a number of civilian applications that fall into this category such as wildlife monitoring and the tracking of livestock. While, depending on the application, nodes in target tracking sensor networks may still send periodic status updates to a central controller they will also engage in more local 'node-to-node' communication to track a detected target as it moves through the area in which the network is deployed. The 'traffic patterns' encountered in these networks are also more unpredictable than those encountered in sensor networks used for monitoring applications. Rather than generating a slow, continuous, flow of traffic, sensor nodes used for tracking applications usually stay inactive for prolonged periods of time only to generate a 'burst' of traffic when a target is detected in the vicinity of the node. In addition, the quality-of-service constraints found in these types of networks, though dependent on the specific application, tend to be significantly more stringent than those encountered in sensor networks used in monitoring applications.

Finally, sensor networks used for *automation* purposes not only measure the environment in which they are deployed but interact with it as well. In home and building automation applications for instance, sensor networks may not only contain 'sensor' devices such as light switches, IR-sensors, temperature and humidity sensors, but also remotely operated devices such as 'smart' lightbulbs, 'smart' window blinds and climate control systems. As with target tracking sensor networks, sensor networks used for automation engage in both 'node-to-node' communication (e.g., when a 'smart' light is turned on by a nearby light switch) and 'node-to-sink' communication (to report sensor readings to a central controller). It should also be noted that the various 'smart devices' in the network should not only react to information received from other nodes, but also to commands received from the central controller (e.g., to automatically turn off all the lights in an office building at a certain time). This means that, in contrast to the applications discussed above, automation applications usually require the network to support *bidirectional* communication between the different nodes and the sink of the network. In addition to home

[6] `http://smartsantander.eu`

and building automation, sensor networks used in, for instance, industrial process control and factory and warehouse automation can also be categorised under 'automation' sensor networks, but for those networks the traffic will predominantly flow between the different nodes and the sink rather than between individual nodes as the control of the devices tends to be more centralised. In addition, the quality of service requirements also tend to be more strict than for home and building automation applications.

1.1.2 Interoperability in sensor networks

Given the resource- and energy constrained nature of sensor networks, it should perhaps not come as a surprise to find that, traditionally, sensor network developers (and researchers) have gone to great lengths to make the operation of these networks as efficient as possible. This need to be 'as efficient as possible' has not only had a significant influence on the hardware design and the software running on these nodes, but also on the protocols that are used to organise the communication between the different sensor nodes. Given that sensor networks were mostly designed for a single purpose and could, at least initially, be assumed to be operating in an 'isolated environment' with no other networks present, sensor network developers and researchers alike have traditionally had little regard for things like 'standards' and 'interoperability'. Instead, they have, over the years, proposed a wide range of network architectures and communication protocols that have been designed with a specific application in mind. As an example, a number of researchers have proposed [15] to use 'mobile sinks' to collect data from a sensor network; the idea being that by periodically flying a drone over the area where the sensor network is deployed, the sensor nodes can minimise the energy expended in relaying sensor readings to the sink. While this design may work well for sensor networks deployed in remote areas, it is unlikely to be feasible solution for those deployed in urban environments. In addition, cross-layer design [16] has also long been a staple of sensor network development. Typical examples of this include integrating the routing and the MAC layer of the network stack (e.g., integrating [17] route selection with TDMA slot allocation) or integrating the routing layer into the application running on the sensor nodes (e.g., to aggregate sensor readings [18] while en-route to the sink or to route packets based on the data they contain [19] rather than based on the address of the node).

Although standards [10, 11, 12] for communication in low power wireless sensor networks were already established quite early on (the first version of IEEE 802.15.4 dates from 2003), these focussed specifically on home and building automation use cases. Most of these standards however, were (and some still are) proprietary which means that they are of limited use to researchers. While IEEE 802.15.4 does provide an open specification for a PHY and MAC layer for low power sensor networks, the initial versions of this standard were so focussed on the 'infrastructured' network topologies found in home and building automation scenarios that the (initial) IEEE 802.15.4 MAC layer could not be used for other sensor network applications. While the PHY layer specified by IEEE 802.15.4 has been broadly adopted by the sensor network community, the MAC layer of this specification has often only been partially implemented or been replaced entirely by an alternative MAC protocol. It should thus come as no surprise that, as further discussed in chapter 2, there currently exists a wide range of sensor network MAC protocols which are either optimised or even specifically designed with a particular sensor network application in mind. While many of these alternative MAC protocols are the work of sensor network

researchers, they are not the only ones to have replaced the IEEE 802.15.4 MAC layer. The commercial WirelessHart [12] specification for communication in low power wireless networks for instance also uses the IEEE 802.15.4 PHY layer but all other layers of the network stack have been designed from scratch.

A similar observation can be made for the routing layer of the network stack. As with the MAC layer, a number of specialised (and proprietary) solutions for routing in sensor networks have been proposed [20, 21] over the years. While an open standard for enabling IPv6 communication in low power wireless networks (6LowPAN [22]) was also established fairly early on (2007) this standard was, initially, not well adhered to with many implementations only using parts of this specification or only adhering to the 'spirit' rather than the letter of the specification.

Over the last decade however, the rise of the *Internet-of-Things* (IoT) has triggered a shift in priorities when it comes to developing sensor networks. Rather than being regarded as stand-alone isolated systems, these networks and the nodes therein are increasingly expected to integrate with a variety of other (types of) networks and cloud-based infrastructure over the internet. This means that things like 'standards' and 'interoperability', which before were often ignored in favour of specialised solutions, have become an increasingly important factor in the development of current (IoT) sensor networks. It should thus not come as a surprise that in the last few years there have been made significant efforts to both develop and increase the adoption of (new) standardised protocols for communication in IoT networks. This has led, for instance, to the development of a number of new standards for communication in low power wide area networks (e.g., LoRaWan [23], Sigfox [24], NB-IoT [25]), but since most of these are specifically designed for 'cellular-type' network architectures in which the developer of the sensor nodes has no control over the network at all, these are not considered in this thesis. In addition, this has also led to the to the development and widespread adoption of an entire 'tool suite' for IPv6 communication in low power wireless (sensor) networks. The 6LowPAN adaptation layer discussed above is an important part of this tool suite, but has undergone a number of major changes [26] since its first inception. In addition, standardised protocols have also been developed for amongst others, autoconfiguration [27], route selection (RPL [28]) and application-level information exchange (CoAP [29]) in low power wireless (sensor) networks.

Although these standardisation efforts have gone some way to improving interoperability at the network layer, little work has been done to address the interoperability issues that exists at the MAC layer of the protocol stack. Moreover, the problem of enabling MAC-layer interoperability between sensor networks is one that is unlikely to be solved through standardisation alone. The MAC layer is after all responsible for deciding when to turn the radio on and off and given that, as discussed in [30], sensor nodes spend a significant portion of their energy reserves on (wireless) communication, the overall energy consumption of the sensor nodes is strongly dependent on the specific MAC protocol used. As a result, the energy wise cost of switching from a highly optimised to a more generic (standard) MAC protocol is in most cases simply too large for this to be a feasible solution for enabling interoperability. This means that, despite there being a number of scenarios in which it would be useful to do so (see sections 1.3 and 1.4), it is in general not possible for two sensor networks deployed in the same (wireless) environment to *locally* exchange information with one another.

1.2 Challenges and Contributions

This thesis focusses on the problem of enabling link-level connectivity between sensor networks using different MAC protocols (hereafter referred to as MAC-heterogeneous sensor networks). Given that, as discussed above, it is not feasible to do so through standardisation alone (i.e., forcing both networks to use the same MAC protocol), this thesis uses an approach whereby each network continues to use its own MAC protocol and specialised *virtual gateway* nodes are used to bridge the communication gap between the two networks. As will be further discussed in chapter 3 this does not require any special hardware to be developed. Instead, already deployed sensor nodes can be configured as a virtual gateway by performing a software update.

While this approach thus has a number of practical advantages it also presents a number of issues and challenges that warrant further investigation. The first one of these is the matter of coexistence between MAC-heterogeneous sensor networks. When two MAC-heterogeneous sensor networks are deployed in the same wireless environment, they will interfere with one another due to the fact that, as discussed in chapter 2, the MAC protocols of these networks are not designed to take one another's presence into account. Given that there is little point in enabling communication between these networks if the interference that exists between them prevents them from coexisting in the same wireless environment, the effect of this interference on the performance of the respective networks first needs to be investigated.

The next question to be answered is whether or not it is feasible to actually use (existing) low-power sensor nodes as virtual gateway devices. As further discussed in chapter 3, this requires these nodes to be able to run multiple MAC protocols simultaneously. The main challenge with this is that sensor nodes (usually) only have a single radio interface available. In addition, running more than one MAC protocol also introduces a certain overhead in terms of both energy usage and processing time. Given that in sensor networks both of these are at a premium, not only the (technical) feasibility of using low-power sensor nodes as virtual gateways, but also the minimal system requirements and performance overhead of doing so, need to be evaluated.

Another challenge in the use of *virtual gateways*, is the matter of deciding which nodes to configure as a virtual gateway. Given that it only makes sense to configure a node as a virtual gateway if it is actually used to exchange information between the respective networks, it is clear that the placement of these virtual gateways depends on the routing paths that are used in and between the respective networks. Inversely, these routing paths also depend on which virtual gateways are being used given that all inter-network communication needs to be routed over a virtual gateway. Another challenge with the selection of the virtual gateway nodes is that, depending on the specific use case, the administrators of these networks may each impose their own set of (performance) requirements on the network and that all these requirements need to be taken into account in the selection of the virtual gateway nodes.

This thesis therefore focusses on addressing the issues and research questions outlined above. More specifically, the contributions of this thesis can be summarised as follows:

- It first investigates co-existence between MAC-heterogeneous sensor networks. To

this end it examines how the performance of these networks is affected, under a number of different traffic conditions, by the interference that exists between them as a result of using heterogeneous (incompatible) MAC protocols.

- It introduces *virtual gateways* as means to enable link-level communication between two MAC-heterogeneous sensor networks. It presents a network stack architecture that allows multiple MAC protocols to be run on top of a single radio interface, investigates the feasibility of using low-power sensor nodes as virtual gateways, determines the performance-wise cost of configuring such a node as a virtual gateway and examines how the introduction of virtual gateways in the wireless environment influences the performance of the networks involved.
- It proposes and evaluates a heuristic algorithm for the selection of virtual gateway nodes that operates based on topology- and performance information collected from the wireless environment as well as the performance requirements specified by the administrators of the respective networks.

1.3 Symbiotic Networking

A significant portion of the work presented in this dissertation was inspired by and started during the IWT-SBO *SymbioNets* [31] project, which was aimed at developing the architecture, algorithms, services and network protocols to support the *symbiotic networking* paradigm. For this reason, both the concept of symbiotic networking, as well as the relevant aspects of the SymbioNets project are briefly discussed below. A more detailed explanation of this concept, as well some of the challenges involved can be found in [32].

1.3.1 Background & Definition

The idea behind symbiotic networking stems from the observation that although today there is usually more than one wireless network present in any given environment (e.g., cellular networks, WiFi, bluetooth, sensor networks, DECT, ...), these networks generally do not take one another's presence into account. Instead each network tries to optimise its own operation without considering the impact on other wireless networks located in the same environment and, possibly, operating in the same wireless spectrum. Since optimisation is thus performed based on a *local* view of the environment, this does "*...not result in efficient communication from a global point of view*" [32]. While there previously had been developed frameworks to enable limited optimisation and cooperation across networks (e.g., cognitive radio, opportunistic networking, ...), the symbiotic networking paradigm applies these concepts in a much broader sense.

More specifically, symbiotic networks are "*independent, co-located homogeneous & heterogeneous, wired & wireless networks that cooperate across all layers and across network boundaries through advanced sharing of information, infrastructure and (networking) services*" [31]. By cooperating in this manner, networks are able to reach a global optimum in terms of, for instance, spectrum utilisation, energy consumption and QoS guarantees. As discussed in [32], this cooperation can take many forms such as:

- *Sharing of information* such as exchanging spectrum information and PHY-parameters used (encoding, tx-power, ...) to reduce interference between the networks;
- *Sharing of infrastructure* such as routing traffic over one another's nodes, sharing processing, memory and storage capacity across networks;
- *Sharing of network & application services*: examples include positioning, time synchronisation, in-network processing/aggregation of sensor data, combining data sources from different networks, ...

1.3.2 Use Cases

To give a better understanding of what symbiotic networks are and how these networks might cooperate, two of the use cases that were explored in the SymbioNets project are discussed below:

Home and Office Scenario

A home or office environment is considered in which multiple wireless networks are deployed for a wide variety of applications. These can include for instance:

- One or more independently operated WiFi networks, which provide network and internet connectivity to the various user devices located in the environment (laptops, smartphones, ...)
- Sensor networks installed by different vendors: e.g., one for home automation (climate control, lighting), one for burglary prevention (motion detection / monitoring of windows)
- External cellular networks which provide voice- and data services to cell phones

Under normal circumstances, all these networks are unaware of each other: there is no direct communication between nodes of different networks and networks using the same spectrum (e.g., WiFi and sensor networks) will, to some extent, interfere with one another. When these networks engage in symbiotic cooperation, they exchange spectrum information and coordinate channel allocation across the different networks to either minimise or even eliminate inter-network interference. In addition, these networks also cooperate at the MAC-, routing- and application layer of the network stack. The access-points of the different WiFi networks for instance seamlessly align to increase one another's coverage and voice calls that were previously directed through the cellular network are now offloaded to the nearest WiFi access-point. Likewise, the sensor networks now use each other's nodes and uplink-gateways (sinks) to reduce the number of hops (and thus the amount of energy) needed to pass sensor readings along to their respective control servers. In addition, these networks now also exchange information at the application level. Information from the motion sensors in the security network is now used by the home automation network to better control the lights in the different rooms while information from the home automation network (e.g., lights being manually turned on or off) is used by the security sensor network to better detect suspicious activities.

Logistics Scenario

This use case considers the (wireless) networks that are used in the management of (international) shipping operations. Cargo containers carrying perishable, valuable or shock-sensitive goods are equipped with multiple wireless sensors (temperature probes, accelerometers, ...) to track the condition of the cargo as it moves through the different links in the logistics chain. Likewise, each logistics provider (shipping/trucking companies, warehouses, container terminals, ...) uses its own infrastructure to facilitate the processing of shipments. Trucks are for instance equipped with a GPS receiver and a PDA to allow their location to be tracked remotely and to provide information to the driver about the container being moved (e.g., weight, destination, ...). Likewise, warehouses and container terminals employ their own (sensor) networks to keep track of the different containers as they are unloaded, possibly moved to temporary storage and finally loaded onto a new truck, train or ship.

Despite operating in the same wireless environment, these networks normally do not communicate with one another as they are all owned and operated by different stakeholders. As further discussed below, the symbiotic networking paradigm overcomes these barriers by having networks automatically *negotiate* on the precise nature of the cooperation of the networks and once a (symbiotic) trust relation has been established the networks engage in advanced information sharing and (application-level) cooperation. Rather than using its own (expensive) cellular data uplink, the sensor network in the cargo container now uses the infrastructure of the (current) logistics provider to report sensor readings back to the owner of the cargo. Similarly, information from the cargo sensor network is also made (directly) available to services running in the network of the logistics provider. This information is then used to, for instance, alert the driver of a truck in case the temperature inside the container becomes too high or help the control centre of a container terminal to determine the priority at which the container needs to be processed.

1.3.3 Mechanisms for Cooperation

This section briefly outlines the strategy used in the SymbioNets project to enable symbiotic cooperation between multiple independently owned and operated wireless networks. Before any cooperation can take place, each network needs to be provided with a so-called *Symbiotic Profile*. This is a self-describing data structure that contains, amongst others, the following information:

- The *incentives* the network has to engage in symbiotic cooperation. (e.g., “I want better coverage/less interference”, “I want access to services running in another network”, “I want to reduce energy consumption”, ...)
- The *identity* of the network. This is in essence a cryptographically secure identifier or certificate, similar to those used to identify websites. This information is used to establish the trust relationship needed for cooperation.
- The *capabilities* of the network. For example: supported interference avoidance algorithms, supported routing protocols/metrics (for routing packets between networks), services running in the network, ...

Cooperation between the networks is then enabled based on the information on these

profiles. This is done in three distinct phases.

During the *Discovery Phase*, networks attempt to discover neighbouring nodes of other (symbiotic) networks. This is done by so-called discovery nodes which regularly broadcast packets to announce the presence of the network and scan the different channels for packets broadcasted by other 'foreign' discovery nodes. Once these discovery nodes have 'found' one another, the symbiotic profiles of respective networks are exchanged and forwarded on to the 'symbiotic controller' node(s) of the network responsible for managing interactions with other networks.

Once the symbiotic profiles have been exchanged, the networks enter the *Network Binding Phase*. During this phase, the 'symbiotic controllers' of the respective networks communicate with one another, through their respective discovery nodes, to exchange further information and negotiate on a possible cooperation between the networks. Depending on the capabilities, incentives, and the level of trust that exists between these networks (based on the identity), this cooperation can range from simple active/passive 'interference avoidance' to routing packets for one another, providing access to each other's services and so on (not cooperating is of course also an option). After negotiation both networks adjust their (network) configuration to enable the network-level cooperation agreed-upon during negotiation. This can for instance include adjusting firewall rules, installing new routes for inter-network communication and setting up monitoring mechanisms.

Once communication between the networks has been established at the network-level, the networks enter the *Service Enabling Phase*. During this phase, application-level cooperation is enabled between the networks. This can for instance include the dissemination of the interfaces through which the available services can be accessed, provisioning services requested by the other network and setting up data-streams between services of different networks.

Although this strategy for enabling cooperation allows many of the administrative and technical barriers that separate wireless networks to be overcome, it does require nodes from different networks to at least be able to exchange information at the MAC-layer of the network stack. In the case of sensor networks this is not necessarily the case since different sensor networks may use different MAC protocols depending on their specific requirements. As a result, the work presented in this thesis is instrumental in enabling symbiotic cooperation between these MAC-heterogeneous sensor networks. The virtual gateway functionality for instance allows communication to be established between the discovery nodes of the respective network using a (temporary) common MAC protocol. In addition, this also enables the symbiotic controllers of the respective networks to exchange information and thus negotiate on a possible cooperation. Once negotiation between the networks has been concluded, these virtual gateways are used to enable communication between the respective networks while the selection algorithm presented in this work enables these networks to optimise the number and the location of the virtual gateways based on the QoS- and energy requirements agreed upon during the negotiation.

1.4 MAC-level interoperability in IoT-networks

Although the principles and algorithms presented in this thesis were originally developed with symbiotic networking in mind, it should be noted that this is not the only domain

in which link-level interoperability is an issue. As briefly discussed in section 1.1.2, the rise of the Internet-of-Things has reintroduced sensor network developers to the concept of 'interoperability', but most of the (initial) work done in this respect has focussed on enabling (end-to-end) connectivity for the centralised, cloud-based architectures that seem to be the focus of IoT infrastructures today.

This centralised approach however is not without its problems. In addition to security and privacy issues[7], researchers have also identified [33, 34, 35] issues related to the scalability, reliability, bandwidth requirements and especially the latency of cloud-based IoT infrastructures. Under the titles "Fog Computing" and "Edge Computing" they have therefore proposed to move the processing of sensor data (as well as the control over these sensors) to the "edge" of the network in which these sensors reside. (e.g., the sink of the sensor network, a local controller in the LAN of a smart home environment, ...)

Coupled with this push for a more distributed approach, both researchers [36, 37, 38] and policy makers [39] have also called for the development and adoption of open standards and other solutions to facilitate integration and interoperability between all these different environments. Under the "IoT-EPI" initiative the European Union has for instance funded several H2020-projects which specifically deal with enabling interoperability in IoT environments. The *SymbIoTe* [40] project for example developed an open framework and API to enable interoperability at the "edge", cloud and application level of the IoT architecture. In contrast the *INTER-IoT* project aimed to enable "*voluntary interoperability at any level of IoT platforms and across any IoT application domain*" [41]. To do so, they propose to use a combination of gateway devices, virtualisation software defined networking and 'overlay APIs' to bridge the gap between existing IoT platforms. Likewise, the *AGILE* [42] project focussed even more on a gateway-based approach by developing a gateway device and modular software stack to translate between devices using different IoT technologies.

The European Union is of course not alone in recognising the need for better interoperability in IoT networks and over the last few years other people have also proposed various solutions to tackle this problem. These include for instance the development of further standards for machine-to-machine communication [43] and the development of (standardised) semantic annotations [44] to allow data to be correctly interpreted across different frameworks. More closely related to the 'device' and 'network level' of the "IoT-stack", researchers have also developed several gateway-based solutions to (locally) translate between different IoT technologies, protocols and platforms. [45] and [46] for instance both present "protocol translators" that run on the (uplink) gateway of the local network and convert between different application-level protocols (such as CoAP, XMPP and HTTP). In contrast, [47, 48] and [49] all propose to use dedicated hardware devices to bridge between different wireless technologies. [47] for instance propose to use smartphones for this purpose as these are already equipped with a large number of radio interfaces used for (domestic) IoT applications today (Bluetooth, Ant+, WiFi, NFC, RFiD, ...) while [48] and [49] use custom hardware to bridge between different technologies.

The work presented in this thesis contributes to this overall effort of enabling interoperability in IoT-networks by addressing the problem of heterogeneity at the MAC-layer of

[7] https://www.forbes.com/sites/theyec/2018/07/31/10-big-security-concerns-about-iot-for-business-and-how-to-protect-yourself/

the network stack. The approach used here is similar to existing gateway-based solutions in the sense that, like these solutions, it enables interoperability by translating between different (existing) communication protocols. At the same time, it is also complementary to these solutions because it does not require any specialised 'smart gateways' to be deployed and instead reuses the existing infrastructure. This not only reduces the hardware cost associated with the introduction of additional gateway nodes into the environment but, given that the virtual gateways introduced here don't necessarily have to be located at the "edge" of the network, also gives the network operator more options in deciding the location of the 'crossover' points between the heterogeneous networks. (The selection algorithm introduced in this thesis can assist in deciding these locations.) As a result, the work presented in this thesis can not only be used to enable cooperation between sensor networks in the *symbiotic networking* use case, but also to remove some of the barriers for communication that exist in IoT-networks today.

1.5 Thesis outline

The remainder of this dissertation is outlined as follows:

Chapter 2 first gives a detailed explanation of the test setup used throughout this thesis. More specifically it gives an overview of the currently available sensor network MAC protocols, discusses the specific MAC protocols that will be considered in this thesis, details the simulator and the network stack used to perform most of the experiments and explains the specific application scenarios considered in this thesis. Afterwards, this chapter investigates the problem of interference between MAC-heterogeneous sensor networks. The focus is on determining to what extent the performance of these networks is affected under varying traffic and interference conditions and determining the viability of enabling communication between such MAC-heterogeneous networks under these conditions.

Chapter 3 introduces *virtual gateways* as a means to enable communication between MAC-heterogeneous sensor networks and investigates the feasibility of this approach. It discusses the software architecture developed to provide this functionality and investigates its performance overhead when implemented on a sensor node. The extremely resource-constrained Tmote Sky [50] sensor node is used as a test-platform to do so. In addition, this chapter also provides an initial assessment of the effect these virtual gateways can have on the network-wide performance of the sensor networks involved.

Chapters 4 and 5 next consider the problem of deciding which sensor nodes to configure as a virtual gateway. The *IRVG*-algorithm (Iterative Removal of Virtual Gateways) is proposed as a means to automatically determine the specific virtual gateways to use based on the topology and the requirements of the networks. As the development of this algorithm required two separate research questions to be answered, the explanation and evaluation of this algorithm is split over two separate chapters. Chapter 4 first discusses the requirements and limitations to be considered in the selection of the virtual gateway nodes and then provides a high-level overview of IRVG itself. Afterwards, it focusses on the problem of predicting how the performance of the networks is affected by the removal of one or more gateways. It explains the prediction algorithm used for this purpose after which the precision of this algorithm is evaluated for a wide range of possible virtual

gateway deployments.

Chapter 5 continues the discussion of IRVG but focusses on the problem of selecting the virtual gateways to use based on the predictions made by the prediction algorithm. It discusses both the selection algorithm developed to do as well as the reward function used to allow network administrators to fine-tune the operation of this algorithm to their specific requirements. Afterwards the performance of IRVG is evaluated in its entirety for a number of different network requirements and MAC protocols.

Finally, chapter 6 applies IRVG to the *node-to-sink* rather than the *random-flows* scenario (both of these scenarios are further discussed in section 2.2.2). After first discussing the modifications made to support this scenario, the (modified) prediction and selection algorithm are then once again evaluated for a number of different (virtual gateway) deployments, network requirements and MAC protocols.

CHAPTER 2

Coexistence between MAC-Heterogeneous sensor networks

As briefly discussed in the introduction of this thesis, the wide range of applications for which wireless sensor networks are being used, coupled with the extreme resource-constrained nature of the devices employed in these networks, has led, over the years, to the development of a wide variety of sensor network MAC protocols. Most of these have been designed with a specific (type of) application in mind and therefore (attempt to) optimise a specific set of performance metrics (such as for instance energy usage or delay) under a specific set of assumptions about the networks (e.g., bandwidth requirements, node deployment, ...).

Although such optimised protocols allow for a more efficient operation of the sensor network, they operate under the implicit assumption that all nodes in the wireless environment make use of the same MAC protocol and that the sensor network itself is thus deployed in an isolated environment. As the popularity of sensor and low-power Internet-of-Things networks continues to grow however, this is increasingly less likely to be the case and it is rapidly becoming rule rather than exception to find multiple sensor networks deployed in the same wireless environment. In that case, two different sensor networks operating in the same wireless environment will most likely use different MAC protocols. Since these MAC protocols have not been designed to take the presence of other MAC protocols into account, they are incompatible and as a consequence they will interfere with one another. The level of interference will clearly depend on the type of MAC protocols.

Before attempting to enable communication between these MAC-*heterogeneous* sensor networks, this chapter therefore first investigates how this interference affects the network performance of these heterogeneous sensor networks. The rationale for doing so is that there is little point in enabling communication between these networks if the inter-

ference between the MAC protocols precludes them from coexisting in the same wireless environment.

Although interference in sensor networks has been well studied, these studies mostly focus on interference between sensor networks (using the IEEE 802.15.4 2.4GHz PHY-layer) and other technologies using the same frequency band (such as for instance WiFi networks and Bluetooth). [51] and [52] both investigate the impact of IEEE 802.15.4 on IEEE 802.11b networks and while [51] conclude that, "*...the IEEE 802.15.4 network will typically have little to no impact on the IEEE 802.11b's performance*", [52] shows that IEEE 802.15.4 can still negatively affect the WiFi performance when the sensor nodes are located very close to the WiFi-receiver. Inversely, [53] experimentally investigates the effect of WiFi and Bluetooth interference on IEEE 802.15.4-based networks and finds that while the interference from Bluetooth networks has a noticeable but manageable effect on the performance, the interference from WiFi networks can have a "*critical*" effect on the performance of IEEE 802.15.4. [54] also experimentally investigates the effects of WiFi interference on the performance of sensor networks using the IEEE 802.15.4 PHY-layer and then proposes a number of techniques to minimise the effects of this interference at the MAC-layer (such as using Forward-Error-Correction). [55], [56] and [57] use a more analytical approach to study coexistence between WiFi and IEEE 802.15.4. [55] finds that the precise effects of interference between WiFi and sensor nodes depends on how well the Clear-Channel-Assessment (CCA) mechanisms of the nodes are able to detect the interfering signal. Since this depends on both the (fixed) transmission power and the distance between the nodes they identify three separate 'interference regions' and construct a 'coexistence model' for each of those. Meanwhile [56] uses beaconing in combination with a distributed channel selection mechanism to avoid harmful interference from WiFi in IEEE 802.15.4 networks. Finally, [57] proposes to equip both WiFi and IEEE 802.15.4 nodes with additional hardware to allow the CCA-mechanism to properly detect transmissions of a different radio-technology in the frequency band used by the node. This allows the respective CSMA/CA protocols to 'back-off' from transmissions made by the other network and thereby allows these networks to co-exist despite using different radio technologies.

In contrast to the work discussed above, [58] does consider interference between nodes using the same PHY-layer. To generate repeatable patterns of interference they use a so-called 'interference engine' that uses the same IEEE 802.15.4 radio-chip as the nodes being interfered with. After establishing the impact of this interference on the network performance they then propose a number of modifications to the X-MAC [59] sensor network MAC protocol to make it more robust against this interference. While their work is thus more closely related to the interference-scenario being investigated in this chapter there are a number of significant differences between their test-setup and the scenarios considered here. More specifically, they only investigate interference on a very small scale (one sender, one receiver and one interference source) and have designed their test-setup so communication between the sender and receiver is completely impossible while the interference-source is active. In contrast, the work presented in this chapter considers interference on a larger scale (between networks rather than between individual sensor nodes) and uses a realistic channel model (see section 2.2) that also considers the distance between the various nodes. In addition in [58] the generated interference is independent from the MAC protocol used by the sensor nodes whereas in the scenario

considered here, the networks interfere with one another and the exact effects of the interference between the networks will therefore depend on the specific MAC protocols used by the respective networks.

The rest of this chapter is organised as follows: section 2.1 first provides an overview of currently available sensor network MAC protocols and then discusses the specific MAC protocols considered in this thesis. Section 2.2 details the simulator, application-scenarios and test setup used in this thesis after which the effects of inter-MAC interference are discussed separately for each considered scenario in sections 2.3 and 2.4. A conclusion is provided in section 2.5.

2.1 Sensor network MAC protocols

There is currently a wide range of MAC protocols available that have been specifically designed for use in low-power sensor networks. Given that a low energy-usage is often of critical importance in these types of networks and that sensor nodes expend a large part of their energy reserves on communicating with other nodes [30], it should come as no surprise that these sensor network MAC protocols have mostly been designed to be as energy efficient as possible for the specific use case envisioned by the developers. To understand how this affects the design of these MAC protocols it is important to know how energy is "wasted" when communicating with other nodes. [60] identifies the following sources of energy waste at the MAC layer of the network stack:

- *Idle listening*: This occurs when a node has its radio chip set in 'RX-mode' while the node is not receiving any packets. Given that the radio chip typically consumes most energy while it is in 'RX-mode' [61], idle listening can result in a significant energy-wise performance overhead.
- *Overhearing*: This occurs when a node receives packets destined for another node. This causes the node to waste energy since the radio was in 'RX-mode' unnecessarily and it also causes the node to expend power processing (and discarding) the received packet.
- *Collisions / Overemitting*: This occurs when a packet is lost either due to collision with another packet or because the receiving node was not ready to receive the packet at that time (e.g., because its radio is not in 'RX-mode'). In either case, the packet will have to be retransmitted.
- *Control Traffic*: While sending control packets may be necessary for the MAC protocol to operate from an application point-of-view these packets do not carry any useful information and as a result they are also a source of energy waste.

Although all of these affect, to some extent, the energy consumption of the node, *idle listening* has by far the most impact. This is because sensor networks generally operate under low to medium-traffic conditions and that the radio will therefore be idle for a significant portion of the time. Because of this, most sensor network MAC protocols focus primarily on minimising idle listening by 'duty cycling' the radio of the sensor node. This, in essence, means that the radio is switched off for most of the time and is only periodically enabled to communicate with other nodes. Hence, the 'duty cycle' of the node refers to the percentage of the time that the radio is 'turned on'. Depending on

the specific use case, these MAC protocols may also have to be able to meet a number of 'secondary requirements'. These can for instance include meeting additional performance requirements (such as a specific delay), being able to cope with frequent topology changes (e.g., because of node movement or nodes leaving / joining the network) and being able to cope with varying network sizes and densities.

Given that these MAC protocols thus have to be able to fulfil a wide range of stringent requirements in a resource constrained environment, it is not uncommon for these MAC protocols to make certain assumptions about the network. It is, for example, not uncommon for a MAC protocol to be specifically designed for a specific type of 'communication pattern'. [62] and [63] for instance assume all traffic to adhere to a 'many-to-one' pattern, which is primarily the case for node-to-sink reporting applications. Conversely, [64] is designed primarily for local gossiping between local sensor nodes, which is more common in target-tracking applications. In addition, MAC protocols often also impose a certain structure in the topology of the network. [65], [66] and [67] for instance organise the network into multiple clusters whereas [62] and [68] require the network to be organised along a tree-structure.

More importantly (at least as far as interference between MAC protocols is concerned), these 'secondary requirements' also have a large influence on the mechanisms used to govern channel access and coordinate duty cycling between the nodes. Depending on the mechanisms used these MAC protocols can be divided into a number of categories, which are discussed in the remainder of this section.

2.1.1 CSMA Based MAC protocols

As discussed in [69], the easiest (viable) method of organising access to the channel is to use some form of CSMA [70], possibly in combination with a number of collision avoidance techniques. CSMA protocols for sensor networks are often heavily simplified versions of IEEE 802.11's DCF [71]. The CSMA implementations included in the TinyOS [72] and Contiki [73] sensor network operating systems[1][2] for example are based on the "low power" version of IEEE 802.11 (IEEE 802.15.4) but use an unslotted (rather than the slotted) version of CSMA to access the channel. This is done to avoid having to do clock synchronisation between the nodes. In addition, they also omit the RTS-CTS mechanism of IEEE 802.11. As noted in [68], the RTS-CTS mechanism is mostly beneficial when the actual data transmission required significantly more airtime than the RTS-CTS packets themselves. In the case of IEEE 802.15.4-radios, which have been specifically designed for low power sensor networks, the maximum packet size is quite small (at most 127 bytes including MAC-headers) and as a result the overhead of using RTS-CTS is significantly larger (40% to 70% [74]).

While CSMA-based MAC protocols are very versatile and easy to implement, they do require the radio to be enabled at all times. While this makes them unsuitable for most sensor network scenarios it is worth mentioning them here since they form the basis for some of the more advanced MAC protocols discussed below. In addition, they are still

[1]TinyOS CSMA Implementation: `https://github.com/tinyos/tinyos-release/blob/tinyos-2_1_2/tos/chips/cc2420/transmit/CC2420TransmitP.nc`

[2]Contiki CSMA Implementation: `https://github.com/contiki-os/contiki/blob/release-2-7/core/net/mac/csma.c`

quite useful for applications where energy usage is less of an issue (e.g., home and building automation).

2.1.2 Scheduled MAC protocols

Scheduled MAC protocols try to reduce the energy used by 'plain' CSMA by periodically turning the radio off and on. S-MAC [75], one of the earliest "true" sensor network MAC protocols, does this by dividing the channel into multiple 'frames' which consist of a 'listen' period during which the radio is enabled and a 'sleep' period during which the radio is disabled. As shown in figure 2.1, the length of these periods is configured through the *activeTime* and *frameTime* parameters which must be set by the network administrator. Although the duration of the 'listen' and 'sleep' period is thus the same for all nodes these frames are not necessarily synchronised between nodes. Packets are always sent during the 'listen'-period of the receiver and CSMA is used in combination with the RTS-CTS mechanism of IEEE 802.11 to gain access to the channel. To ensure that packets are sent during the 'listen'-period of the destination node, each node maintains a schedule table that contains the schedules of its neighbours. Nodes announce their schedule by broadcasting a so-called 'SYNC'-packet every few frames and periodically stay awake for an entire frame to discover new neighbours. Although S-MAC does not enforce the use of a single schedule, the different nodes of the network will try to synchronise to a common sleeping schedule where possible. To do so each node will, at startup, adopt the schedule of the first 'SYNC'-packet they receive. If, after a random time, no 'SYNC'-packet is received the node selects its own schedule.

Although S-MAC succeeds in limiting the energy wasted because of idle listening, it does have a number of limitations, the most important of which is that the length of the 'listen' and 'sleep'-period is statically configured. This not only requires the network administrator to fine-tune the *activeTime* and *frameTime* parameters to the specific requirements of the network but also prevents S-MAC from adapting to variations in the required bandwidth. This is, for instance, a problem for event-monitoring or target-tracking applications which tend to require additional bandwidth when a target (or event) is detected. In addition, S-MAC also requires the length of the 'listen' and 'sleep'-period to be the same for all nodes in the network. This is, for instance problematic for "node-to-sink" data-gathering applications in which nodes closer to the sink require more bandwidth than nodes more distant to the sink since in this case length of the 'listen'-period must be tuned towards the maximum required bandwidth, which in turn will cause nodes further

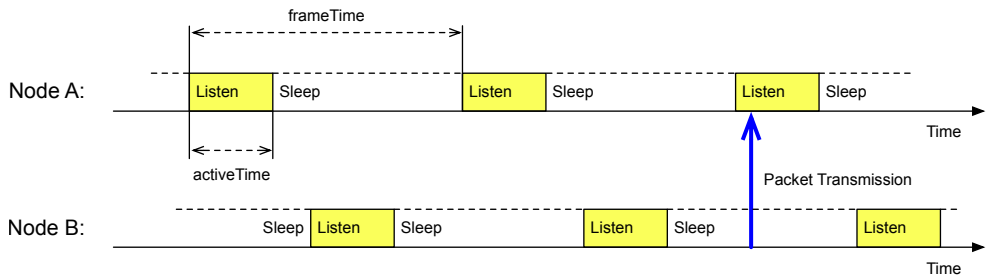


Figure 2.1: Frame structure used by the S-MAC protocol.

away from the sink to keep their radio on for an unnecessarily long period of time.

The authors of [76] propose a number of changes to S-MAC to make it more suited for node-to-sink applications. The most important change is that, in contrast to S-MAC, their "T-MAC" protocol will vary the length of the listen period based on the amount of traffic in the immediate vicinity of the node. Each frame begins with an "active period" of a certain minimum length during which the node will turn its radio on, listen to the channel and will, if needed, transmit any queued packets to neighbouring nodes. Every time the node detects a so-called *activation-event* during the active period, it will extend its active period so the radio will remain on for at least another "TA" time after the event. As discussed in [76] these activation-events include:

- "*the reception of any data on the radio*"
- "*the sensing of communication on the radio*" (i.e., the clear channel assessment mechanism of the radio indicating that the channel is busy)
- "*the end-of-transmission of a node's own data packet or acknowledgement*"
- "*the knowledge, through overhearing prior RTS and CTS packets, that a data exchange of a neighbour has ended.*"

This means that as long as there is still activity on the channel in the vicinity of the node, that node will stay awake to exchange packets with its neighbours. Once no activation-event has been detected for at least a period "TA", the node will turn its radio off until the start of the next frame. Any packets arriving from the network layer during that time are queued until the next frame begins. Just like S-MAC, T-MAC uses the RTS-CTS mechanism of IEEE 802.11 to solve the hidden-node problem but unlike S-MAC, T-MAC does not make use of CSMA when sending RTS packets. The rationale for doing so is that (queued) packets are sent in a burst at the beginning of the frame and that, as stated by the authors: "*a node may* [therefore] *expect to be in a fierce fight for winning the medium every time it sends an RTS. An increasing contention interval is not useful, since the load is mostly high and does not change*"[76]. Instead of using a back-off scheme, T-MAC therefore uses a fixed contention interval that is tuned for maximum load. T-MAC also includes two mechanisms that can be used to prevent nodes from ending their active period prematurely (i.e., when other nodes still have packets queued for them). Firstly, nodes whose transmit buffer is almost full can give priority to sending rather than receiving packets. To do so a node will, upon receiving an RTS-packet, send an RTS-packet of its own to a different node rather than sending a CTS-packet to the sender of the original node. By doing so, the node has a better chance of sending its queued packets before the receiver goes to sleep. Secondly, nodes are allowed to send a so-called "Future-Request-To-Send" (FRTS) packet to a node after overhearing a CTS-packet intended for another node. By sending such an FRTS packet, a node can signal to another node that it still has packets queued for them but is unable to send them because it has lost contention with another node.

While S-MAC and T-MAC are undoubtedly the most well-known scheduled MAC protocols, there are also a number of other sensor network MAC protocols of this type available. DSMAC [77], like T-MAC, also tries to improve on S-MAC by varying the length of the active period with the load in the network. Unlike T-MAC however, this is not done by detecting channel activity. Instead, nodes announce the current size of their transmission-

queue to their neighbours by embedding the average "queue-time" into data packets being sent. This information is then used by the receiving node(s) to dynamically tune the length of their listen-period to the network-load in the vicinity of the node. The listen-schedule of the node is then communicated back to the sending nodes by embedding it in the SYNC-messages that are periodically sent. P-MAC [78] also builds upon the foundation of S-MAC but instead of using a fixed-length 'frame'-structure it instead uses so-called "*sleep-wakeup patterns*" which are generated based on the traffic requirements of both the node itself and the surrounding nodes. Although the authors of P-MAC manage to achieve a significant performance improvement over S-MAC it should be noted that their protocol does require the channel usage requirements to be known in advance, which is generally not the case.

D-MAC [68] tries to minimise the delay of S-MAC for nodes organised in a tree-structure. The authors note that using S-MAC in a node-to-sink scenario incurs a significant end-to-end delay. This is because, when transmitting a packet to the sink, the intermediate nodes along the routing path to the sink will often enter sleep mode before the packet reaches the sink and as a result the packet has to be queued until the next active period begins. The authors of D-MAC address this issue by using a "staggered" wakeup schedule in which nodes further away from the sink wakeup slightly before nodes closer to the sink so sensor readings can be relayed from the sensor to the sink without having to wait for any intermediate nodes to wake up.

The authors of [79] propose to use reinforcement learning to optimise the listen/sleep schedule of the nodes in the network. Their RL-MAC protocol uses a frame structure similar to S-MAC but the length of the active period is decided using a Q-Learning algorithm, the reward function of which balances between the packets queued at the node itself, the node's duty cycle and the number of attempts neighbouring nodes require to successfully deliver packets to the node (a high number of attempts indicate that the node's active period is too short to receive all packets sent by neighbouring nodes). Finally H-MAC [80], combines elements of scheduled MAC protocols and TDMA MAC protocols to allow nodes to send unicast-transmissions during the sleep period rather than during the listen period. To do so, the sleep-period is divided into fixed-length slots (as is the case in TDMA) and nodes use the active period to agree on a slot for the actual unicast transmission. This allows the length of the active period to be greatly reduced and ensures that only the nodes involved in the unicast transmission are awake while the transmission is taking place.

2.1.3 Low Power Listening Protocols

Low Power Listening MAC protocols use a different approach for reducing the duty cycle of the node. They try to address the issue that in many sensor network scenarios the nodes spend most of their time idling and will only occasionally have data to transmit. In those cases, maintaining a listen/sleep cycle can be quite expensive given that in most listen-periods no traffic will be sent at all. Instead of maintaining a listen/sleep schedule, Low Power Listening MAC protocols will keep the radio disabled for most of the time but will periodically sample the channel for activity. This is usually done by (briefly) turning the radio on, performing a CCA-check and immediately turning off the radio if no activity is detected on the channel. When transmitting a packet, the sending node

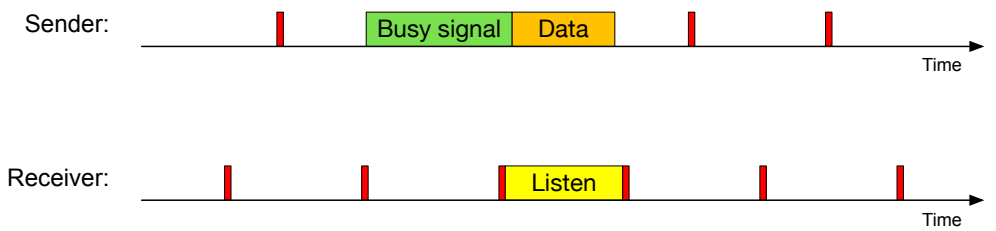


Figure 2.2: Synchronisation in Low Power Listening MAC protocols.

"wakes up" the receiver by prefixing the data packet with a so-called "busy signal". Upon detecting this "busy signal" the receiving node will stay awake so the data packet can be received.

As illustrated by figure 2.2 this method of radio duty-cycling does not require any form of synchronisation between the nodes while the network is idle. Instead the sending and receiving nodes are synchronised when there is actual data to be transmitted. Moreover, the energy-wise cost of doing so is shifted to the sender which allows *idle listening* to be greatly reduced on the receiver. It should be noted however that, depending on the specific MAC protocol and sampling interval used, the "busy signal" that is prefixed onto the data packet can block the channel for a significant amount of time and that as a result of this low power listening MAC protocols are mostly suited for sensor networks that have extremely low throughput requirements.

The concept of low power listening was first proposed by [81] but, in contrast to later work, it is not used as a technique to conserve energy at the MAC layer of the network stack. Instead, it is used as a mechanism for waking up nodes after a long period of inactivity (once the nodes have been woken up, a regular MAC protocol takes over). In addition, they use a second radio to do the actual low power listening since otherwise the length of the used sample interval would cause the "busy signal" to interfere with normal data transmissions. The developers of WiseMAC [82] and B-MAC [83] were the first to integrate low power listening functionality directly into the MAC layer of the network stack. To do low power listening, both of these MAC protocols make use of the fact that when sending a packet over a wireless channel, this packet needs to be prepended by a so-called "preamble" to allow the radio chip of the receiving node to properly decode the incoming packet. Since, as far as the receiver is concerned, this preamble can be of any size both MAC protocols generate a busy signal by transmitting a preamble that is longer than the sampling interval.

Although both WiseMAC and B-MAC use CSMA to control access to the channel there are some differences between these MAC protocols. WiseMAC is specifically designed for single-hop infrastructured networks. It assumes all sensor nodes to be within range of a mains-powered basestation and as a result only uses low power listening for packets sent from the basestation to the sensor nodes. Packets sent by the sensor nodes are sent without a "busy signal" since the basestation always keeps the radio enabled. In addition, WiseMAC includes a number of features designed to minimise the overhead of the busy signal. The basestation will, for instance, keep track of when each sensor node was last woken up in order to predict when it will sample the channel for activity. This

information is then used to minimise the length of the preamble required to wakeup the sensor node by only starting the transmission right before the sensor node is predicted to poll the channel. In contrast, B-MAC is primarily intended to be used in multihop sensor networks with a flat topology and, given the resource constrained nature of the typical sensor node, is designed to be as simple as possible. Because of this B-MAC does not contain any special features apart from a clever CCA-algorithm designed to filter out false positives.

Since then a number of other Low Power Listening MAC protocols have been proposed that all try to enhance the earlier MAC protocols in various ways. X-MAC [59] for instance, is designed to minimise the delay and energy-wise overhead resulting from the long preambles used by WiseMAC and B-MAC. This is done by replacing them with a burst of short "probe packets" containing the address of the receiver. This reduces idle listening and overhearing by allowing nodes that are not the intended recipient to go back to sleep immediately after overhearing as probe packet addressed to another node. As discussed in [59], these probe packets are moreover sent in a "strobed" manner. In essence, the sender inserts a short pause between consecutive probe packets to allow the receiver to acknowledge the reception of the probe packet to the sender which in turn allows the stream of probe packets to be interrupted in case the receiver wakes up earlier than expected. While this reduces both the time and energy required to transmit (and receive) the packet it should be noted that this mechanism does require the receiver to keep its radio on for a longer period of time when sampling the channel for activity. Otherwise, incoming packets may be missed due to the receiver sampling the channel in between two probe packets. Another advantage of X-MAC is that it is compatible with a wider range of low-power radios since it does not require the radio to be capable of sending preambles of indeterminate length (a feature which is missing in most IEEE 802.5.4-compliant packetizing radios [59]). Like X-MAC, MaxMAC [84] also embeds the address of the receiver in the "busy signal". Instead of sending probe packets however, the authors propose to embed the destination address directly into the preamble of the data packet itself. To minimise the length of the "busy signal", MaxMAC reuses the preamble-minimisation technique employed by WiseMAC. In addition, the authors propose to dynamically tune the length of the sampling period to the traffic load based on a number of thresholds.

The TinyOS and Contiki sensor operating systems also include a low power listening MAC protocol. The low power listening MAC protocol included in TinyOS[3] borrows a number of ideas from X-MAC but uses data rather than probe packets to wake up the receiver. To transmit a packet to the receiver, the sender will first attempt to send it using CSMA/CA. If no acknowledgement is received, it assumes that the receiving node is in sleep mode and will then continue to transmit the same packet over and over in a so called "packet-train" until it either receives an acknowledgement or a timeout occurs. ContikiMAC [85] (the low power listening MAC protocol included with Contiki) operated in a similar fashion except that, like WiseMAC, it keeps track of when its neighbours were last active in order to minimise the number of packets needed to wake up the receiver.

[3] `https://github.com/tinyos/tinyos-release/blob/tinyos-2_1_2/tos/chips/cc2420/lpl/DefaultLplP.nc`

2.1.4 TDMA Based Protocols

Time Division Multiple Access (TDMA) MAC protocols organise access to the channel by dividing the channel into multiple *time slots* and assigning each node with specific time slots for packet transmission and reception. Each time slot is usually long enough to allow a single full-sized packet to be transmitted. In addition, some MAC protocols also allow an acknowledgement to be transmitted within the same time slot as well. Moreover, these time slots are usually organised into repeating fixed-sized *(super)-frames* but this once again depends on the specific MAC protocol used.

One major advantage of using TDMA in low power sensor networks is that it allows both idle listening and overhearing to be (almost) completely eliminated. Nodes only need to keep their radio enabled during time slots in which the node either transmits a packet itself or is scheduled to receive a packet from another node. During all other time slots the radio can be disabled. In addition, TDMA has built-in support for *Quality-of-Service* since the TDMA slot allocation can be calculated according to specific throughput and delay requirements imposed on the network. That being said, the use of TDMA in low power sensor networks also brings with it a number of challenges. Firstly, using TDMA requires nodes to agree on the exact time that each slot begins and therefore requires precise (usually sub-millisecond) time synchronisation between the nodes of the network. Secondly, the nodes also need to agree on the slot allocation that is used to transmit packets between different nodes. While both these problems are relatively simple to solve for single-hop networks, such as for instance cellular networks, it is significantly more difficult to do so for a (large-scale) multi-hop network consisting almost entirely of resource constrained nodes.

Over the years, both these issues have been extensively investigated for a wide range of scenarios but most researchers focus either on the synchronisation or on the slot allocation problem. Perhaps the simplest method of synchronising the clock of the different nodes is to build a so-called 'synchronisation tree' in which each node synchronises its clock to that of its parent node by adjusting its clock to the exact time at which it receives 'sync' messages that are periodically broadcasted by said parent. As noted by [86] however, this method of synchronisation causes the synchronisation-error to accumulate with the number of hops to the root node of the tree. As a result, it is only suited for applications where the data flows along the same paths as those used for synchronisation and even then problems may arise when neighbouring nodes have a different synchronisation-path to the root node. They therefore propose a synchronisation-mechanism that uses pulses (rather than packets) to synchronise the clocks of the different nodes, the idea being that each node regularly sends out a 'pulse' to its neighbours but that the exact time at which this is done depends on how many pulses it receives from its neighbours. Given that during the transmission of a pulse other incoming pulses are ignored this causes all the clocks in the network to synchronise to one another. Although this mechanism scales very well, it does require dedicated hardware and as a result is not usable in most sensor network scenarios.

The authors of [87] propose a synchronisation mechanism whereby multiple receivers synchronise to one another by recording the exact time at which a 'sync'-packet is received from an independent sender in range of all receivers. By afterwards exchanging the (local) times at which the 'sync-'packet was received with one another, all receiving nodes

are able to calculate the offsets needed to synchronise the local clocks to one another. Both [88] and [89] use a tree-based approach for synchronising the nodes in the network and use a two-way handshake between parent and child to compensate for clock drift and propagation delay. [90] also use a tree-based approach but go to even greater lengths to minimise the synchronisation error. They embed multiple timestamps into a single 'sync'-packet to allow the receiver to correct not only for clock drift but also for jitter in the local clock of the sender. In addition, nodes use multiple 'reference points' (rather than a single synchronisation parent) to synchronise to the root node with only a minimal synchronisation error. [91] expand upon this by measuring the reception time of the 'sync'-packets with multiple clocks (sourced from the same oscillator), which allows the synchronisation error to be reduced even further.

In contrast to these tree-based approaches, [92] propose a synchronisation-mechanism that uses regular data packets to synchronise the clocks of the nodes. Instead of (or in addition to) relying on dedicated sync-packets for time synchronisation, nodes compare the expected reception-time of incoming packets with the actual time these packets were received in order to track the clock drift of their neighbours. Once the accumulated error exceeds a certain threshold, the node will adjust its clock to minimise the average clock drift of all its neighbours and will subsequently refuse to update its clock for a certain period of time to allow its neighbours to synchronise to the new schedule. Finally, [93] discuss a synchronisation mechanism that is similar to the one proposed in [92] but that uses dedicated 'sync' packets instead of relying on data packets to announce timing information. These sync packets are sent during a so-called 'sync period' during which all nodes stay awake to receive the 'sync' packets from their neighbours. In addition, every node is assigned its own time slot within this sync period to transmit its own sync packet. Every time a sync packet is received, the node records the clock drift of the sending node (w.r.t its local clock). At the end of the sync period, every node aggregates the different clock drifts into a single value that is then used to adjust the local clock. The authors of [93] propose various aggregation functions that can be used for this purpose ranging from a simple median calculation to ones using more advanced methods such as least squares optimisation and discrete time Kalman filters. Although this method of synchronisation does incur the overhead of requiring every node to transmit timing information, it has the advantage of not imposing any topology on the network and of being more resilient to interference and packet-loss since every node effectively has multiple redundant clock-sources to synchronise to.

As with the time synchronisation issue, slot allocation for wireless (TDMA-based) sensor networks has also received widespread attention from researchers over the years. The main challenge here is that the slot allocation for a given node not only depends on the slot allocation of its immediate neighbours and that, in order to avoid collisions, the slot allocations of more distant nodes also need to be taken into account. As discussed in [94] the problem of slot allocation in multi-hop TDMA networks is related to the (minimal) graph colouring problem and it should therefore not come as a surprise that finding a minimal TDMA slot allocation is an NP-Hard problem [95, 96]. Complicating matters even further is that the optimal slot allocation not only depends on the network topology but also on the bandwidth requirements of the nodes and the paths along which packets are being forwarded. In addition, changes in the network topology (such as nodes joining, leaving and/or moving in the network) can have a global effect on the slot allocation of the

network. Because of this, most slot allocation mechanisms (and by extension TDMA MAC protocols) work best if the network topology and/or routes in the network are (relatively) stable. While some researchers focus solely on the problem of slot allocation [94, 97, 63], others propose more complete TDMA MAC protocols but even then the issue of time synchronisation is often ignored.

To counter the complexity of multi-hop slot allocation, a number of researchers have proposed MAC protocols that divide the network into multiple single-hop *clusters* of nodes. A single node within each cluster is selected as cluster-head and is responsible for assigning slots to the other nodes in the cluster. The earliest of these clustering MAC protocols is probably LEACH [65], which operates under the assumption that every node in the network can reach the basestation (in a single hop) but that it is takes a lot of energy to do so. Nodes therefore report their sensor readings to a local cluster-head which then aggregates the data and forwards it to the basestation. Cluster-heads are selected more-or-less randomly and are responsible for assigning time slots to the other nodes in the cluster. To prevent the cluster-heads from depleting their energy reserves ahead of time the clustering process is restarted periodically. BMA [98] expands on LEACH by, during each superframe, only assigning slots to nodes that have data to send, which allows both the cluster-heads and the other nodes to turn off their radio during idle periods. PACT [66] also uses clustering to decide the slot allocation. Cluster-heads are selected based on remaining battery-capacity and unlike leach, PACT does not assume all nodes to be within range of the basestation. Instead PACT uses so-called 'gateway-nodes' to link different clusters together.

Other TDMA MAC protocols organise the network along a tree-structure and use either centralised or distributed slot allocation mechanisms. [63] and [17] for instance propose centralised mechanisms for determining the slot allocation. [63] focusses solely on the slot allocation problem itself and describes a slot allocation algorithm that uses particle-swarm optimisation techniques to trade-off between energy usage and delay. [17] on the other hand proposes a mechanism whereby the basestation controls both the routes and the slot allocation used in the network. Routes are regularly adjusted based on the battery-status of the nodes and the slot allocation is calculated from the routing table using either bread-first or depth-first graph traversal techniques. The MAC protocol proposed by [62], 'FlexiMAC', is similar to the one proposed by [17] in the sense that it also integrates route selection into the MAC protocol and that the slot allocation is built using a depth-first traversal of the routing tree. In contrast to [17] however, FlexiMAC does so in a distributed manner. [99] propose a tree-based slot allocation mechanism that uses Egyptian fractions to ensure that slots are allocated evenly in the super frame and that existing slot allocations are not affected by the addition of new nodes. [100] also proposes a TDMA MAC protocol for (small) low power wireless networks but rather than adhering to a fixed slot-allocation, nodes instead determine (and announce) their own slot allocation based on the slot allocations overheard from their parent-and child-nodes. Finally, the developers of 'SS-TDMA' [101, 64] impose an even more rigid structure on the network and require nodes to be deployed along a rectangular grid. This allows each node to calculate its slot allocation purely from its (physical) position in the network.

There are also a number of TDMA MAC protocols that are designed for a 'flat' network topology and that therefore do not impose any structure on the network at all. L-MAC [102] for instance uses a node-centric approach to slot allocation (meaning that

time slots are assigned to nodes rather than to links between nodes) in which nodes periodically announce the time slots in use by themselves and their neighbours. This allows (new) nodes to learn which time slots are in use before claiming their own slot. [103] in contrast propose a MAC protocol that uses a link-centric approach to slot allocation. The general idea is that once two neighbouring nodes discover one another they immediately choose, a pair of time slots for packet transmission over the link between them (regardless of whether that link will actually be used). To prevent time slots of neighbouring links from overlapping with one another, frequency division multiplexing is applied by having the nodes select a random channel for communication over the link between them. (This of course assumes that there is a lot of bandwidth available.) Instead of having nodes 'randomly' select a time slot, NAMA [97] and TRAMA [104] use an 'implicit' form of slot reservation whereby each node calculates its assigned slots purely from a list of other 'contenders' (i.e., the two-hop neighbourhood). To decide which node takes ownership of a particular slot every node calculates its 'priority' for that slot as a hash of both its node-id and the slot-id, with the slot being assigned to the node with the highest priority. Given that every node knows who the other contenders are, it is able to also calculate the priorities of the other nodes and thus determine which slots are assigned to who without exchanging any control message with its surrounding nodes. This of course assumes nodes to know who the other contenders are, which is not always the case. Finally [105] tries to enhance this slot allocation mechanism by measuring which nodes will actually interfere with one another to build the 'contender-list' rather than relying solely on (two-hop) neighbour lists.

2.1.5 Hybrid and Other MAC protocols

Not all sensor network MAC protocols fall neatly into one of the four categories discussed above. There are also a number of 'Hybrid' MAC protocols that combine multiple duty cycling and channel access mechanisms. A number of the TDMA MAC protocols discussed above for example use CSMA for sending control traffic during a so-called 'contention period' [94, 104] (data packets are still sent using TDMA) or use CSMA to exchange neighbour information during network startup [62]. Z-MAC [74] integrates CSMA even more tightly with TDMA by allowing nodes to 'steal' time slots assigned to other nodes by performing CSMA in that time slot if they detect that it is not being used by the owner. Owners of a time slot must perform a random backoff within a fixed time period and perform a CCA before actually transmitting their packet. Non-owners of a time slot must wait at least the length of owners' backoff time period before (trying) to send their packet in the time slot using CSMA. (The maximum length of the owners backoff timer period is the same for all nodes and set at compile time.) In addition, owners of a time slot can transmit a so-called 'explicit congestion notification' to instruct their (two-hop) neighbours to vacate the time slot entirely. By doing so Z-MAC behaves more-or-less like slotted CSMA under low traffic conditions (which has a positive effect on delay) but behaves more like TDMA when the traffic rate (and thus the contention) in the network increases. In contrast to Z-MAC, SCP-MAC [106] combines techniques of low power listening and scheduled MAC protocols. In SCP-MAC all nodes will regularly poll the channel for activity just like any other low power listening MAC protocol. In contrast to other low power listening MAC protocols however, all nodes poll the channel at exactly the same time. While this does require nodes to maintain clock synchronisation with

one another it has the advantage that every node knows exactly when its neighbours will poll the channel and that the length of the 'wakeup'-signal can therefore be significantly reduced. In addition nodes perform a two-step contention procedure when accessing the channel (one to send the wakeup-signal, one to send the actual data packet itself) to reduce packet collisions and will poll the channel more frequently (for a limited time) after receiving a packet to allow it to adapt to variations in the traffic load.

Other researchers take a different approach and apply machine learning techniques to the problem of efficient channel access in sensor networks. [107] for instance present a 'Reconfigurable MAC Architecture' that allows multiple MAC protocols to be implemented on a single sensor node without having too large an impact on the code or memory footprint. They couple this with a "MAC Selection Engine" that is able to switch the MAC protocol used in the entire network depending on varying traffic requirements and the current level of external noise experienced by the nodes. The researchers propose to use a supervised learning approach that builds a decision tree that is then used to determine the best MAC protocol to use for a given set of requirements (energy, traffic) and environmental factors (noise). [108] on the other hand embed Q-Learning into the design of the MAC protocol. Q-Learning is a machine learning technique that allows an 'actor' (or node in this case) to autonomously learn from the environment which action yields the best long-term reward in a given situation (see [109] for a more in-depth explanation). [108] use slotted ALOHA as a basis for their MAC protocol, but rather than having nodes select their transmission slot randomly, Q-Learning is used to learn which transmission slots (in a given superframe) yield the best packet delivery ratio. Since every node applies the same technique for slot selection the idea is that this eventually causes the entire network to select those slots that best avoid packet collisions. Since Q-Learning continues to be used after the initial learning phase a node can switch is 'preferred slots' over time. In addition, each node will regularly announce its 'preferred slots' to its neighbours, which allows them to turn off their radio during the remaining slots.

2.1.6 IEEE 802.15.4

One cannot give an overview of sensor network MAC protocols without also discussing the IEEE 802.15.4 standard. IEEE 802.15.4 specifies a PHY- and MAC-Layer for communication in "*Low-Rate Wireless Personal Area Networks*" and is generally regarded as the counterpart of IEEE 802.11 for sensor networks. That being said, the early versions [10, 110, 111] of IEEE 802.15.4 have not been as well adopted as IEEE 802.11. This can partly be attributed to the fact that this standard is quite complex and therefore difficult to implement on low power sensor nodes. In addition, the early versions of IEEE 802.15.4 were specifically designed for infrastructured (planned) data gathering networks and therefore lacked some of the features that are important to other types of networks (e.g., support for node mobility, support for variations in the traffic load, ...). As demonstrated by the wide range of MAC protocols discussed above, this has caused many sensor network developers to use an alternative MAC protocol on top of the PHY-layer of IEEE 802.15.4. Since then the IEEE 802.15.4 standard has been revised a number of times and the most recent [112] version of the standard not only includes a much wider selection of PHY-layers to choose from but also adopts a number of the mechanisms introduced by the MAC protocols discussed above.

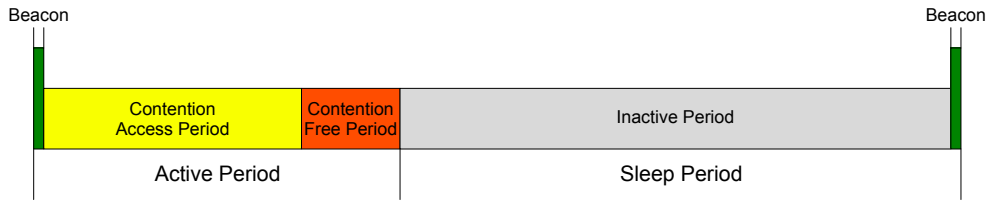


Figure 2.3: IEEE 802.15.4-2003 superframe structure.

The MAC-Layer specified by the 2003-version of IEEE 802.15.4 organises nodes into multiple 'Personal Area Networks' (PANs). Each PAN is managed by a so-called PAN-controller which functions as a sort of cluster head. Access to the channel is governed by one of two modes of operation: beaconed or non-beaconed mode. In non-beaconed mode nodes essentially access the channel using CSMA/CA and since this mode does not support any form of duty cycling it also means that nodes must keep their radio enabled at all times. In beaconed mode the channel is divided into multiple superframes. As shown in figure 2.3, the start of each superframe is signalled by a beacon-frame sent by the PAN-coordinator and the superframe itself consists of a 'Contention Access Period' (CAP), 'Contention Free Period' (CFP) and an 'Inactive Period'. During the CAP, nodes access the channel using slotted CSMA. During the CFP, TDMA is used to transmit packets (slots are assigned by the PAN-coordinator) and during the inactive period all nodes turn their radio off. The length of each of these periods is decided by the network administrator and is announced in the beacon-frame broadcasted by the PAN-coordinator at the beginning of each superframe. The duty cycling and channel access mechanisms used by the beaconed mode of IEEE 802.15.4-2003 are thus quite similar to the ones used by S-MAC, except that IEEE 802.15.4 adds a limited number of TDMA slots at the end of the Contention Access Period.

One important limitation of IEEE 802.15.4-2003 is that its beaconed mode can only be used in single-hop cluster PANs. This is because the beacons at the beginning of the active period can only be sent by the PAN-coordinator. To address this issue, the 2006-version of IEEE 802.15.4 expands the single-hop cluster of IEEE 802.15.4-2003 to a hierarchical cluster. To do so IEEE 802.15.4-2006 distinguishes between regular 'coordinator' nodes and the (single) PAN coordinator. Like the PAN-coordinator, regular coordinators periodically broadcast beacons to announce the start of the active period to the nodes in their (sub)cluster but they also synchronise to the active period advertised by their parent coordinator. To prevent multiple active periods from interfering with one another, each coordinator broadcasts its beacon at a different offset w.r.t the begin of the superframe. This means that a single superframe can contain multiple active periods and, like nodes using S-MAC, coordinator-nodes in IEEE 802.15.4 essentially have to synchronise to multiple schedules.

The IEEE 802.15.4e-extension [113], which has since been incorporated into IEEE 802.15.4-2015 [112], makes IEEE 802.15.4 more useable for a wide range of use cases by extending the capabilities of the existing (beaconed and non-beaconed) modes of operation as well as by adding a number of new modes. One extension to the existing modes is for instance support for low power listening during the CAP of the superframe. The mechanisms involved are referred to as "coordinated sample listening" in the standard and

are quite similar to the ones used by the X-MAC protocol discussed above, except that wakeup frames cannot be acknowledged by the receiver. Other extensions include techniques to limit the number of beacons that need to be sent and techniques to enable interoperability between beacon-enabled and non-beacon-enabled PANs. IEEE 802.15.4e also specifies a number of new modes of operation. In "*Low Latency Deterministic Network (LLDN)*" mode, the PAN behaves effectively as a single-cluster, single-hop TDMA network. In LLDN mode, the superframe does not contain a CAP and the PAN-coordinator is responsible for assigning upstream (or downstream) slots to nodes in the PAN. The "*Deterministic and Synchronous Multi-channel Extension (DSME)*" mode of IEEE 802.15.4 in essence extends the CFP to the end of the superframe and to multiple channels, the general idea being that nodes can exchange management-packets during the CAP to allocate and deallocate dedicated timeslots for communication during the CFP.

Finally, the "*Timeslotted Channel Hopping (TSCH)*" mode specifies an FTDMA structure that spreads over multiple channels and in which a single time slot is uniquely identified by an ever increasing "absolute slot number" and a "channel offset". As the name suggests, channel hopping is used to limit the impact of outside interference on the communication. Instead of using a superframe structure, TSCH uses repeating "*slotframes*" which are configured by the higher layers of the network stack. In contrast to superframes, slotframes are not necessarily the same for all nodes in the network. Multiple slotframes can co-exist in the same network and not every node has to subscribe to the same set of slotframes. Each slotframe always contains a number of "*shared slots*" to which all nodes must listen and that are used mainly for signalling and broadcast communication. All other time slots can be used for dedicated (unicast) communication between nodes but, as with the slotframes, this must be configured by the higher layers of the network stack.

While the current version of IEEE 802.15.4 thus contains a number of different channel-access and duty-cycling mechanisms that can be used for a wide selection of sensor network use cases, it is *not* a "one-mac-protocol-fits-all" solution for communication in low power sensor networks. The fact that it contains so many different 'modes of operation' illustrates once again that the requirements of the various sensor network use cases are simply too diverse to allow them to be met by a single MAC protocol. Sensor network developers must therefore choose the mode that best fits their use case and even then there are a wide range of parameters to be tuned to the requirements of the application (e.g., active period / sleep period times, whether or not to use low power listening, length of a single time slot, ...). Depending on the mode used, IEEE 802.15.4 also delegates a number of use-case specific issues to higher layers in the network stack. The TSCH mode of IEEE 802.15.4 is a fine example of this: IEEE 802.15.4 specifies the general framework for organising node-to-node communication but is up to the developer of the higher layers in the network stack to decide which slotframe and time slot allocation to use. Since the introduction of IEEE 802.15.4e, a number of frameworks [114, 115] have been proposed for coordinating slotframe and timeslot selection in the network but even then the problem of deciding which slot allocation to use continues to be an active area of research [116, 117, 118, 119].

2.1.7 MAC Protocols considered in this thesis

Given the wide selection of sensor network MAC protocols currently available, it is not possible to consider all of them in this work. Therefore, a single representative MAC protocol is selected from each of the categories discussed in sections 2.1.1 to 2.1.4. IEEE 802.15.4 is (apart from its CSMA/CA mechanism) not directly considered since, as discussed in section 2.1.6, most of the channel access and duty cycling mechanisms described in this standard are based on those discussed in sections 2.1.1 to 2.1.4.

The CSMA/CA MAC protocol considered here is the one that is specified for the non-beaconed mode of IEEE 802.15.4-2003. This means that carrier sensing (with exponential back-off) acknowledgements and retransmissions are used to transfer data from one node to another but that the RTS/CTS mechanism is not used. The reason for using it here is that, although the rest of the standard is not always implemented, the non-beaconed mode can be regarded as the *de facto* standard CSMA/CA protocol for low power sensor networks and serves, for instance, as the basis for the CSMA/CA MAC protocols included in the popular TinyOS and Contiki sensor network operating systems.

The low power listening MAC protocol used here is the 'LPL-MAC' protocol included with TinyOS version 2.1.0. The main reason for using this specific protocol is that TinyOS is one of the first truly popular sensor network operating systems and this MAC protocol has therefore been widely used in sensor network research. Since TinyOS treats low power listening as an 'extension' built on top of CSMA/CA the same approach is used here. This means that, as discussed in section 2.1.3, nodes are woken up by transmitting a 'packet train' of copies of the same data packet but that each packet is transmitted using CSMA/CA. Similar to X-MAC, this allows the receiving node to acknowledge the data packet before the end of the packet train.

T-MAC has been chosen to represent the scheduled MAC protocols. The reason for doing so is that, similarly to TinyOS, T-MAC is one of the first truly popular scheduled sensor network MAC protocols. While this is also the case for S-MAC, the fact that this protocol is unable to cope with varying traffic loads makes it an ill candidate for the application scenarios considered here (see section 2.2.2). The T-MAC implementation used in this work is based on the protocol specified in [76] but there are some differences between the two. Firstly, the "Future-request-to-send" mechanism is not used. This is because this mechanism operates on the assumption that packet collisions only affect nodes in the one-hop neighbourhood of the transmitters. While this may be the case for the radio model used by [76] (which assumes nodes have a fixed transmission distance), this is not the case for the more realistic channel model of the simulator used in this thesis (see section 2.2). Given that [76] themselves omit this mechanism from the T-MAC implementation they use to evaluate the protocol on real hardware, this mechanism is also not used here. In addition, their 'real hardware' implementation also does not contain the "multiple schedules" mechanism and therefore it is also omitted from the T-MAC implementation used in this work. To keep the clocks of the nodes synchronised, the 'real hardware' implementation specified in [76] uses a basic clock-drift correction mechanism whereby nodes synchronise by adjusting their schedule to the average of their own schedule and that of the received schedule every time they receive a sync-packet. While this mechanism worked well for [76] it should be noted that their test setup only contained two nodes. To make the synchronisation mechanism more robust for larger networks, the T-MAC

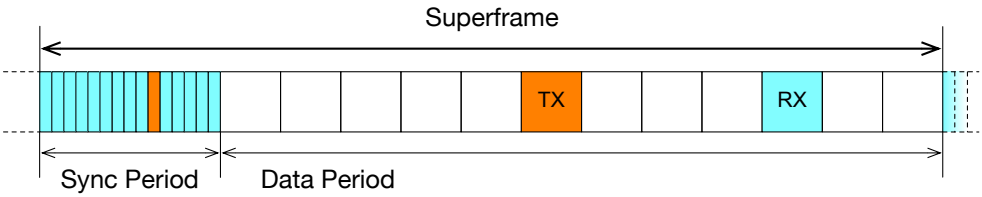


Figure 2.4: Frame structure of the TDMA MAC protocol considered in this thesis.

implementation used here expands on this basic mechanism by using the 'median drift-correction' technique discussed in [93]. Instead of immediately adjusting its schedule after receiving a sync-packet, each node will record the clock drift between the sending node and itself. Synchronisation is only performed when the node is due to transmit a sync-packet itself. At that point the node will adjust its clock so the median of all clock drifts observed since the transmission of the previous sync packet is minimised. A final difference between the T-MAC protocol discussed in [76] and the T-MAC implementation used here, is that the one used here does not use the RTS/CTS mechanism. This is done to avoid the enormous overhead (see [74]) this mechanism incurs.

In contrast to the previous three categories of MAC protocols, finding a suitable representative TDMA MAC protocol proved to be more difficult than initially anticipated. As discussed in section 2.1.4, the synchronisation and slot allocation aspect of TDMA have mostly been investigated separately. In addition, most of these impose a specific structure on the topology of the network whereas the use cases considered in this work require the MAC protocol to also support flat topologies. The few "full-fledged" TDMA MAC protocols that don't impose a specific network topology either assume the availability of enormous amounts of bandwidth [103] or use contention in some part of the protocol [104, 105] and are therefore not 'pure' TDMA MAC protocols. For this reason, the TDMA MAC protocol considered in this thesis is a custom-built one that combines the versatile and straightforward synchronisation mechanism discussed in [93] with a simple slot allocation mechanism that allocates slots to nodes and links based on the routing paths used in the network.

The superframe structure of the resulting TDMA MAC protocol is shown in figure 2.4. Each superframe is divided in a 'sync period' and a 'data period'. During the 'sync period', nodes exchange timing information by transmitting a sync packet during their assigned time slot and by listening for sync packets received from other nodes. For simplicity's sake, both the assigned 'sync slot' and the length of the 'sync period' are assumed to be programmed into the nodes prior to deployment. This is not an unreasonable requirement given that TDMA MAC protocols are mainly used in infrastructured (i.e., planned) networks. When the synchronisation period ends, the nodes then adjust their clock so the median clock drift is minimised. During the 'data period' nodes exchange packets according to their assigned data slots. As illustrated by figure 2.4, 'data slots' are larger than 'sync slots'. This is because 'sync slots' only need to be long enough to allow a small, 13 byte, sync-packet to be transmitted whereas data packets can be as large as 127 bytes (the maximum packet length allowed by IEEE 802.15.4-2006). It should also be noted at this point that data packets are *not* acknowledged by the receiver of the packet. The reason for omitting the acknowledgement-mechanism from the MAC protocol is that this

mechanism is also not considered by most of the MAC protocols discussed in section 2.1.4. Every slot in the 'data period' of the superframe can be one of three types: 'sleep', 'rx' or 'tx'. As the name suggests the TDMA MAC protocol keeps the radio disabled during sleep-slots. At the beginning of an rx-slot the radio is enabled to allow an incoming packet to be received. To reduce idle listening, the radio is disabled again if no activity is detected on the channel after a certain timeout. Otherwise, the radio remains enabled either until the PHY-Layer reports that a packet has been received or until the slot ends. Tx-slots are used to transmit packets to other nodes. Given that access to the channel is governed using *only* TDMA, nodes do not contend for channel access and as a result packets are transmitted *without* performing a CCA-check beforehand.

As discussed in section 2.1.4 there are many algorithms available to do TDMA slot allocation but most of these either rely on a simplified (graph based) model of the network or are designed for a specific use case. Given moreover that these can be quite complex and that TDMA slot allocation is *not* the primary focus of this thesis, the TDMA MAC protocol considered here instead uses a fairly straightforward, node-centric slot allocation mechanism. For simplicity a part of the slot allocation is decided statically (before deployment) and as a result, the number of nodes present in the network also needs to be known prior to deployment. While it would be possible to expand the slot allocation mechanism to support nodes joining/leaving the network (e.g., by performing a network-wide reconfiguration) this is outside of the scope of the work presented here.

The slot allocation mechanism differentiates between broadcast and multicast slots. Broadcast slots can be used by the owner of the slot to transmit either a unicast packet to any node within its range or for transmitting a broadcast packet to all neighbouring nodes, whereas multicast slots can only be used to transmit a unicast packet to any of the next-hop neighbours (the set of next-hop neighbours is decided based on the routes selected by the routing layer). When assigning slots to nodes, the slot allocation mechanism will never "reuse" time slots. This means that when a time slot is assigned to a particular node, it will not be assigned to another node even if the two should be out of each other's range based on their location. While this does increase the number of time slots required it also eliminates interference between TDMA nodes of the same network.

To allow the routing layer of the network to establish and maintain routes, each node is assigned a single broadcast slot per superframe. For simplicity's sake these slots are, like sync-slots, assigned prior to deployment. In addition, the topology of the network is not taken into account when doing so. This, in essence means that when a node is assigned a specific broadcast-slot, a corresponding rx-slot is assigned to the remaining nodes in the network regardless of whether they are within range of slot owner or not. While this may not be the most efficient (broadcast) slot assignment possible it does provide the most flexibility to the routing layer to establish and maintain routes. In addition, the energy-wise overhead of doing so should be minimal since each node is only assigned a single broadcast slot and nodes turn their radio off prematurely during rx-slots if no activity is detected.

Multicast slots are assigned in a 'semi-static' fashion. Prior to deployment each node is assigned a fixed number (3 in this work) of multicast slots, but with an empty "next-hop neighbour" list. At network startup, each node is thus aware of which multicast slots are assigned to which nodes but since the "next-hop neighbour" lists are still empty at

that time, no 'rx-slots' are scheduled yet. For the same reason the scheduled 'tx-slots' remain unused. As routing paths are established by the routing protocol, the routing information received from the routing layer is used to update the "next-hop neighbour" lists and to schedule the 'rx-slots' needed to allow data packets to be forwarded along the established paths. The route selection mechanism used in conjunction with this TDMA MAC protocol is discussed in section 2.2.1.

2.2 Simulator & Test Setup

The tests discussed in this chapter, as well as the work presented in chapters 4 to 6 required a suitable sensor network simulator to be chosen. Given that inter-node interference is expected to be a significant factor in the considered scenarios, the selected simulator must be able to take both path loss and interference between concurrent transmissions into account. For this reason, all simulations discussed in this thesis are performed using version 3.2 of the Castalia [120, 121] simulator. Castalia is a sensor- and body area network simulator based on the Omnet++[122] discrete time event framework. As discussed in [120] "*Castalia features an accurate channel/radio model ... and forces the user to deal with many of the unpleasant -but yet important- aspects of communication.*". Links between nodes do not use fixed packet reception probabilities. Instead "*...these probabilities are calculated on the fly based on the transmission power of* ***all*** *transmitting nodes* [emphasis added] ." "*From* [this] *modelling, features such as non-unit-disk coverage, non-symmetrical links, and probabilistic nature of packet reception emerge*"[121]. In addition, Castalia also simulates node clock drift. This is an important feature given that both TDMA and scheduled MAC protocols rely on clock synchronisation between the nodes for channel access and/or duty cycling.

It should be noted at this point that, despite these advanced features, Castalia is still a simulator and thus only a *model* for how sensor nodes behave in real-life. As with all simulation-based studies, the validity of the obtained results thus depends on the accuracy of this model and consequently also on the assumptions made by the simulator. This means that when a simulator is used to predict how a sensor network will behave in a particular environment (e.g., to plan the deployment of new sensor nodes) it becomes necessary to *calibrate* the simulator to the specific hardware and environment being investigated. This work however does not target any specific hardware platform or any specific deployment environment. Instead, it focusses on investigating the behaviour of co-located MAC-heterogeneous sensor networks *in general*. Moreover, calibrating wireless network simulators to a real-world environments is a research-area in itself and thus not within the scope of this thesis. Because of this, Castalia's channel- and radio-model are used *as-is*, without any further calibration. While the performance results obtained using this simulator thus don't necessarily match those of a specific real-life test-setup, it should be noted that both [123] and [121] already have done extensive work to validate both the channel model and the Castalia simulator itself. Despite the lack of further calibration, the simulator should thus be realistic and advanced enough to allow the *general behaviour* of co-located MAC-heterogeneous sensor networks to be investigated.

During the course of this PhD, the channel- and radio-model of the Castalia simulator have remained more-or-less unaltered but the rest of the simulator has, inevitably, undergone a number of modifications. These modifications have happened gradually as the

research progressed and the requirements evolved. As a result, the current version of the simulator is not the same as the one used to do the initial 'heterogeneous MAC'-experiments discussed in [124]. To ensure that the results discussed in the different chapters of this thesis are all based on the same simulation environment, the early experiments have therefore been rerun using the version of the simulator discussed in this section. Apart from fixing a bug in the "TimerService"-module, only the network stack embedded in the Castalia simulator has been altered. For this reason, the network stack of the simulator, as well as the changes made to it, are discussed in more detail in section 2.2.1. Section 2.2.2 discusses the considered application scenarios while section 2.2.3 details the test setup used to investigate the effect of interference between MAC-heterogeneous sensor networks.

2.2.1 Network Stack

In the Castalia simulator, the physical layer of the network stack is comprised of two components: the "Radio" module which implements the radio chip functionality of a sensor node and the "WirelessChannel" module which simulates the transmission of packets between the Radios of the different nodes. The channel model embedded in the "WirelessChannel" module[4] does not consider packet transmissions directly. Instead it models every packet transmission as an electromagnetic signal that is generated by the transmitting node and that is received, with a varying signal strength, by every other node in the wireless environment. Path loss between the transmitter and receiver is simulated using a log-normal path loss propagation model that considers both shadowing and non-uniformity in bidirectional links. Since multiple nodes can transmit at the same time, these signals can overlap and as a result a node can receive multiple signals at the same time. Naturally, at most one of these can be decoded as a valid packet by the radio chip of the receiver. To model this, the "Radio" module of the receiving node keeps track of the signal strength, duration and start time of all incoming signals. This information is then used to calculate the signal-to-noise ratio (SNR) of the signal that is currently being decoded. Based on the SNR and the modulation of the signal, the bit-error-ratio (BER) is calculated which in turn is used to probabilistically determine the number of bit-errors in the received packet. While doing so, the "Radio" module accounts for the fact that the SNR, and thus the BER of the signal being decoded, can change over time as a result of signals ending or new signals being added. Once the packet has been 'decoded', it is either dropped or passed to the higher layers of the network stack depending on how many bit-errors were encountered (and how many bit-errors are allowed by the modulation used).

While working on this thesis, the CCA-threshold was increased from -82dBm to -77dBm to match the default value specified in the specification of the CC2420 radio chip (which is used for the tests discussed in chapter 3), but apart from that no changes were made to the physical layer of the simulator.

In contrast to the physical layer, both the MAC- and routing-layer of the network stack have been extensively modified. Firstly, the "HAL"-architecture to be introduced in chapter 3 was implemented in between the physical layer and the MAC-layer of the network

[4]Castalia 3.2 Wireless Channel Model: `https://github.com/boulis/Castalia/blob/3.2/Castalia/src/wirelessChannel/WirelessChannel.cc`

stack to allow multiple MAC protocols to be used at the same time. This functionality is not further discussed here given that it is fully explained in chapter 3 and that it is not relevant for the tests discussed in this chapter. At the MAC-layer of the network stack, the MAC protocols provided by Castalia have been replaced with the ones discussed in section 2.1.7. Although Castalia already provides a "TunableMAC"-protocol that could have been used as a basis for both the CSMA/CA and LPL-MAC protocols, "*... this protocol was built with broadcast communication in mind...*"[125] and therefore does not provide support for unicast communication. As a result, is was easier to implement the CSMA/CA and LPL-MAC protocols from scratch than it would have been to rework the code of the "TunableMAC" protocol. Castalia also provides an implementation of the T-MAC protocol and given that this implementation closely matches the specification of the *full* T-MAC protocol given by [76], it was initially used to do inter-MAC interference experiments. Since then it has become apparent that when this (full) T-MAC protocol is used in combination with the more realistic channel model of Castalia rather than with the "fixed transmission distance"-model used by [76], it becomes highly unstable for the network sizes and density considered in this thesis (even without outside interference). To alleviate this issue, the modifications discussed in section 2.1.7 were made to the T-MAC implementation. Finally, the TDMA MAC protocol discussed in section 2.1.7 was implemented from scratch as Castalia does not provide a TDMA MAC protocol of its own.

At the routing-layer of the network stack Castalia only provides a single routing protocol that forwards packets by, essentially, broadcasting them in the right direction (some filtering is done to ensure that nodes do not re-broadcast packets received from nodes closer to the destination). Although this is a very simple and robust routing protocol, it is also extremely wasteful in terms of energy consumption and channel utilisation. In addition, the developers themselves state that "*... if the traffic passes a certain low threshold, congestion can kill performance*"[125]. Given that this routing mechanism is thus not usable for the work presented here, a new routing mechanism along with a suitable routing metric had to be chosen.

As a routing metric, the 'Expected Transmission Count' [126] (ETX) metric is used. As discussed in [126] the ETX is the expected number of (data) packets that need to be transmitted, including retransmissions, in order to successfully send a single (data) packet from the source to the destination node over a particular routing path. The main reason for using this metric here is that it naturally balances between reliability and hop count without requiring any tuning. It generally prefers high quality links but also allows lower reliability links to be used if needed. In addition this metric has been well adopted within the sensor network community as it is the default metric for route selection in both TinyOS and Contiki and it is the *recommended* [127] default routing metric for RPL [28] (the current IETF-standard for routing in low-power wireless networks).

Although this metric is very easy to calculate it does require the reliabilities of the various links in the network to be known. Given that signal-strength is a poor indicator for link reliability [128], the authors of [128, 129] discuss a number of alternative methods that can be used to estimate link reliability instead. Although the authors of [128] favour a "passive snooping" approach (because it has a low performance overhead), this method unfortunately requires the radio to be enabled at all times and is therefore not usable here. They also note that "passive probing" (i.e., counting the number of undamaged

packets received on a link) is well established in wired networks but cannot be used in wireless networks. To circumvent these issues, link reliability is therefore estimated using "active probing" instead. To do so, all nodes in the network will broadcast a 'probe' packet once every 30 seconds. (This value was selected as a trade-off between obtaining accurate link statistics and the energy-wise cost associated with doing so.) Since link reliability can vary significantly even when the network is stable [128], the "Window Mean with Exponential Weighted Moving Average" estimator proposed by [129] is used to convert the raw measurements into a more stable link reliability estimate. This estimator has two parameters t and α. t is the period between consecutive updates to the estimate, measured in number of probe reception opportunities, and α is the learning rate which determines how quickly the estimate adapts to variations in the link reliability. The link reliability estimate is updated as follows:

$$R_{cur} = \frac{P_{rx}}{t}$$
$$R_{new} = \alpha R_{old} + (1\text{-}\alpha)\, R_{cur}$$

In the above formula R_{cur} is the current reliability and is calculated from the number of received probe packets (P_{rx}) and the parameter t. R_{new} is the new value for the reliability estimate and is calculated from the current reliability, the α-parameter and the old reliability estimate (R_{old}). Within the scope of this work, parameter $t = 8$ (which means that the estimate is updated every 4 minutes) and $\alpha = 0.6$, which is the value used for the 'stable' configuration used in [129].

In contrast to the route metric, selecting the routing protocol itself proved to be more of a challenge. The main issue was that the use cases considered in this thesis (see section 2.2.2) require the network stack to be able to cope with both node-to-sink and node-to-node communication. At the time work on this thesis began however, most existing sensor network routing protocols either only considered node-to-sink communication [130, 131, 132] (e.g., cluster-based approaches) or were based on routing protocols for wireless ad-hoc networks and thus assumed [133, 134, 135, 136] the use of an always-on CSMA/CA-based MAC protocol. Moreover, even today the standard [28] protocol for routing in low-power wireless networks (RPL) is mostly designed for node-to-sink communication. (Node-to-node communication is supported but only by routing the packet over the nearest common ancestor in the routing-tree.) Given that the development of sensor network routing protocols is *not* the primary focus of this thesis, route selection is done periodically by a central controller rather than (on-the-fly) by the nodes themselves. Every 15 minutes, the central controller gathers the current link reliability statistics from the nodes in the network and then calculates the ETX value for every link in the network. Next, the routes themselves are calculated using Dijkstra's algorithm [137], after which the calculated routes are disseminated back to the nodes. For simplicity's sake, the actual communication between the controller and the nodes is done out-of-band (i.e., not through the wireless channel). While this, admittedly, causes the communication overhead of collecting and disseminating topology and route information to be neglected it is argued that, given the selected route update interval, this only represents a very small portion of all packet transmissions taking place in the wireless environment. Moreover, this simplification does not ignore the cost of determining the link reliabilities which, given the probe interval used, represents the majority of the communication overhead associated with route selection.

2.2.2 Application Scenarios

In this work two separate scenarios are considered: a 'node-to-sink' and a 'random-flows' scenario. From a software point of view, these scenarios differ only in the application running on the nodes in the network. The network stack, as well as all the considered (simulation) parameter values are exactly the same in both scenarios.

The 'node-to-sink' scenario mimics the typical "data gathering" use case in which sensors monitor the environment in which they are deployed and in which the main purpose of the network between these sensors is to deliver the performed measurements from the individual nodes to a central server for further processing. To emulate this behaviour, nodes periodically generate data packets that are then forwarded, over multiple hops, to a single 'sink' node deployed at the edge of the network. Data packets are only generated after a route to the sink has been obtained (i.e., after the first route update has been pushed by the central controller) and the (fixed) interval at which they are generated is one of the parameters considered in the inter-MAC interference tests discussed in this chapter (this interval is hereafter referred to as the "data generation interval"). The data packets themselves always have a payload of 100 bytes which means that the packets themselves are between 123 and 125 bytes large depending on the specific MAC protocol used.

The 'random-flows' scenario examines node-to-node rather than node-to-sink communication and is thus more relevant to the logistics and building automation use cases discussed in chapter 1. The node-to-node interactions considered in this scenario are based on those defined by CoAP [29] (Constrained Application Protocol), which is a "*...web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks*" (i.e., the equivalent of HTTP for low power networks). In the 'random-flows' scenario, nodes communicate by either querying one another for information (which matches the "Request/Response" interaction mode of CoAP) or will, at the request of the destination node, transmit a continuous 'stream' of information to a specific destination (which matches the CoAp "Observer" interaction model defined in [138]). As implied by the name of the scenario both the specific nodes with which each node communicates, as well as the method by which these interact, is selected more-or-less randomly during the course of the test run. As a result, the same is true for the actual traffic flows in the network resulting from these interactions. Similarly, the data rate of the transmitted streams and the time interval between subsequent queries is also decided randomly for each individual flow. As with the 'node-to-sink' scenario, data packets always have a payload of 100 bytes. In addition, the 'random-flows' scenario also considers the "data generation interval"-parameter but in contrast to the 'node-to-sink' scenario this parameter does not denote the (fixed) interval between the generation of two data packets. Instead, it is used to tune the various random number generators so the overall *expected* rate at which data packets are generated by the nodes, matches the specified interval.

When running the random-flows application, each node behaves both as a 'client' and a 'server' in the sense that it both sends requests (queries or 'streaming-requests') to other nodes and processes incoming requests received from other nodes. At the start of each test run, every node randomly selects a fixed number of 'server' nodes (4 within the scope of this work) it will interact with during the test run. Once routes to these nodes have been established, communication with the server nodes is initiated. A client node will

only interact with one server at the same time. The specific server to use is randomly selected from the list of server nodes and a new server node is selected every four to six minutes. (The exact interval between the selection of two server nodes is randomly selected from a normal distribution with a mean of 5 minutes, a standard deviation of 30 seconds and a 'cut-off' interval of 60 seconds on either side of the mean.)

Every time a new server node is selected, the client also randomly selects the 'transportation mode' used to retrieve data from the server. Information is either retrieved in 'query-mode' or in 'streaming-mode' and each mode is selected with equal probability. When in 'query-mode', the client will regularly send a 'query' message to the server to which the server responds with one or more response packets. These response packets are sent 'back-to-back' (without any delay) and the number packets to be sent is embedded in the query message received by the server. The number of response packets requested by the client varies between 1 and 4 and is selected randomly for each query. If, after sending a query message, no response is received within 20 seconds, the original query is retransmitted. Upon reception of the response sent by the server, the client node will wait for $(1 + numResponses) * PoissonRnd$ seconds before sending the next query (where $numResponses$ is the number of responses previously requested from the server and $PoissonRnd$ is randomly selected from a Poisson distribution with an average equal to the "data generation interval").

When a client enters 'streaming mode' after selecting a new server, it will send a 'stream setup' message containing both the requested data rate for the stream (specified as the time interval between the transmission of two consecutive packets) and the time period the stream should remain active (specified as the total number of packets to be transmitted). Upon receiving this message, the server will start sending 'stream data' packets to the client using the specified transmission interval until the requested number of packets have been sent. The transmission interval to use is randomly selected by the client from a list of 5 possible transmission intervals. These transmission intervals, as well as the probabilities for selecting these intervals, are shown in table 2.1. In this table, $dataGenerationInterval$ refers to the "data generation interval"-parameter considered by the 'random-flows' scenario. The intervals and probabilities shown, have been selected so most streams will use a transmission interval that is 'reasonably' close to the "data generation interval" but more divergent transmission intervals are also sometimes used. The number of packets specified in the 'stream setup' message is always chosen in such a

Probability	**Transmission Interval**
0.06135	$(1\text{-}\frac{3}{4})dataGenerationInterval$
0.2448	$(1\text{-}\frac{3}{8})dataGenerationInterval$
0.3877	$dataGenerationInterval$
0.2448	$(1 + \frac{3}{8})dataGenerationInterval$
0.06135	$(1 + \frac{3}{4})dataGenerationInterval$

Table 2.1: Transmissions intervals used for 'streaming' communication in the 'random-flows' scenario.

way that the server keeps sending 'stream data' packets until the client is due to select a

new server. If no 'stream data' packet is received within 20 seconds of transmitting the 'stream setup' message (the first 'stream data' packet is sent immediately by the server after the reception of a 'stream setup' message) the setup message is retransmitted. If after three retransmissions still no 'stream data' packet is received, interaction with the server is terminated prematurely and a new server is selected by the client.

2.2.3 Test Setup

To investigate the effects of inter-MAC interference, a wireless environment is simulated in which either one or two sensor networks are deployed. Simulations where only one network is present are used to establish the baseline network performance. When two networks are present in the wireless environment each network uses a different MAC protocol. In addition, the case where two heterogeneous (i.e. non-compatible) TDMA-based MAC protocols are used, is also considered. To investigate this case, both networks are equipped with the TDMA MAC protocol discussed in section 2.1.7, but the synchronisation mechanism and slot allocation used will not take the presence of the other network into account.

As discussed in section 2.2.2, both a 'node-to-sink' and a 'random-flows' scenario are considered in the investigation of inter-MAC interference. To investigate the influence of the network load, the "data generation interval"-parameter considered by these scenarios is varied between 10 and 60 seconds. In addition, two different 'network sizes' are considered: one where each network contains 25 nodes and one where each network contains 100 nodes. In both cases the nodes of each network are deployed along a 'randomized grid' pattern. This is similar to a normal (fixed) grid deployment except that a random variation is added to the position of each node. When 25 nodes are used per network, these nodes are deployed as a 5x5 random grid in an area measuring 50 by 50 meters. When 100 nodes are used per network, the nodes are deployed in an area measuring 100 by 100 meters and a 10x10 grid layout is used. In order to create interference between the two networks, the geographical areas in which these networks are deployed overlap. Moreover, the amount of overlap is varied in order to create several levels of interference. When there is 0% overlap, both networks are deployed adjacent to each other. In this case interference mostly occurs between the nodes near the shared edge of the networks. When there is 100% overlap, the deployment areas of the networks coincide and interference occurs between all nodes of both networks. These node deployments are illustrated in figure 2.5. It should be noted that the sink nodes shown in this figure are only relevant for the 'node-to-sink' scenario. In the 'random-flows' scenario these nodes behave just like any other node in the network.

For each combination of parameters, 100 independent test runs (each using different seeds for the random number generators of the Castalia Simulator) are performed and each test run lasts for a period of 4 simulated hours. At the end of each test run, the metrics used to measure networking performance are recorded. Three different metrics are considered: duty cycle (the percentage of time that the radio is active), end-to-end reliability and hop count. Duty cycle is measured by recording the amount of time the radio is in 'rx', 'tx' and 'sleep' mode for each node. Duty cycle is then calculated for each individual node as the ratio between the time that the radio was active (i.e., in 'rx' or 'tx' mode) and the total time. These different duty cycles are then averaged to calculate the duty cycle for

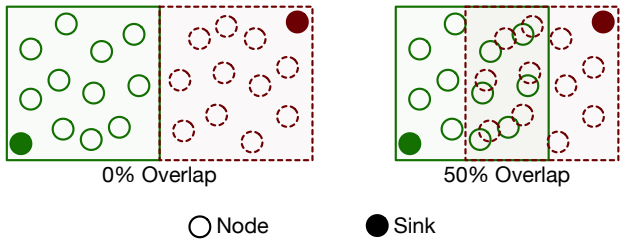


Figure 2.5: Illustration of the node deployments for 0% and 50% overlap.

the entire network.

The end-to-end reliability of the network is the percentage of application-level packets that are correctly transmitted through the network to their intended destination. This metric is recorded by keeping track of the total number of application-level packets sent and received by each node and then calculating the end-to-end reliability as the ratio of the total number of packets received versus the total number of packets sent. To measure hop count, the path used to forward each packet is systematically added as 'meta-information' (i.e., this does not alter the packet size) to the packet as it travels through the network. This information is then recorded by the destination upon reception of the packet. At the end of the test run, the hop count of the network is subsequently calculated as the average length of the routing paths used to forward the individual packets. The results of all these tests are discussed in the remainder of this chapter.

2.3 Interference in the node-to-sink scenario

The tests performed for the node-to-sink scenario revealed that in general, the impact of inter-MAC interference on the performance increases both with the overlap between the geographical areas of the networks and when the data generation interval is reduced. This behaviour is hardly surprising given that increasing the overlap between the networks effectively increases the number of nodes that are affected by inter-MAC interference and that using a lower data generation interval increases channel utilisation, which in turn increases the probability of (heterogeneous) MAC-packets colliding with one another. Since there are no counterintuitive variations in the effects of inter-MAC interference with regard to these two parameters, it is thus not necessary to consider every single combination of overlap and data generation interval in this section. For every considered MAC protocol and network size, the influence of the network overlap and data generation interval on the performance are instead illustrated using only two graphs: one which varies the overlap between the networks for a fixed data generation interval and one wherein the data generation interval is varied for a fixed network overlap. For those interested, the full data set of the experiments discussed in this chapter is posted online [139].

2.3.1 Influence on the Duty Cycle

The effect of inter-MAC interference on the duty cycle of the LPL-MAC, T-MAC and TDMA MAC protocols is discussed below. The CSMA/CA MAC protocol is not discussed

given that this protocol always keeps the radio enabled and that the duty cycle of this protocol will thus not change as a result of inter-MAC interference. To prevent confusion between the two, the duty cycle measured for a specific network is always expressed as a percentage between 0% and 100% while the *relative difference* between two individual duty cycle measurements is expressed as a number of 'percentage-points' (%-points for short). (e.g., A duty cycle of 12.5% is 25%-points higher than a duty cycle of 10%.)

2.3.1.1 Duty Cycle of LPL-MAC

Figures 2.6 and 2.7 show the average, 5- and 95-percentile of the duty cycles measured for a network using the LPL-MAC protocol. Figure 2.6 shows the duty cycle for a data generation interval of 20 seconds and a varying amount of overlap between the networks while in figure 2.7 the data generation interval is varied and the overlap is fixed at 100%. In both figures the graph on the left shows the duty cycles measured for the 5x5 node deployment and the graph on the right displays those measured for the 10x10 node deployment. As expected, LPL-MAC has a higher duty cycle when another sensor network is also operating in the wireless environment than in the 'baseline' scenario where LPL-MAC receives no external interference. Moreover, the impact on the duty cycle increases with the overlap between the networks. When there is 0% overlap between the networks the duty cycle rises from 21.8% to at most 23.5% for the 5x5 deployment and from 29% to at most 30.3% for the 10x10 deployment. When the amount of overlap between the networks is increased, the average duty cycle can rise as high as 29% for the 5x5 deployment and as high as 40% for the 10x10 deployment (which constitutes an increase of respectively 33%-points and 37%-points). This increase in duty cycle is mainly caused by LPL-MAC nodes being woken up unnecessarily by channel activity generated by the interfering MAC protocol.

Figure 2.6 also shows that for both network sizes, TDMA has a larger impact on the duty cycle than any of the contention-based MAC protocols. This can be partially attributed to the fact that CSMA/CA and T-MAC will only send packets on the channel when they have actual data to send while nodes in the TDMA network will also send out SYNC-packets on a regular basis to keep the clocks of the nodes in the network synchronised. (T-MAC will also occasionally send out synchronisation messages, but these messages are sent at a much lower rate than the SYNC-packets of the TDMA protocol.) In addition, both CSMA/CA and T-MAC perform a CCA before sending a packet, which not only reduces collisions, but also the amount of time the LPL-MAC radio spends retransmitting packets and as a result the duty cycle is also reduced. Given that TDMA does not employ this mechanism, the difference between the effect that TDMA and the contention-based MAC protocols have on the duty cycle of LPL-MAC can also be partially attributed to this.

When the influence of interference is considered for different data generation intervals (see figure 2.7), it can be observed that while the overall duty cycle decreases for larger data generation intervals, the effect of interference on the duty cycle varies with the interfering MAC protocol used. When a CSMA/CA-network is deployed in the vicinity of the LPL-MAC network, the relative difference between the baseline duty cycle and the duty cycle of LPL-MAC when the CSMA/CA network is active, decreases when the length data generation interval is increased. For a data generation interval of 10 seconds, interference

with CSMA/CA results in a relative increase of 26.3%-points (from 27.3% to 34.5%) for the 5x5 deployment and 29.6%-points (from 38.6% to 50%) for the deployment compared to the baseline duty cycle. In contrast, a data generation interval of 60 seconds yields a relative increase in duty cycle of respectively 12.7%-points (from 17.8% to 20%) and 19.7%-points (from 21.2% to 25.4%). This behaviour can be attributed to the fact that increasing the data rate also increases the probability of both collisions and overhearing foreign packets. In contrast to the CSMA/CA MAC protocol, the overhead induced by T-MAC is more or less the same for all considered data generation intervals: around 18%-points for a network size of 25 nodes and around 20% for a network size of 100 nodes. The only exception is when a data generation interval of 10 seconds is used for the 10x10 deployment in which case the overhead is a little bit lower (around 15%-points). This different behaviour might be caused by the fact that, as further discussed in section 2.3.3, T-MAC suffers from scalability issues which cause the reliability to drop steeply when the data rate in the network is increased. This has the effect of tempering the amount of traffic sent in the T-MAC network and as a result the same is true for the effect this traffic has on the duty cycle of LPL-MAC.

For the TDMA MAC protocol, the relative impact on the duty cycle of the LPL-MAC protocol *increases* rather than decreases when a larger data generation interval is used. For the 5x5 deployment this effect is relatively minor with the relative increase in duty cycle varying between 31.7 and 33.3%-points. For the 10x10 deployment however, interference from the TDMA MAC protocol results in a relative increase in the duty cycle of LPL-MAC of 29.3%-points (from 38.6% to 49.9%) when a data generation interval of 10 seconds is used while for a data generation interval of 60 seconds, the duty cycle is increased by 42.8%-points (from 21.2% to 42.8%). To explain why this is the case it should first be noted that as discussed above, TDMA nodes regularly transmits SYNC-packets to keep the clocks of the nodes synchronised. Since the number of SYNC-packets transmitted only depends on the number of nodes used, this means that the TDMA MAC protocol effectively generates a *fixed* amount of 'synchronisation traffic' independent of the actual data generation interval used. In addition, it should also be noted that an LPL-MAC node can only be woken by a transmission from an interfering MAC protocol if it happens to be in sleep mode while that interfering transmission is taking place. Since LPL-MAC nodes spend more time in sleep mode if a larger data generation interval is used, this effectively means that the fixed amount of synchronisation traffic generated by the TDMA MAC protocol will have a relatively larger effect on the duty cycle of LPL-MAC when a large rather than a small data generation interval is used. Given moreover that for a sufficiently large data generation interval this synchronisation traffic is responsible for the overall majority of the interference generated by TDMA, this also explains why the impact of this interference on the duty cycle of LPL-MAC rises with the data generation interval.

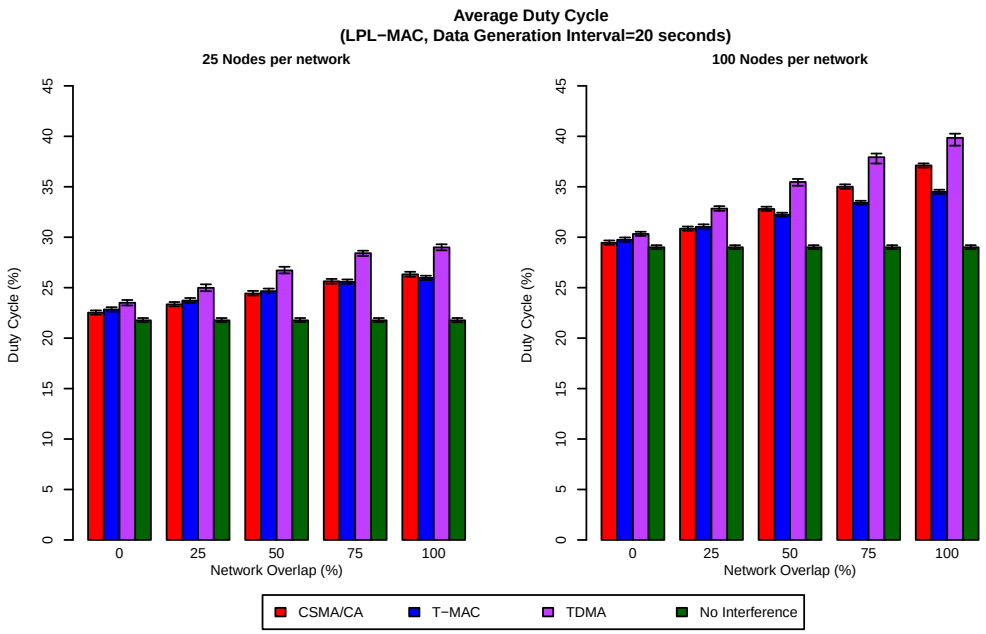


Figure 2.6: Average, 5- and 95%-tile duty cycle of LPL-MAC for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

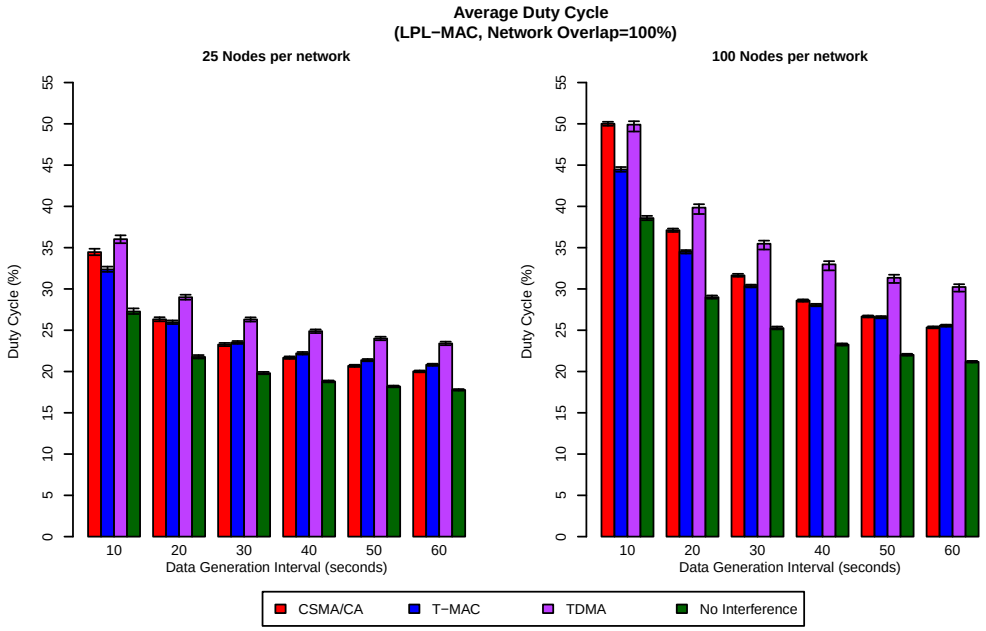


Figure 2.7: Average, 5- and 95%-tile duty cycle of LPL-MAC for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.1.2 Duty Cycle of T-MAC

Figures 2.8 and 2.9 show the average, 5- and 95-percentile of the duty cycles measured for a network using the T-MAC protocol when respectively the network overlap or the data generation interval are varied. As with the LPL-MAC protocol, the impact of interference on the duty cycle varies with the interfering MAC protocol used. It also varies with the overlap between the networks but this effect is less pronounced. For the 5x5 deployment, the duty cycle of T-MAC increases by at most 12.1%-points (from 14.5% to 16.3%) and even then this is only the case when the TDMA MAC protocol is used to generate interference. If one of the contention-based MAC protocols is used, the increase in duty cycle is never more than 5%-points. A similar observation can be made for the 10x10 deployment except that the increase in duty cycle is slightly higher: on average around 14.5%-points (from 17% to 19.5%) for TDMA, and less than 10%-points for the contention-based MAC protocols). The fact that the duty cycle of T-MAC is less affected by interference than the duty cycle of LPL-MAC can be explained by the fact that the duty cycle of a T-MAC node is only affected by interference if this interference is received at the end of the current active period of the node (i.e., when the node is polling the channel to ensure that there is no activity before entering sleep mode). This means that for the T-MAC protocol the fraction of the time during which interference can affect the duty cycle is much narrower than it is for the LPL-MAC protocol and as a result the impact of this interference on the duty cycle is also lower.

When the effect of the data generation interval is considered (see figure 2.9), it can be observed that the relative increase in duty cycle resulting from interference, rises when the data generation interval is reduced. In contrast to the LPL-MAC case, this is not only the case when T-MAC is interfered by a contention-based MAC protocol but also when it receives interference from the TDMA MAC protocol. The fact that LPL-MAC and T-MAC thus react differently to interference received from TDMA can be attributed to these protocols reacting differently to the interference resulting from TDMA's 'synchronisation traffic'. As discussed above, T-MAC nodes have to be in 'active' mode in order for their duty cycles to be affected by interfering transmissions. In addition, the duty cycle of T-MAC (and thus the length of the active period) rises with the load in the network. This means that, in contrast to LPL-MAC, the interference resulting from TDMA's 'synchronisation traffic' is more likely to affect the duty cycle of T-MAC for smaller rather than larger values of the data generation interval parameter and as a result the same is true for the overall effect that TDMA has on the duty cycle of T-MAC. When the relative differences between the baseline and 'interference' duty cycle measurements is considered it can once again be observed that T-MAC reacts less strongly to outside interference than LPL-MAC. For both deployments the duty cycle increases by at most 16.8%-points (from 16.5% to 19.3% for the 5x5 deployment and from 19.9% to 23.2% for the 10x10 deployment).

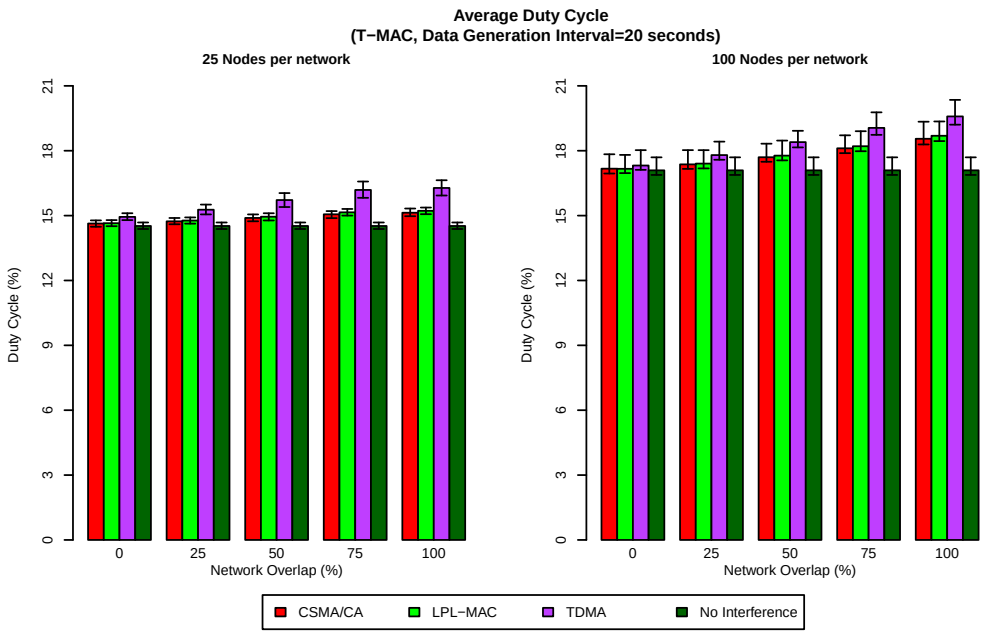


Figure 2.8: Average, 5- and 95%-tile duty cycle of T-MAC for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

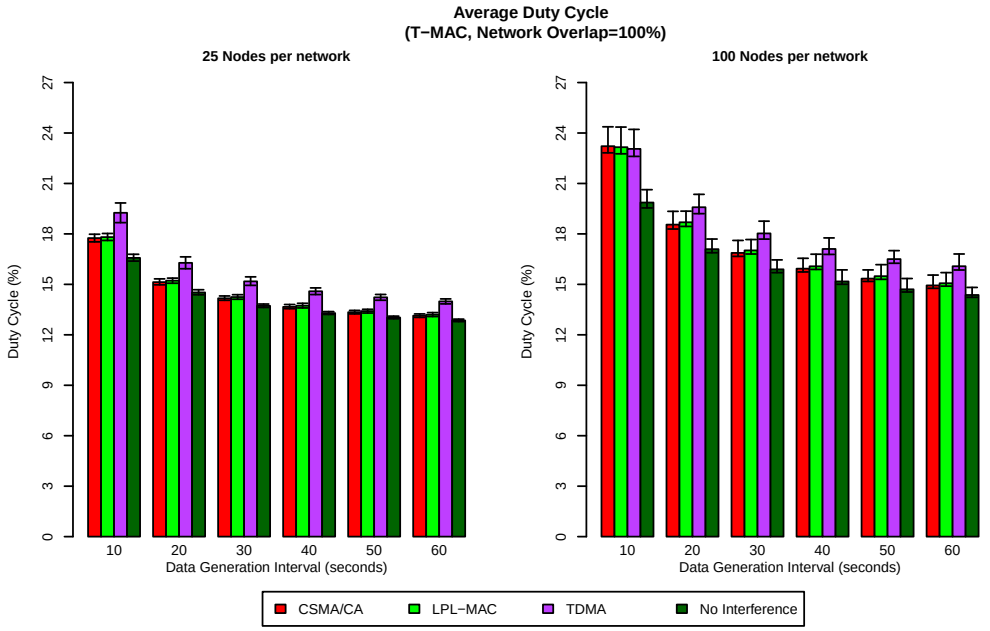


Figure 2.9: Average, 5- and 95%-tile duty cycle of T-MAC for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.1.3 Duty Cycle of TDMA

Figure 2.10 shows the duty cycle of TDMA for a varying amount of overlap between the networks while figure 2.11 shows the duty cycle of TDMA for different data generation intervals. These figures clearly show that in most cases the duty cycle of TDMA is only minimally affected by interference from another MAC protocol. In high-interference conditions (i.e with 100% overlap and/or a very small data generation interval) the duty cycle of TDMA does rise as a result of interference but even then this effect is very small (on average less than 5.6%-points over the entire range of considered parameters).

The reason for this resilience to interference, at least as far as duty cycle is concerned, is that the time at which TDMA nodes wake up and go to sleep is almost entirely decided by their slot allocation. The only manner in which interference can affect the duty cycle of TDMA is by preventing TDMA nodes from turning off their radios prematurely during 'rx'-slots (nodes turn off their radio in 'rx'-slots before the end of the slot unless activity is detected on the channel). Given that even then the radio will only remain enabled until the end of the slot and that TDMA slots are very small (±5ms) interference only has a minimal effect on the duty cycle of TDMA.

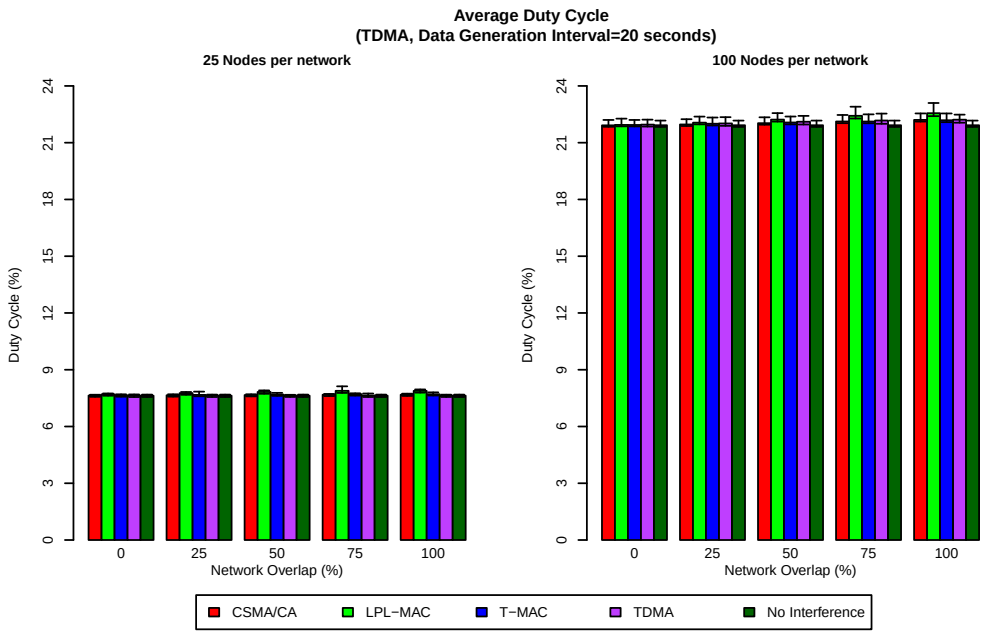


Figure 2.10: Average, 5- and 95%-tile duty cycle of TDMA for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

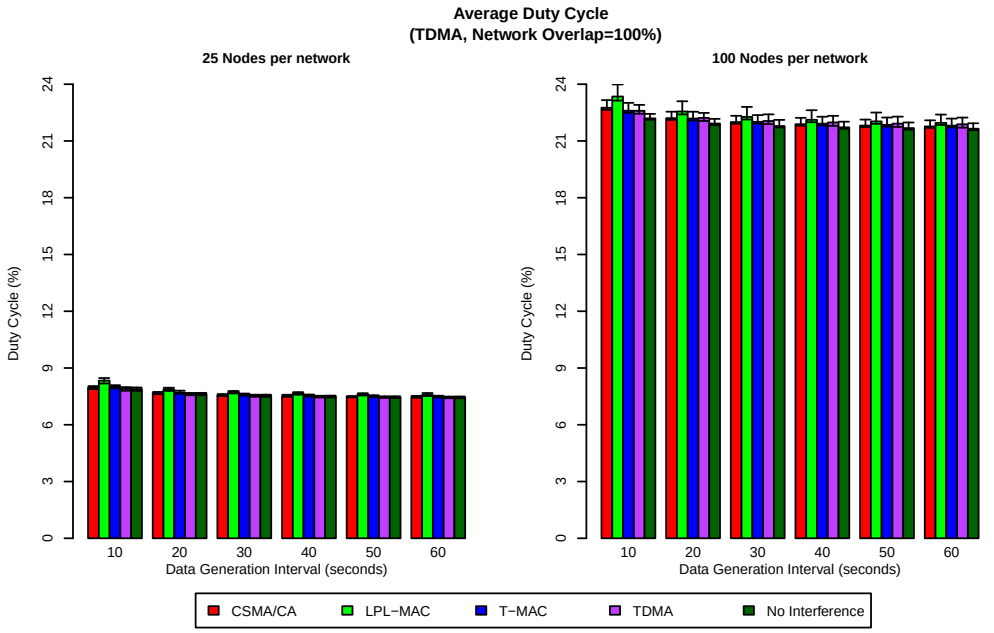


Figure 2.11: Average, 5- and 95%-tile duty cycle of TDMA for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.2 Influence on the Hop Count

The average hop counts of the different MAC protocols are shown for varying amounts of overlap and varying data generation intervals in figures 2.12 to 2.19. The first thing to notice in these figures is that inter-MAC interference has a significantly smaller impact on the hop count than it does on the duty cycle of the networks involved. For the 5x5 deployment, interference only has a very small or even negligible effect on the average hop count, regardless of which MAC protocols are used by the networks. For the 10x10 deployment, the impact of interference is more significant but even then it is only noticeable in what might be referred to as 'high interference' situations (i.e., when the network overlap is very high and the data generation interval is very low at the same time).

Interestingly, inter-MAC interference will not necessarily cause the average hop count to rise. Depending on the specific MAC protocols used, it can also cause the average hop count to decrease slightly. This is for instance the case when a TDMA network is being interfered by a network using T-MAC (see figure 2.19). To explain this behaviour, it should first be noted that interference does not affect the hop count of a network directly and is instead a consequence of packets from heterogeneous MAC protocols colliding with one another. (This also explains why the hop count is only affected in 'high interference' situations and why, as is clear from the figures discussed in section 2.3.3, variations in hop count tend to coincide with a significant drop in reliability.) When such collisions occur, they can affect the hop count of the network in one of two ways. When interfering transmissions collide with probe packets sent by the routing layer, this will cause the reliability reported for the nearby links to decrease. Since routes are calculated based

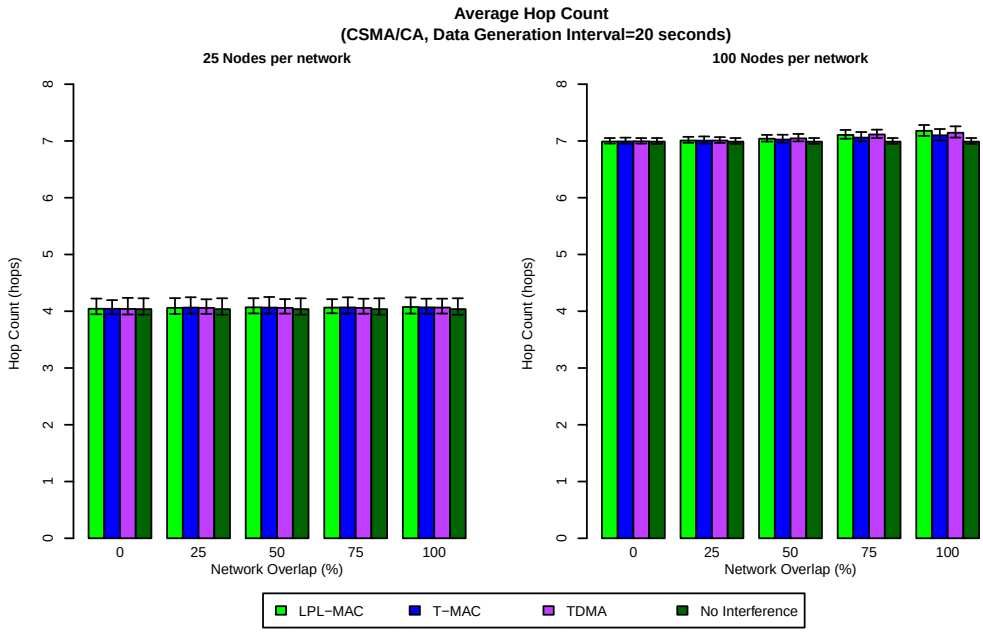


Figure 2.12: Average, 5- and 95%-tile hop count of CSMA/CA for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

on the 'ETX'-metric and that, as discussed in section 2.2.1, this metric balances between reliability and hop count, this will cause the routing layer to select longer paths that avoid the affected links which in turn causes the hop count of the network to increase.

When interfering transmissions collide with regular data transmissions however, this tends to reduce the hop count recorded for the network receiving the interference. The reason for this is that a packet has to arrive at the destination in order for the path it followed to be considered in the calculation of the hop count of the network. Given moreover that interference tends to have a stronger effect on the end-to-end reliability of longer paths (more collision opportunities) this causes shorter paths to be counted more than longer paths which results in a lower average hop count. Depending on the specific combination of MAC protocols used, either of these effects can impact the average hop count more strongly and as a result interference between the networks will in some cases cause the hop count of the network to rise while for others the hop count is slightly lowered. Regardless of which effect happens to be the strongest for a particular case, these effects either cancel each other out pretty well or don't affect the average hop count all that much. This is clear from the measurements shown in figures 2.12 to 2.19 which show that even in the worst case scenario, there is only a 1.5% difference between the average baseline hop count and the hop count when receiving interference from another network.

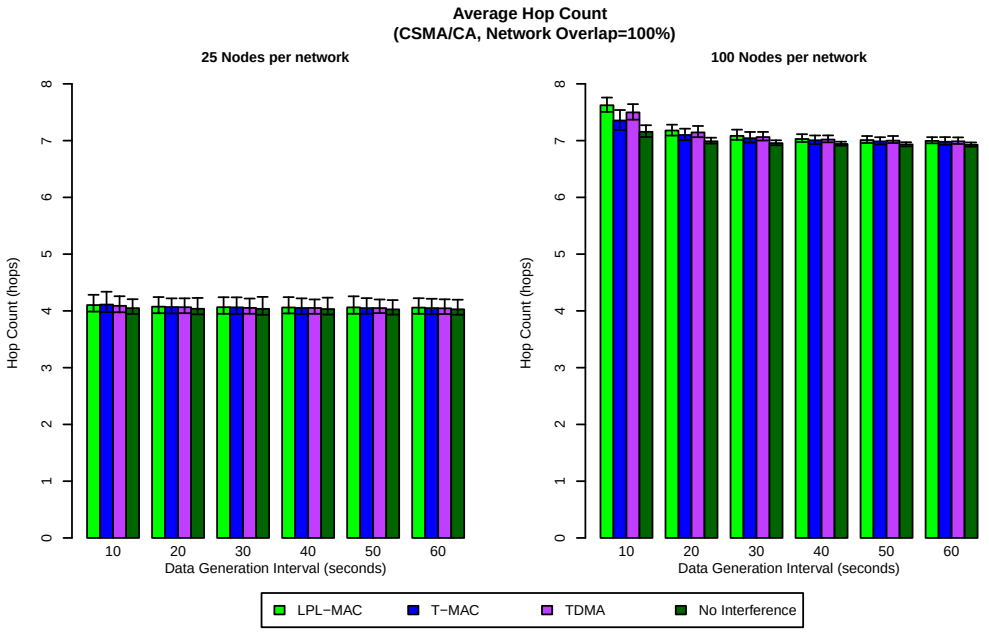


Figure 2.13: Average, 5- and 95%-tile hop count of CSMA/CA for the node-to-sink scenario in the presence of interference for different data generation intervals.

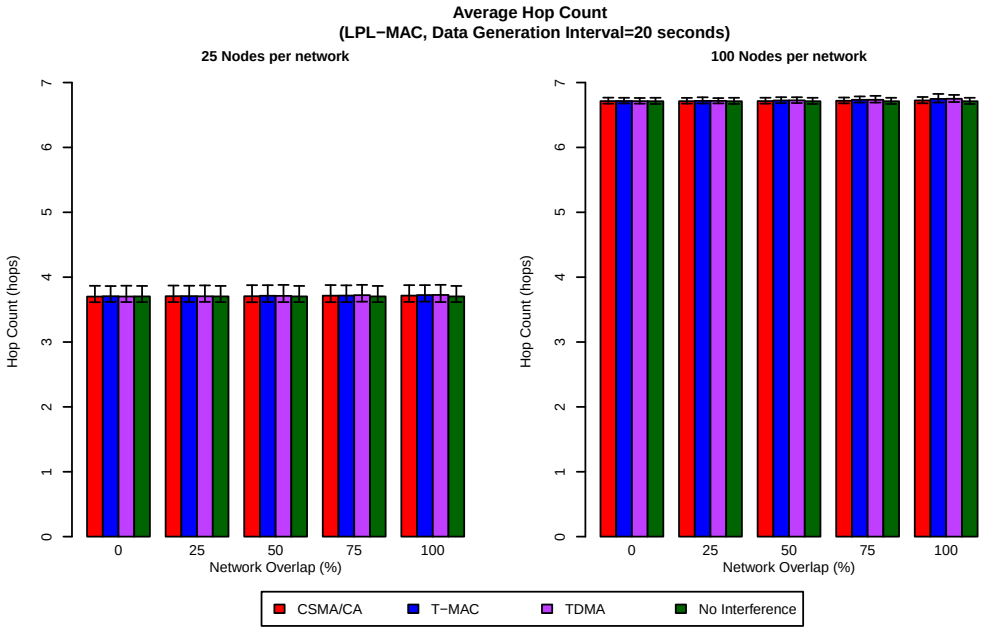


Figure 2.14: Average, 5- and 95%-tile hop count of LPL-MAC for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

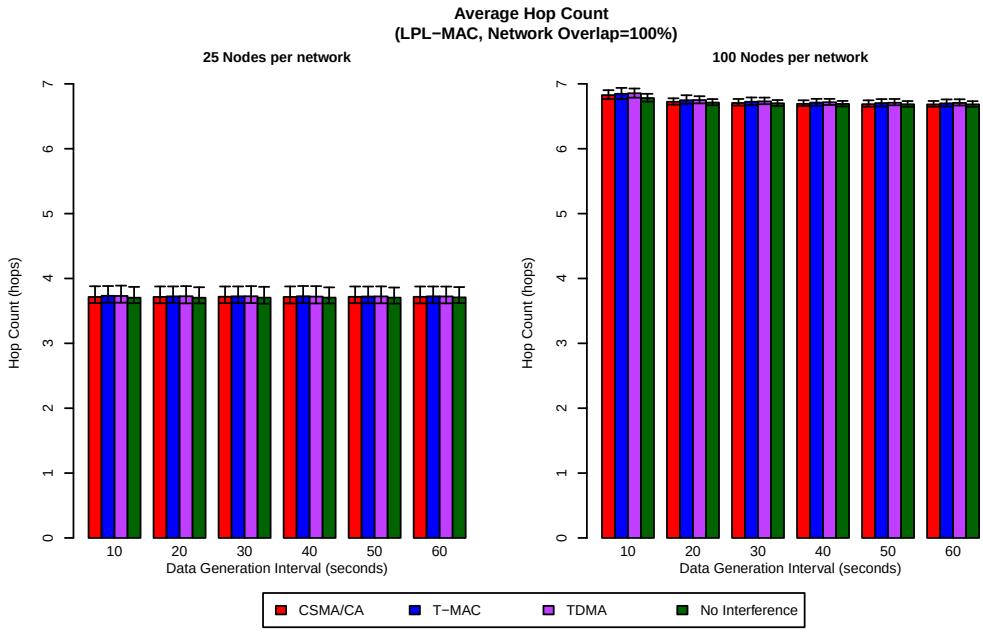


Figure 2.15: Average, 5- and 95%-tile hop count of LPL-MAC for the node-to-sink scenario in the presence of interference for different data generation intervals.

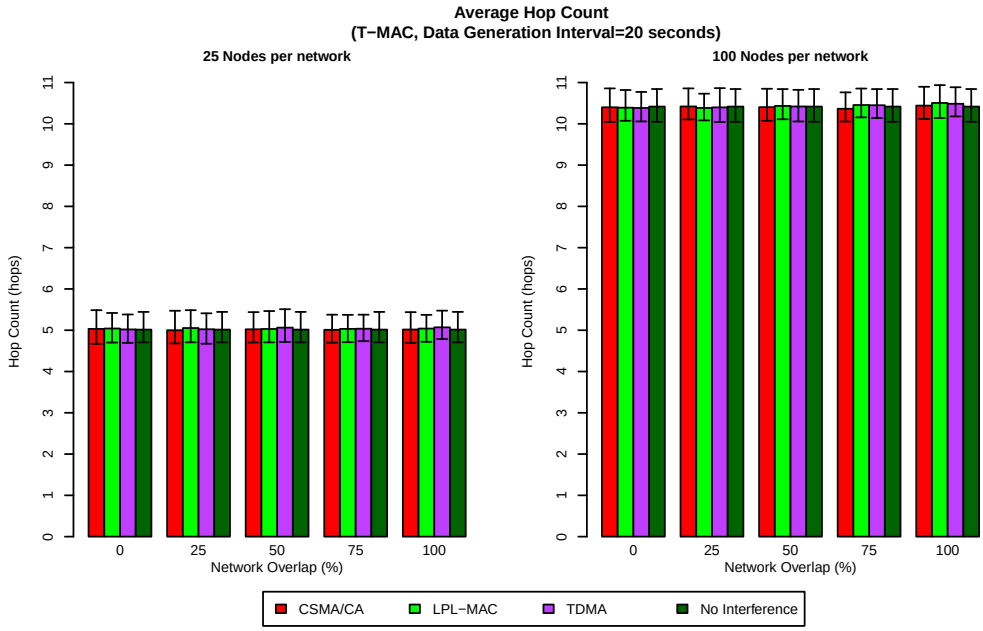


Figure 2.16: Average, 5- and 95%-tile hop count of T-MAC for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

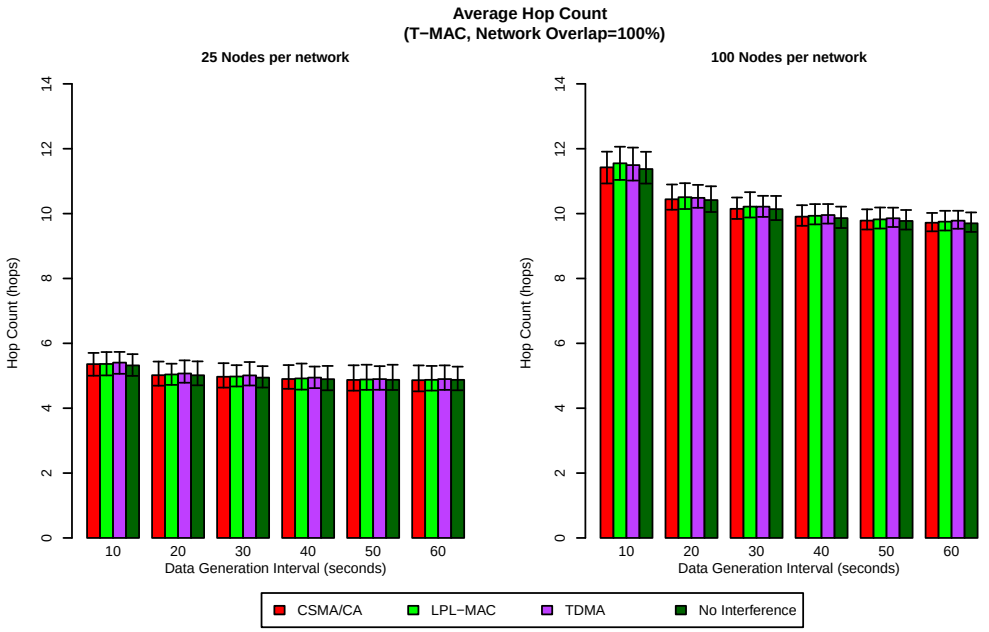


Figure 2.17: Average, 5- and 95%-tile hop count of T-MAC for the node-to-sink scenario in the presence of interference for different data generation intervals.

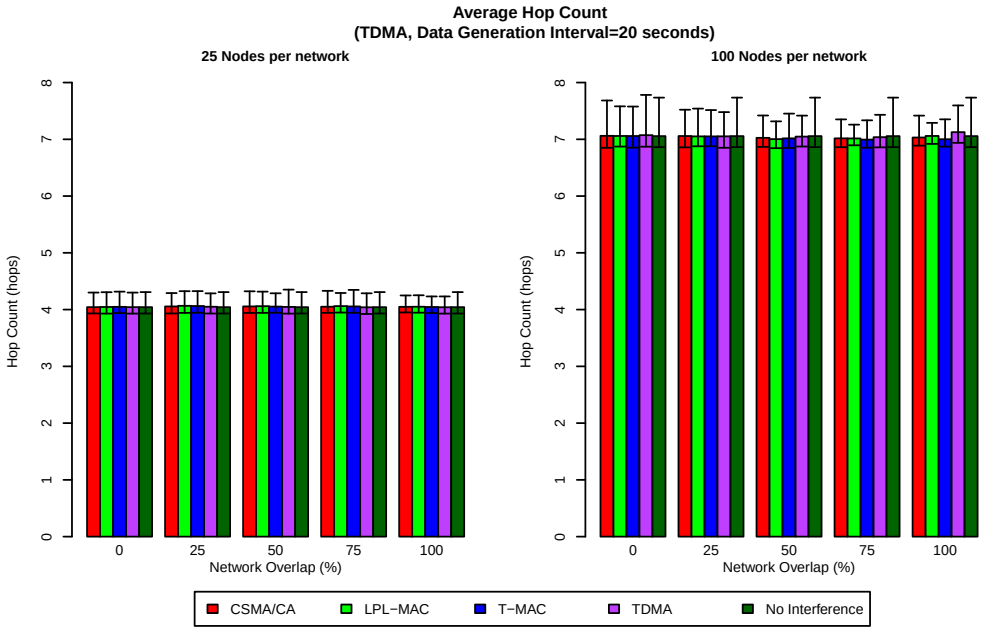


Figure 2.18: Average, 5- and 95%-tile hop count of TDMA for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

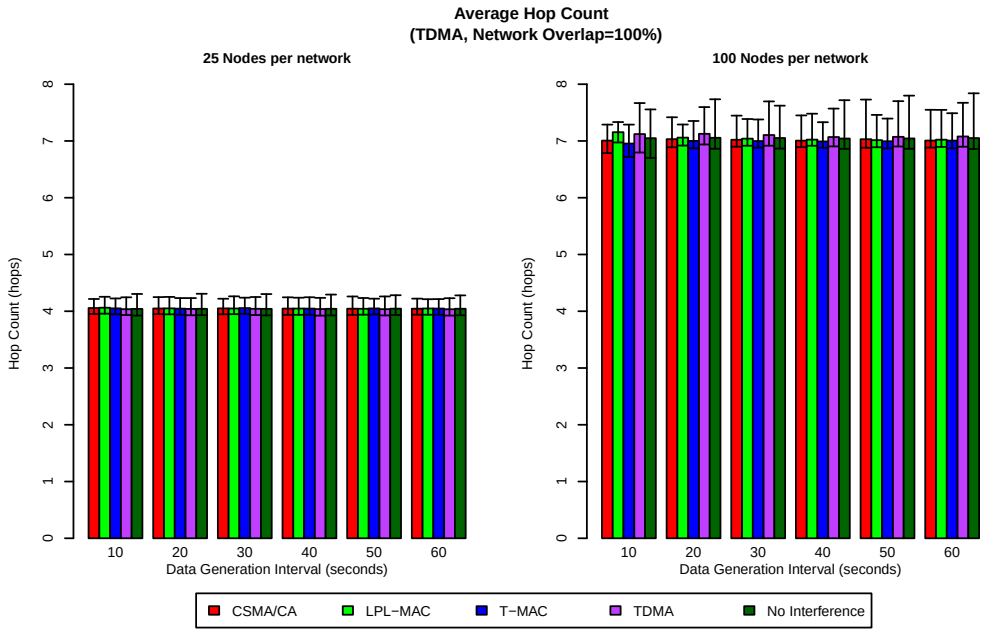


Figure 2.19: Average, 5- and 95%-tile hop count of TDMA for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.3 Influence on the Reliability

The effect of inter-MAC interference on the end-to-end reliability of the MAC protocols is discussed below. As with the duty cycle metric, reliability is always expressed as a percentage between 0% and 100% and the *relative difference* between two individual reliability measurements is expressed as a number of 'percentage-points' (%-points for short). (e.g., a reliability of 81% is 10%-points lower than a reliability of 90%.)

2.3.3.1 Reliability of CSMA/CA

Figures 2.20 and 2.21 show the average, 5- and 95-percentile of the reliabilities measured for a network using the CSMA/CA MAC protocol. Figure 2.20 shows the reliability for a fixed data generation interval and a varying amount of overlap between the networks while figure 2.21 considers the reliability for different data generation intervals.

Before discussing the effects of inter-MAC interference however, it should first be noted that the reliability of CSMA/CA is not decided solely by the 'amount' of interference it receives from another MAC protocol. This is because, even with the CCA-checks and exponential backoff algorithm used to govern access to the channel, CSMA/CA packets can still collide with one another due to the hidden terminal problem. In addition, the CSMA/CA implementation used here (i.e., the non-beaconed mode of IEEE 802.15.4) will only perform a limited number of 'backoffs' before dropping the packet entirely (this is done to reduce congestion). Given both that the probability of packets colliding and the probability of packets being dropped rises with the network load, it should come as

no surprise that reducing the data generation interval will also result in a lower reliability. For the 5x5 node deployment this effect is minimal as the reliability drops from 94% to at worst 91.7% when the data generation interval is reduced. For the 10x10 node deployment however the reliability can drop from nearly 95% to a little under 80%, which indicates that the cumulative data rate of the different nodes is starting to exceed the capacity that can be achieved using CSMA/CA.

When the 'baseline' performance of CSMA/CA is compared to the case where it also receives interference from another network, it is clear that the reliability of CSMA/CA is negatively affected by this interference regardless of the network overlap or the data generation interval used. Moreover, the extent to which the reliability is affected rises both when the overlap between the networks is increased and when the data generation interval is reduced. This behaviour is to be expected given that CSMA/CA reacts in much the same way to interfering transmissions as it does to transmissions from other CSMA/CA nodes and that the effect of interference on the reliability thus rises with the number of interfering transmissions taking place in the immediate vicinity of the CSMA/CA nodes.

The effect of inter-MAC interference also varies significantly with the number of nodes used. When 25 nodes are used per network, interference only has a limited effect on the reliability of CSMA/CA: over all considered test cases, the average end-to-end reliability drops by at most 2.7%-points (from 91.65% to 89.2%) as a result of interference. When 100 nodes are used per network, interference can cause the reliability of CSMA/CA to drop by as much as 9.4%-points (from 79.7% to 72.2%), but only if a data generation interval of 10 seconds is used. For a data generation interval of 20 seconds or higher, the average reliability never drops by more than 5.5%-points (from 89.3% to 84.6%).

When the impact of interference is compared between the different interfering MAC protocols it is clear that interference from LPL-MAC has the largest effect on the reliability of CSMA/CA. This is most likely due to the fact LPL-MAC will (re-)transmit a packet numerous times to wakeup neighbouring nodes and that it will therefore generate significantly more interfering transmissions per transmitted (network layer) data packet than any of the other MAC protocols. TDMA has a smaller effect on the reliability of CSMA/CA but considering that this protocol generates the lowest number of interfering transmissions for each data packet (no retransmissions), the difference between LPL-MAC and TDMA is not as large as one might expect (interference from LPL-MAC reduces the reliability by at most 9.4%-points, for TDMA this is at most 7.5%-points). This relatively high impact is most likely caused by the fact that TDMA does not check that the channel is clear before transmitting packets and that TDMA transmissions therefore have a larger probability of colliding with ongoing CSMA/CA transmissions.

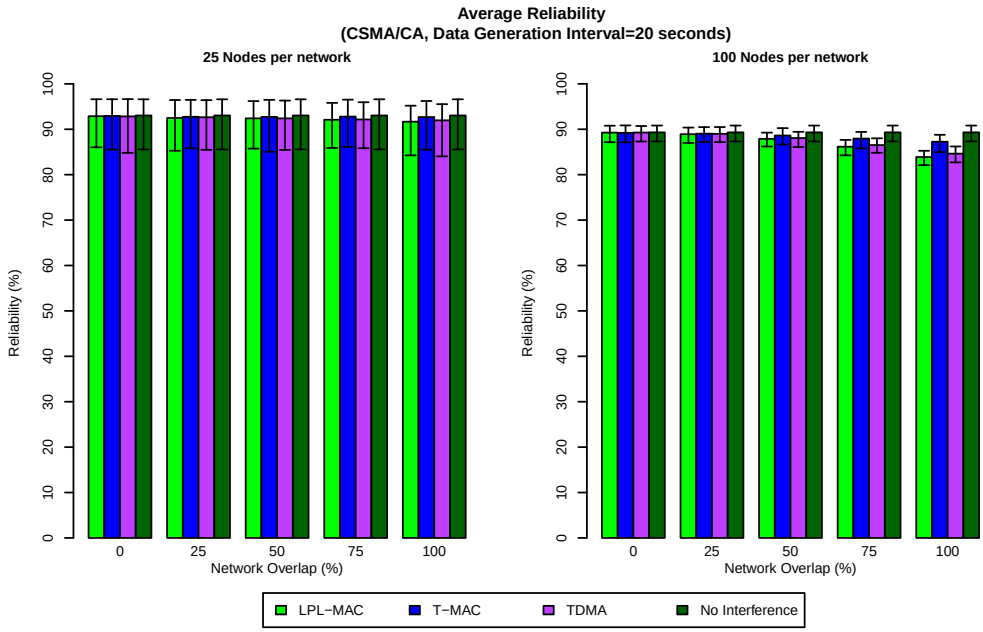


Figure 2.20: Average, 5- and 95%-tile reliability of CSMA/CA for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

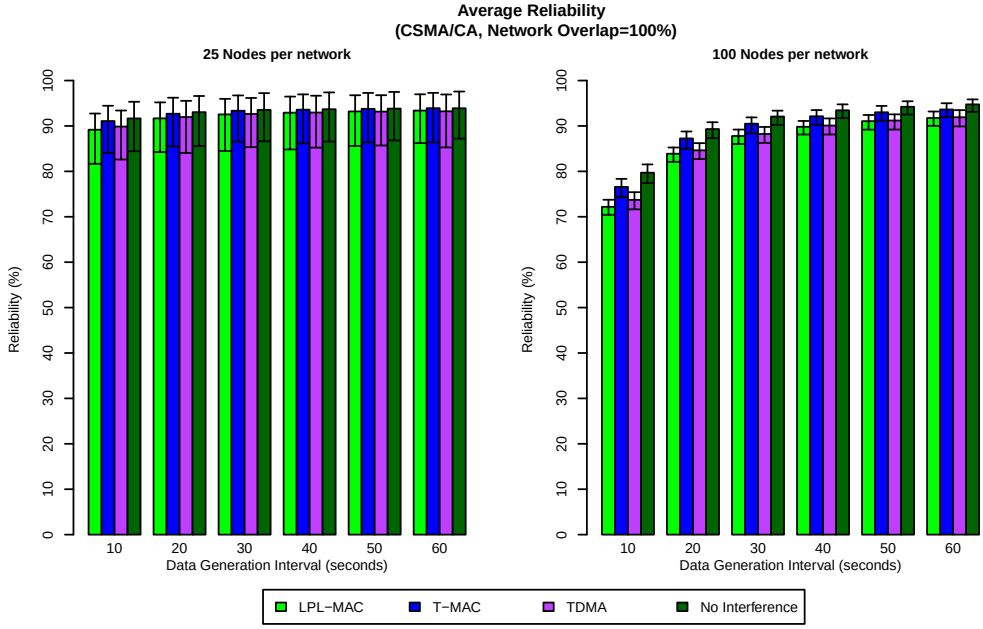


Figure 2.21: Average, 5- and 95%-tile reliability of CSMA/CA for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.3.2 Reliability of LPL-MAC

The average, 5- and 95%-tile of the reliabilities measured for the LPL-MAC protocol are shown in figures 2.22 and 2.23. Figure 2.22 shows the reliability of LPL-MAC for a varying amount of overlap while figure 2.23 displays the reliability for a varying data generation interval. From these figures it is clear that LPL-MAC reacts in much the same way to interference as the CSMA/CA MAC protocol. As with CSMA/CA, the impact of interference rises with the amount of overlap between the networks and increases when the data generation interval is reduced. In addition, as with the CSMA/CA MAC protocol, the reliability of LPL-MAC also varies with the data generation interval even if no other network is present in the wireless environment. This behaviour should not come as a surprise given that, as discussed in section 2.1.7, the LPL-MAC protocol used here is implemented as an extension on top of CSMA/CA. That being said, figures 2.22 and 2.23 also make it plain that LPL-MAC is noticeably more resilient to interference than CSMA/CA: over the entire range of considered test parameters, inter-MAC interference will cause the average reliability of LPL-MAC to drop by at most 2.3%-points (86.5% to 84.5%).

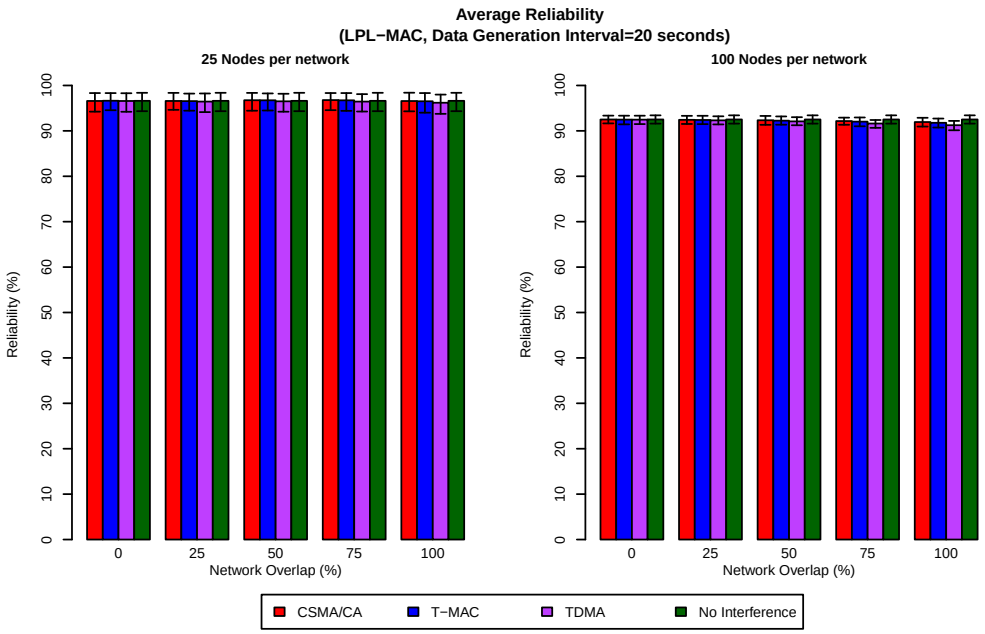


Figure 2.22: Average, 5- and 95%-tile reliability of LPL-MAC for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

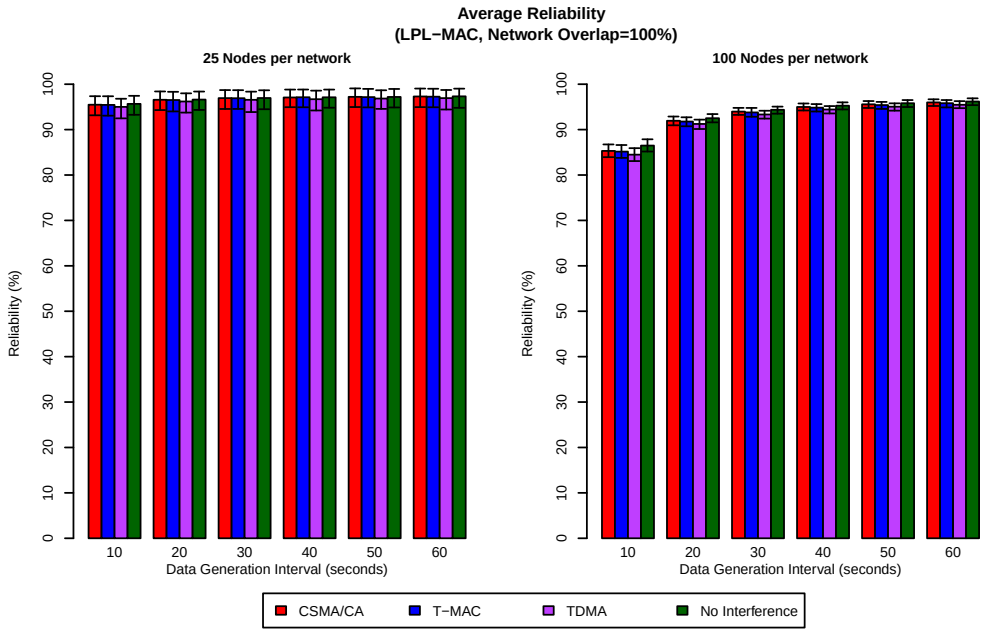


Figure 2.23: Average, 5- and 95%-tile reliability of LPL-MAC for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.3.3 Reliability of T-MAC

Figures 2.24 and 2.25 show the average, 5- and 95-percentile of the reliabilities measured for a network using the T-MAC protocol. Before discussing the effects of interference, it should first be noted that even without outside interference T-MAC is by far the least reliable of the contention-based MAC protocols. As with CSMA/CA and LPL-MAC, reducing the data generation interval has a negative effect on the reliability of the network but for the T-MAC protocol this effect is much more significant than it is for the other two MAC protocols. Moreover, this is not only the case for the 10x10 deployment but also for the 5x5 deployment. For the 5x5 deployment, reducing the data generation interval from 60 to 10 seconds causes the average reliability to drop from 76% to 52%. For the 10x10 deployment the average reliability is at best 55% and can drop down to 26% when the data generation interval is reduced.

While it is clear that the T-MAC protocol used here thus suffers from severe scalability issues, it should be noted that this is not because of the modifications discussed in section 2.1.7. As previously discussed in section 2.2.1, the T-MAC implementation provided by Castalia (which is a close implementation of the original specification) was originally used in this work, but this implementation had even more severe reliability issues (for both the 5x5 and 10x10 deployment the average reliability was at most 41%) and was extremely unstable. The modifications discussed in section 2.1.7 have increased both the reliability and stability of T-MAC but have clearly not fully resolved the scalability issues. While additional modifications might have alleviated some of these issues, that would have caused the T-MAC implementation used here to deviate too much from the

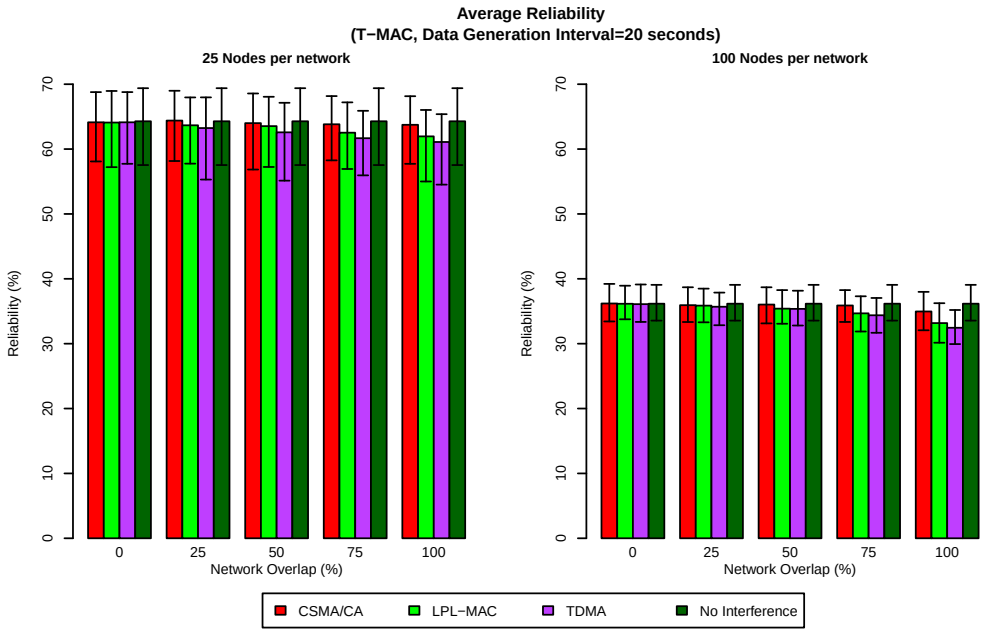


Figure 2.24: Average, 5- and 95%-tile reliability of T-MAC for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

original specification and as a result only the modifications discussed in section 2.1.7 are used.

When the effect of inter-MAC interference is considered it is clear that, as with CSMA/CA and LPL-MAC, the effect of interference on the reliability of T-MAC increases when the overlap between the networks is increased as well as when the data generation interval is reduced. Depending on the interfering MAC protocol used, this interference can cause the average reliability of the T-MAC protocol to drop by 7.7%-points (from 51.6% to 47.7%) for the 5x5 deployment while for the 10x10 deployment, the average reliability is reduced by at most 15.8%-points (from 26.3% to 22.2%).

When the impact of interference is compared between the different interfering MAC protocols it is clear that TDMA has the largest impact on the reliability of the T-MAC protocol. As was the case for the LPL-MAC protocol, the fact that TDMA has a noticeably higher impact than any of the contention-based MAC protocols is most likely due to the fact that TDMA does not perform a 'CCA-check' before transmitting a packet. When the impact of the two contention-based MAC protocols is compared it can be observed that of the two, LPL-MAC has the largest effect on the reliability of T-MAC, which is most likely the result of LPL-MAC performing significantly more transmissions to send a single packet than CSMA/CA.

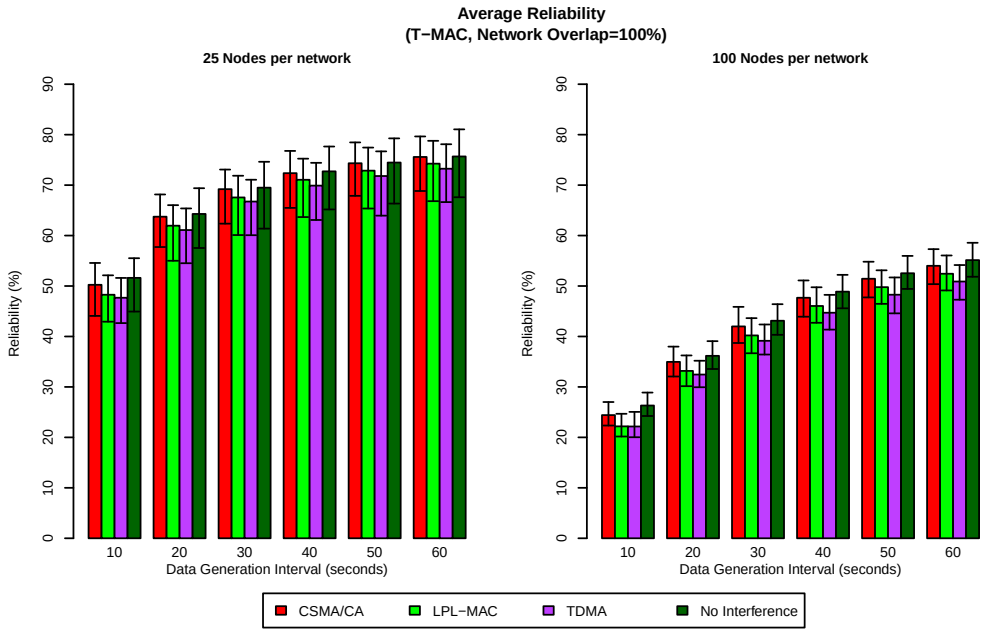


Figure 2.25: Average, 5- and 95%-tile reliability of T-MAC for the node-to-sink scenario in the presence of interference for different data generation intervals.

2.3.3.4 Reliability of TDMA

The reliability of the TDMA MAC protocol is shown in figures 2.26 and 2.27 for respectively a varying amount of overlap between the networks and a varying data generation interval. The first thing to notice in these figures is that there is a lot more variation in the reliability measured for individual test runs (the 5- and 95%-tiles are much further apart) than for the other MAC protocols. This can be attributed to the fact that the TDMA MAC protocol used here does not retransmit lost packets and that variations in link quality therefore have a much larger influence on the overall end-to-end reliability. In addition, the effect of the data generation interval on the reliability of TDMA when not under the influence of inter-MAC interference should also be noted.

For the 5x5 deployment, the data generation interval does not affect the reliability of the TDMA MAC protocol at all. This is in contrast to the other three MAC protocols and is explained by the fact that, since each node is assigned a fixed number of dedicated time slots for sending packets, nodes don't need to contend for channel access. Given that TDMA packets therefore do not collide with one another this means that, as long as each node is assigned sufficient time slots for packet transmission, the reliability of the network is unaffected by the data generation interval used. The same behaviour can be observed for the 10x10 node deployment but only if the data generation interval is sufficiently large (i.e., 50 or 60 seconds). For smaller values, the reliability of TDMA drops dramatically with the length of the data generation interval. This is caused by the fact that, as discussed in section 2.1.7, each node is assigned a fixed number of time slots for transmission, regardless of the actual bandwidth requirements of the node. This creates

a problem for the node-to-sink scenario considered in this section since in this scenario all traffic is directed to a single sink node which means that nodes in the immediate vicinity of the sink require more bandwidth than nodes that are further away. Under high network-load conditions this causes data packets to be dropped by intermediate nodes which in turn affects the end-to-end reliability of the network. (When this happens, the network can be said to be "overloaded").

When considering the influence of inter-MAC interference, it can be observed that, as with the other MAC protocols, the effect of interference on the reliability of TDMA increases with the overlap between the networks. As shown in figure 2.26, the effect of interference is negligible when the two networks are deployed side by side (0% overlap). When the overlap between the network increases however, the reliability can drop by 8.9%-points (from 76.5% to 69.7%) and 13.9%-points (from 47.7% to 41%) for respectively the 5x5 and 10x10 deployment. This drop in reliability is caused by the fact that, despite the collision-avoidance techniques used, collisions with TDMA packets will still occur if the node using a contention-based MAC protocol starts its transmission before TDMA does.

When the effect of inter-MAC interference is considered for a varying data generation interval, it turns out that the manner in which the reliability is affected by interference depends on whether or not the TDMA network is overloaded. If the network is not overloaded (i.e., when either the 5x5 deployment or a data generation interval larger than 40 seconds is used), the relative difference between the average 'baseline' reliability and the reliability of TDMA in interference-conditions increases when the data generation interval is reduced (which is the same behaviour observed for the other MAC protocols). When the TDMA network is overloaded however, this relative difference decreases rather than increases when the data generation interval is reduced. For a data generation interval of for instance 10 seconds, the average reliability of TDMA drops by at most 12.2%-points (from 27.9% to 24.5%) while for a data generation interval of 40 seconds, the average reliability can drop by as much as 17%points (from 68% to 56.5%). This different behaviour can be attributed to the fact that, as discussed above, TDMA nodes will drop packets when the network is overloaded. This in turn limits the actual number of TDMA transmissions taking place in the network and thus the number of packets that can collide with interfering transmissions.

When the impact of interference is compared between the different interfering MAC protocols it is clear that, as before, interference from the LPL-MAC protocol has a much more significant impact on the reliability than the other two contention-based MAC protocols. When LPL-MAC is used to generate interference, the reliability of TDMA drops by at most 15%-points (from 76.6% to 65.2%) as a result of interference while for CSMA/CA and T-MAC the average reliability drops by at most 7.8%-points (from 76.6% to 70.7%). When both networks involved use TDMA however, the reliability does not react to interference in the manner one might expect. Given that the TDMA MAC protocol used here uses neither CCAs nor retransmissions one would expect the reliability to be severely affected by the interfering transmissions generated by the respective networks. Instead, this interference only has a negligible impact on the average reliability for the 5x5 deployment while for the 10x10 deployment the effect of this interference is significant but still not as large as one might expect: the average reliability drops by at most 18%-points (from 71.5% to 58.6%) which is only a little bit more than the 15%-point drop caused by interference from the LPL-MAC protocol (which does use CCAs).

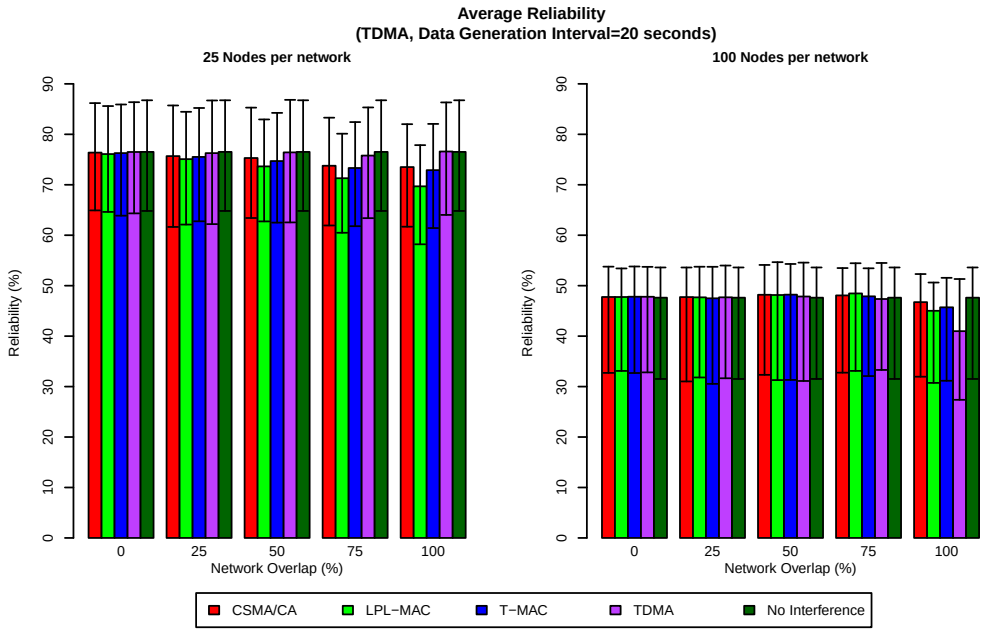


Figure 2.26: Average, 5- and 95%-tile reliability of TDMA for the node-to-sink scenario in the presence of interference for a varying amount of network overlap.

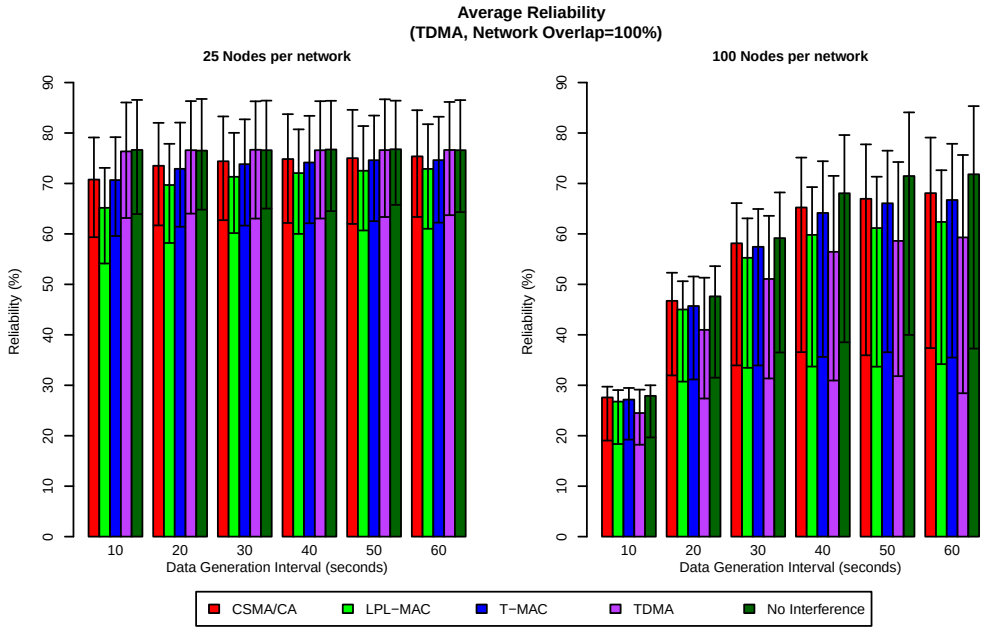


Figure 2.27: Average, 5- and 95%-tile reliability of TDMA for the node-to-sink scenario in the presence of interference for different data generation intervals.

It should be noted at this point that in earlier tests which used pre-calculated static routes (but otherwise the same network stack) interference between the two TDMA MAC protocols did have a profound effect on the reliability. Once the dynamic route selection mechanism discussed in section 2.2.1 was added however, the effect of TDMA - TDMA interference all but disappeared for the 5x5 deployment while for the 10x10 deployment the impact on the reliability was much reduced. This indicates that the relatively small impact of interference observed in the TDMA - TDMA case is not due to the properties of the TDMA MAC protocol itself but is instead the result of the routing layer selecting, where possible, paths that do not interfere with one another.

2.4 Interference in the random-flows scenario

The tests performed for the random-flows scenario revealed that in general, the impact of inter-MAC interference on the performance increases both with the overlap between the geographical areas of the networks and when the data generation interval is reduced. As with the node-to-sink scenario there are no counterintuitive variations in the effects of inter-MAC interference with regard to these two parameters and as a result not all combinations of overlap and data generation interval need to be considered. As with the previous section, the influence of the network overlap and data generation interval on the performance are illustrated using two graphs: one which varies the overlap between the networks for a fixed data generation interval and one wherein the data generation interval is varied for a fixed network overlap.

2.4.1 Influence on the Duty Cycle

The effect of inter-MAC interference on the duty cycle of the LPL-MAC, T-MAC and TDMA MAC protocols is discussed below. The CSMA/CA MAC protocol is not discussed given that this protocol always keeps the radio enabled and the duty cycle of this protocol will thus not change as a result of inter-MAC interference.

2.4.1.1 Duty Cycle of LPL-MAC

Figures 2.28 and 2.29 show the average, 5- and 95-percentile of the duty cycles measured for a network using the LPL-MAC protocol. Figure 2.28 shows the duty cycle for a data generation interval of 20 seconds and a varying amount of overlap between the networks while in figure 2.29 the data generation interval is varied and the overlap is fixed at 100%. These figures show that LPL-MAC reacts very similar to interference in the random-flows scenario as it did in the node-to-sink scenario. As with the node-to-sink scenario, the effect of interference on the duty cycle varies with the interfering MAC protocol used and rises when the amount of overlap between the networks is increased. Moreover, the relative differences in duty cycle measured for the random-flows scenario are pretty similar to those measured for the node-to-sink scenario. When there is 0% overlap between the networks, the average duty cycle rises from 21.2% to at most 22.6% for the 5x5 deployment (versus 21.7% to 22.5% in the node-to-sink scenario). For the 10x10 deployment the average duty cycle rises from 28.4% to at most 29.9% (versus 29% to 30.3%). When the amount of overlap between the networks is increased, the duty cycle can rise as high as 29.4% and 42.7% (versus 29% and 40%) for respectively the 5x5 and 10x10 deployments.

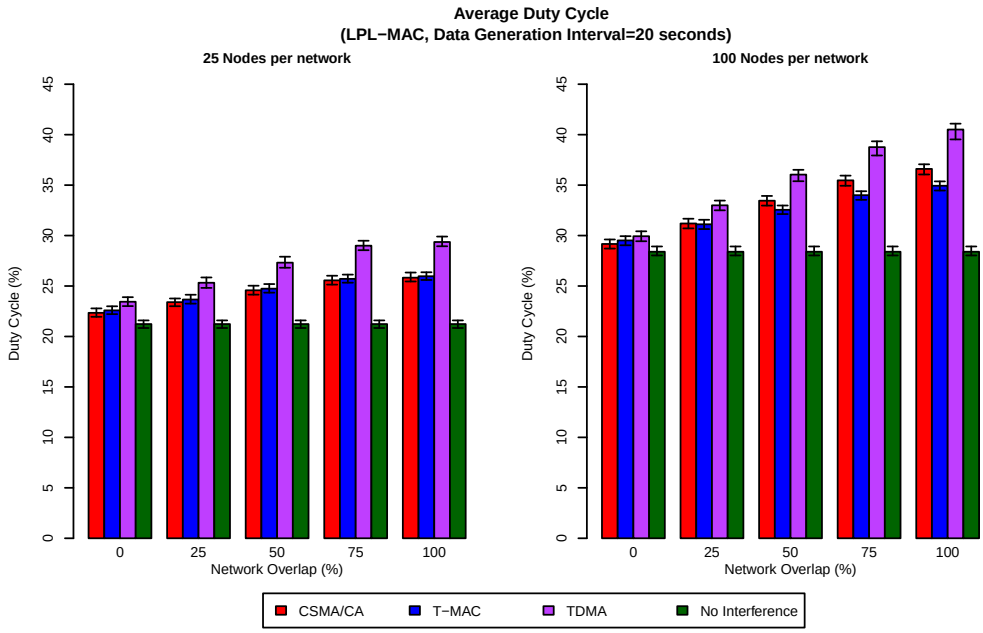


Figure 2.28: Average, 5- and 95%-tile duty cycle of LPL-MAC for the random-flows scenario in the presence of interference for a varying amount of network overlap.

The impact of interference also varies with the data generation interval but the precise effect of the data generation interval depends on the interfering MAC protocol used. As with the node-to-sink scenario, the impact of interference from CSMA/CA increases when the data generation interval is reduced while for interference from the TDMA MAC protocol the opposite is once again true: in that case the average duty cycle of LPL-MAC is more affected by interference when a larger rather than a smaller data generation interval is used. When the size of the performance overhead is considered, it can be observed that, when the CSMA/CA MAC protocol is used to generate interference, this performance overhead is almost the same in the random-flows scenario as it is in the node-to-sink scenario. For the random-flows scenario the performance overhead varies between 14.2 and 26.9%-points for the 5x5 deployment (versus 12.7 to 26.4%-points in the node-to-sink scenario) and between 22.1 and 28.9%-points for the 10x10 deployment (versus 19.6 and 29.7%-points).

When interference is generated by TDMA, the performance overhead is somewhat larger for the random-flows scenario than it is for the node-to-sink scenario. The average overhead varies between 36.7 and 38.3%-points for the 5x5 deployment (versus 31.7 to 33.2%-points) and between 31.2 and 51%-points for the 10x10 deployment (versus 29.3 and 42.8%-points). When LPL-MAC is combined with T-MAC, it can be observed that the overhead induced by T-MAC is once again more or less the same for all considered data generation intervals but that it is also somewhat higher than in the node-to-sink scenario. For the 5x5 deployment the performance overhead hovers around 21%-points (versus 18-%points in the node-to-sink scenario) while for the 10x10 deployment the average duty cycle is increased by around 27.5%-points (versus 20%-points). As with the node-to-sink

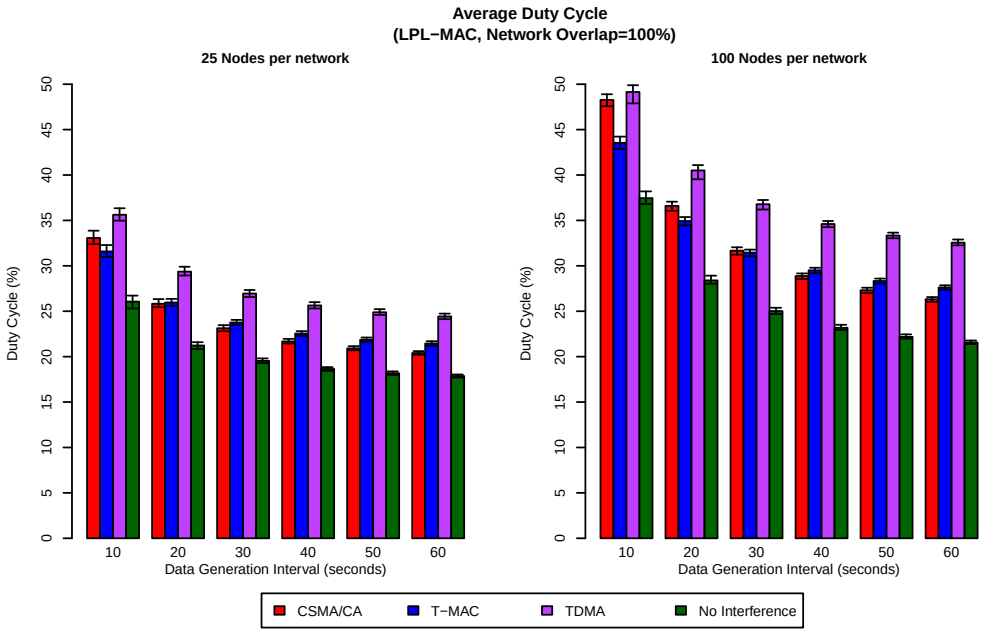


Figure 2.29: Average, 5- and 95%-tile duty cycle of LPL-MAC for the random-flows scenario in the presence of interference for different data generation intervals.

scenario the only exception to this is when a very low data generation interval (10 or 20 seconds) is used for the 10x10 deployment in which case the overhead is even lower (16 to 23%-points).

2.4.1.2 Duty Cycle of T-MAC

Figures 2.30 and 2.31 show the average, 5- and 95-percentile of the duty cycles measured for a network using the T-MAC protocol when respectively the network overlap or the data generation interval are varied. As with the LPL-MAC protocol, the T-MAC protocol reacts in mostly the same way to interference in the random-flows scenario as it did in the node-to-sink scenario. As in the node-sink scenario, the effect of inter-MAC interference on the average duty cycle of T-MAC rises both when the amount of overlap between the networks is increased and when the data generation interval is reduced. Moreover, as was the case for the node-to-sink scenario, T-MAC reacts in general more strongly to interference received from TDMA than it does to interference received from either of the contention-based MAC protocols. For the 5x5 deployment, the average duty cycle of T-MAC rises by at most 6.9%-points (from 17.3% to 18.5%) when receiving interference from a contention-based MAC protocol while the average duty cycle is increased by up to 14%-points (from 17.3% to 19.7%) when receiving interference from TDMA.

A similar observation can be made for the 10x10 deployment except that the gap between the performance overhead caused by interference from TDMA and the one caused by interference from the contention-based MAC protocols is more narrow when a data generation interval of 10 seconds is used. For a data generation interval of 10 seconds,

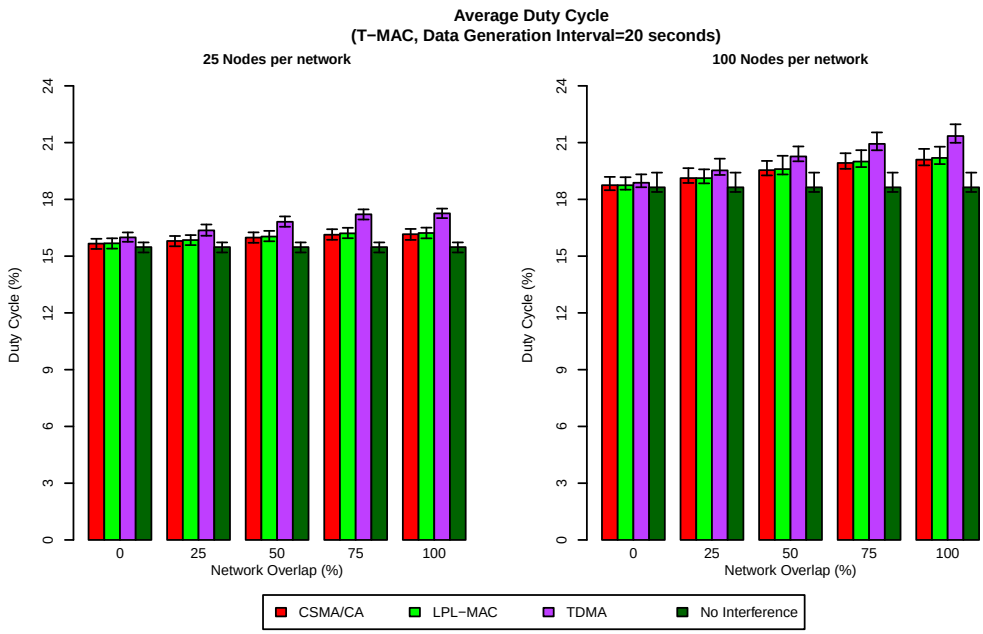


Figure 2.30: Average, 5- and 95%-tile duty cycle of T-MAC for the random-flows scenario in the presence of interference for a varying amount of network overlap.

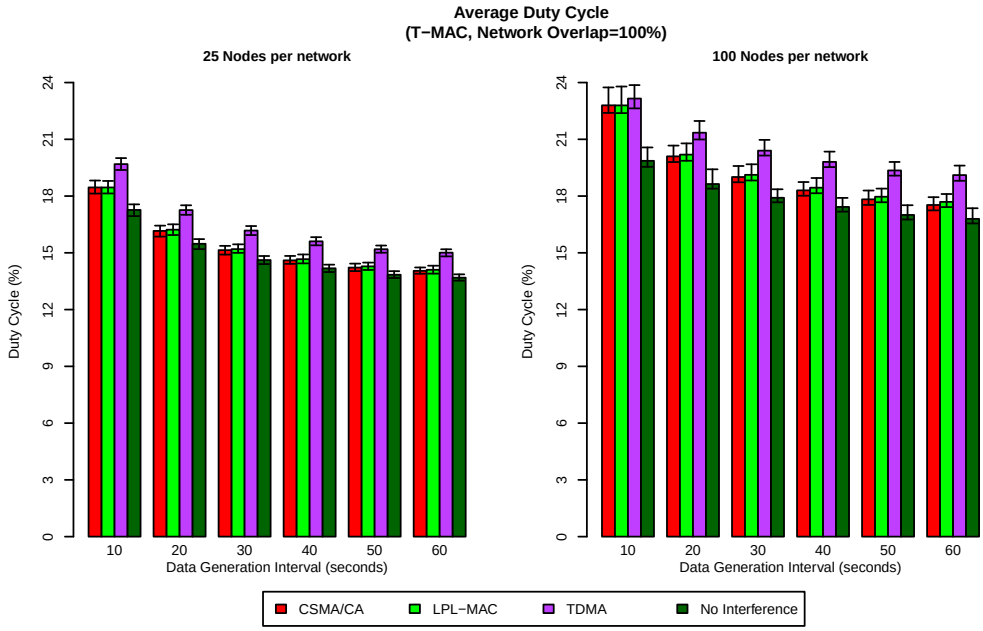


Figure 2.31: Average, 5- and 95%-tile duty cycle of T-MAC for the random-flows scenario in the presence of interference for different data generation intervals.

both CSMA/CA and LPL-MAC induce a performance overhead of around 14.8%-points (from 19.9% to 22.8%) while TDMA induces a performance overhead of, on average, 16.6%-points (from 19.9% to 23.2%). When a larger data generation interval is used, the performance overhead induced by CSMA/CA or LPL-MAC is at most 8.4%-points (from 18.6% to 20.2%) while the one induced by TDMA is at most 14.6%-points (from 18.7% to 21.4%). When these performance figures are compared to the ones obtained for the node-to-sink scenario it quickly becomes clear that there is also not much difference in the size of the incurred performance overhead. In the node-to-sink scenario the relative difference between the average 'baseline' duty cycle of T-MAC and the average duty cycle measured under interference conditions is at the very most 17%-points while for the random-flows scenario this is at most 16.6%-points.

2.4.1.3 Duty Cycle of TDMA

Figure 2.32 shows the duty cycle of TDMA for a varying amount of overlap between the networks while figure 2.33 instead shows the duty cycle of TDMA for different data generation intervals. These figures show that, as with the node-to-sink scenario, the duty cycle of TDMA is only minimally affected by inter-MAC interference. In most cases the performance overhead is well below 2%-points and even under high-interference conditions (i.e., 100% overlap and a small data generation interval) the average duty cycle rises by at most 5.2%-points (from 9% to 9.4%). This is around the same increase in duty cycle as observed for the node-to-sink scenario which shows that there is also not much difference between the two scenarios as far as duty cycle is concerned.

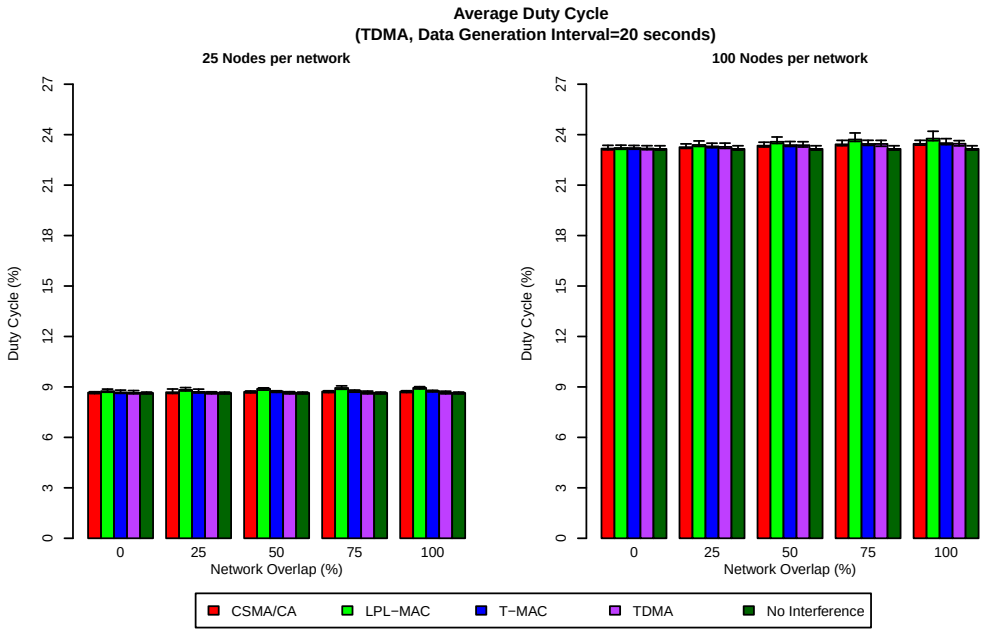


Figure 2.32: Average, 5- and 95%-tile duty cycle of TDMA for the random-flows scenario in the presence of interference for a varying amount of network overlap.

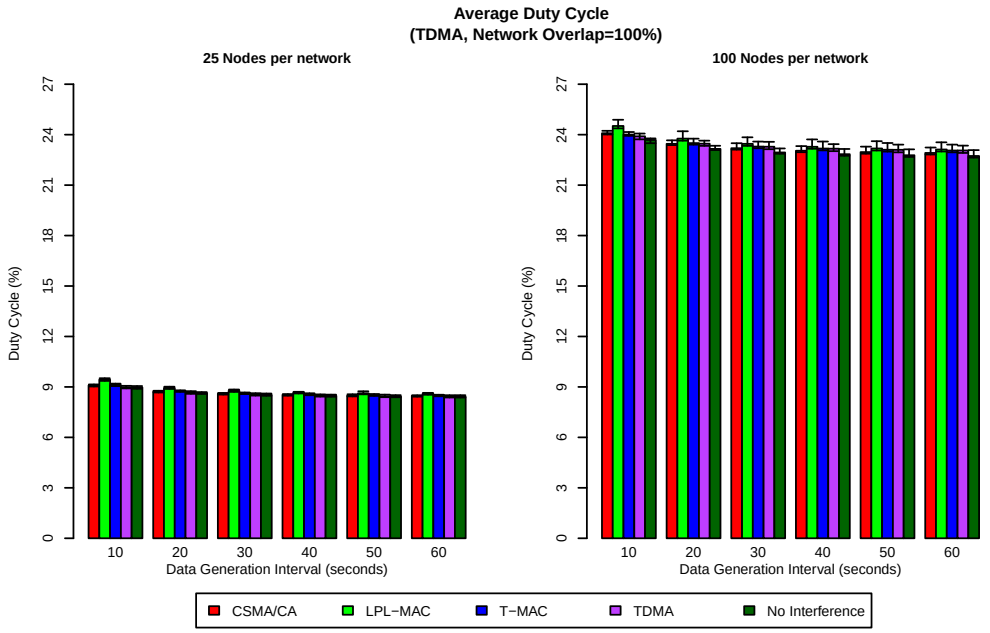


Figure 2.33: Average, 5- and 95%-tile duty cycle of TDMA for the random-flows scenario in the presence of interference for different data generation intervals.

2.4.2 Influence on the Hop Count

The average hop count of the different MAC protocols are shown for varying amounts of overlap and varying data generation intervals in figures 2.34 to 2.41. The first thing to notice in these figures is that, regardless of the interference received, the average hop counts measured for the random-flows scenario are noticeably lower than the ones measured for the node-to-sink scenario. (In the random-flows scenario the hop count is generally around 3.5 and 5.5 for respectively the 5x5 and 10x10 deployment while for the node-to-sink scenario the hop count is generally around 4 and 7). This is because in the node-to-sink scenario the sink node is located at the edge of the network, which means that a significant portion of the nodes have a relatively long path to the sink. In the random-flows scenario however, the destinations are randomly selected and thus more 'spread out' over the network which means that, on average, fewer long paths are required.

When the influence of inter-MAC interference is considered for either the CSMA/CA or LPL-MAC protocol (figures 2.34 to 2.37) it is immediately clear that, as far as hop count is concerned, these MAC protocols react in much the same way to interference as they did in the node-to-sink scenario. For the 5x5 deployment, interference only has a very small or even negligible impact on the average hop count. For the LPL-MAC protocol the same is also true for the 10x10 deployment, but for the CSMA/CA the impact of interference is more noticeable in the 10x10 deployment than it is in the 5x5 deployment. Even so, the impact of interference is only significantly higher when there is 100% overlap between the networks and when the data generation interval is sufficiently low (10 or 20 seconds).

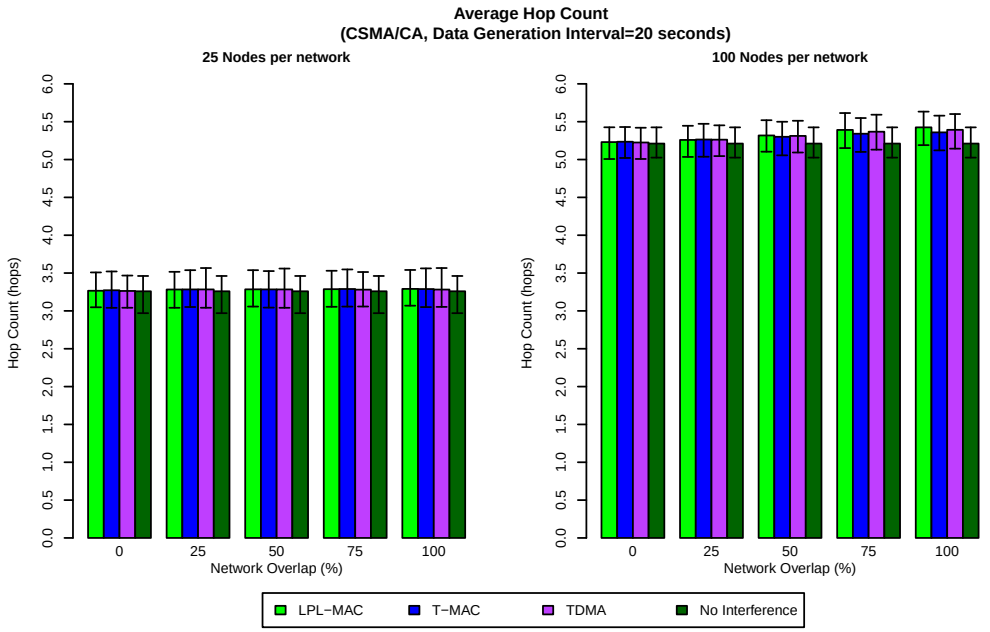


Figure 2.34: Average, 5- and 95%-tile hop count of CSMA/CA for the random-flows scenario in the presence of interference for a varying amount of network overlap.

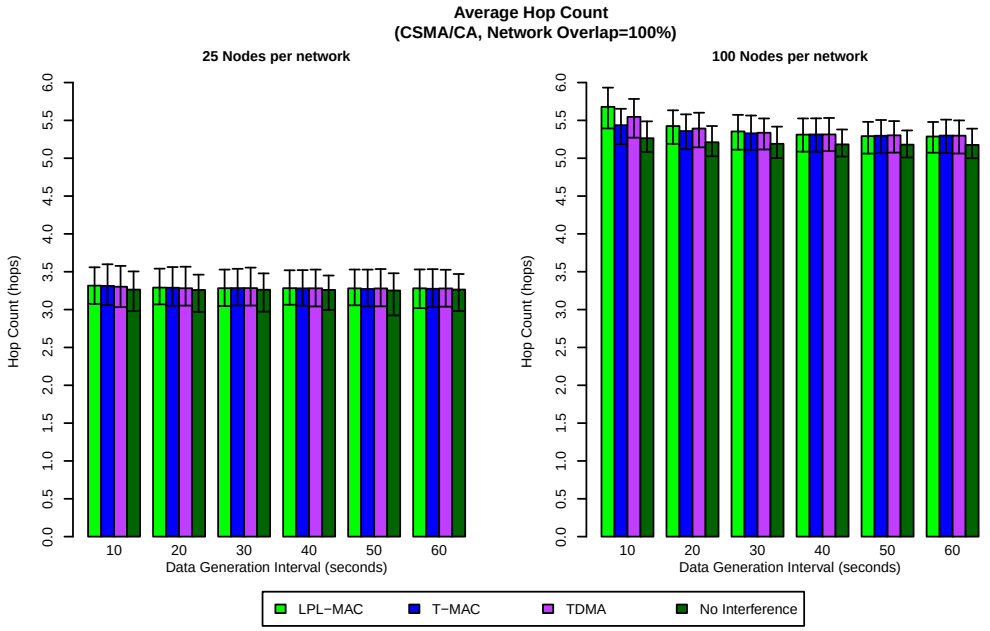


Figure 2.35: Average, 5- and 95%-tile hop count of CSMA/CA for the random-flows scenario in the presence of interference for different data generation intervals.

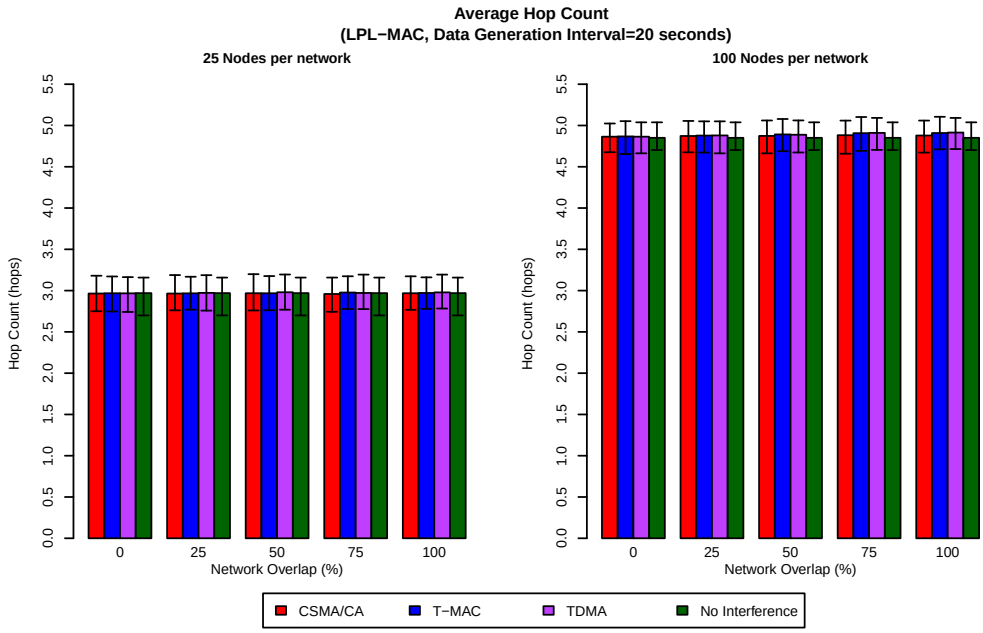


Figure 2.36: Average, 5- and 95%-tile hop count of LPL-MAC for the random-flows scenario in the presence of interference for a varying amount of network overlap.

In those cases the average hop count is altered by at most 7.9%, which is more than in the node-to-sink scenario but not dramatically so.

When the average hop counts recorded for the T-MAC protocol are considered (figures 2.38 and 2.39) is clear that these are only minimally affected by interference: over the entire range of considered parameters the average hop count of T-MAC is altered by at most 3% as a result of interference. In contrast to the node-to-sink scenario however, the average 'baseline' hop count of T-MAC does not increase when the data generation interval is reduced. This different behaviour is most likely because the sensor network applications used in these two scenarios generate different 'traffic patterns' in the network. In the node-to-sink scenario all nodes send data to one and the same sink node which not only creates a very directional traffic pattern but also causes significant congestion in the immediate area around the sink node. This, combined with the scalability issues of the T-MAC protocol, can cause a significant number of the probe messages transmitted by the routing layer to be lost which will cause the routing layer to select longer paths. In the random-flows scenario this is less of an issue given that nodes select their destinations 'semi-randomly' which will cause the different traffic flows to be more evenly distributed over the wireless network.

When the influence of inter-MAC interference is considered for the TDMA MAC protocol (figures 2.40 and 2.41) it can be observed that there is a clear difference between the 5x5 and the 10x10 deployment. For the 5x5 deployment, interference only has a very small impact on the average hop count of TDMA (which is the same behaviour as observed for the node-to-sink scenario). For the 10x10 deployment however, the impact of interference

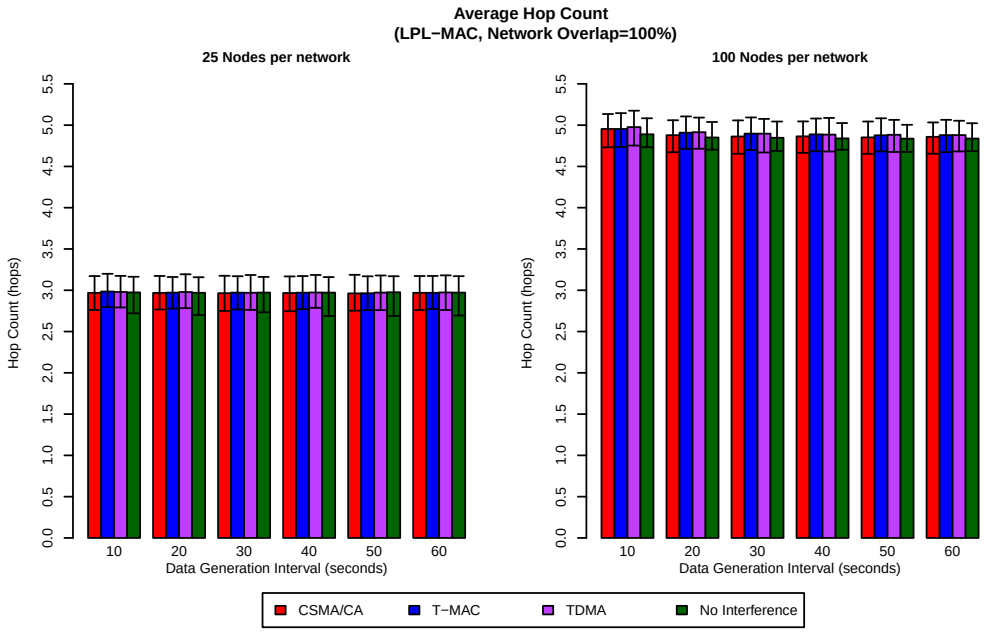


Figure 2.37: Average, 5- and 95%-tile hop count of LPL-MAC for the random-flows scenario in the presence of interference for different data generation intervals.

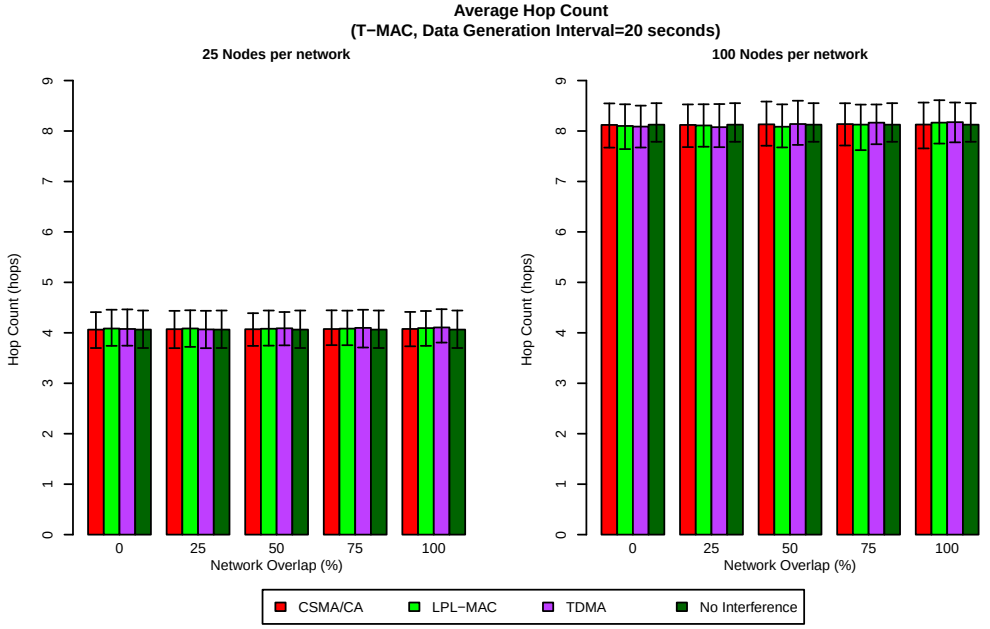


Figure 2.38: Average, 5- and 95%-tile hop count of T-MAC for the random-flows scenario in the presence of interference for a varying amount of network overlap.

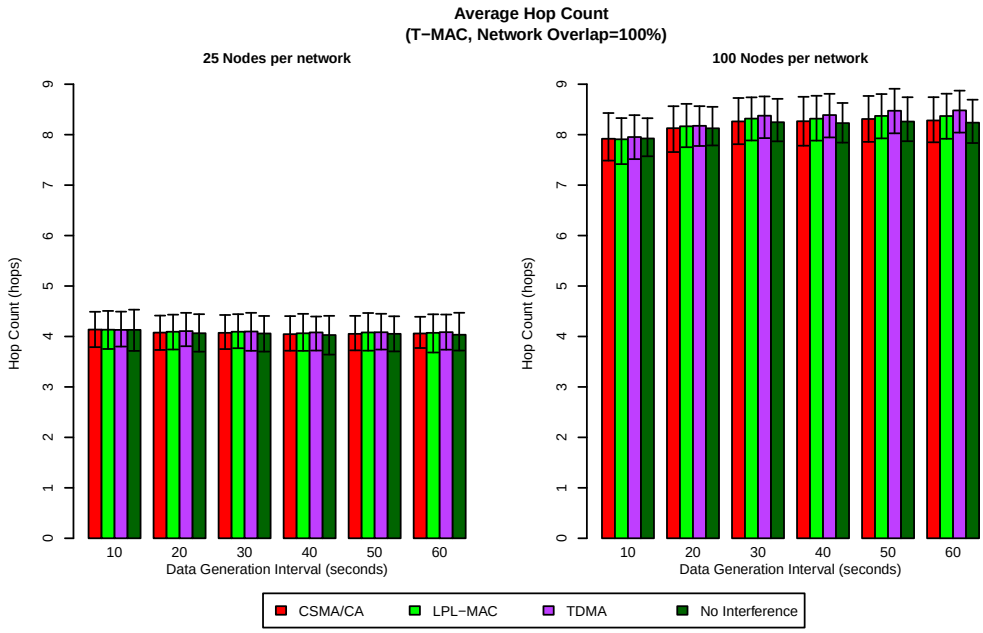


Figure 2.39: Average, 5- and 95%-tile hop count of T-MAC for the random-flows scenario in the presence of interference for different data generation intervals.

rises with the amount of overlap between the networks. For an overlap of 0%, interference still only has a negligible effect on the average hop count but when the overlap between the networks is increased the effects of interference become more noticeable but even so these are still quite small: at worst the average hop count is increased by 3.4%. When the effect of interference is considered for a varying data generation interval it can be observed that the relative difference between the average 'baseline' hop count and the average hop count measured under interference conditions rises with the length of the data generation interval. (For a data generation interval of 10 seconds the average hop count is altered by at most 1%, for a data generation interval of 60 seconds this is at most 3.4%.) This behaviour the result of the fact that, as further discussed in section 2.4.3, inter-MAC interference also has a significant impact on the end-to-end reliability of the TDMA MAC protocol and that this impact increases when the data generation interval is reduced. Given that, as discussed in section 2.3.2, hop count is measured only over packets which are correctly received and that the probability of a packet colliding with an interfering transmission rises with the number of hops between the source and destination node, this will also cause the reported hop count to be slightly reduced.

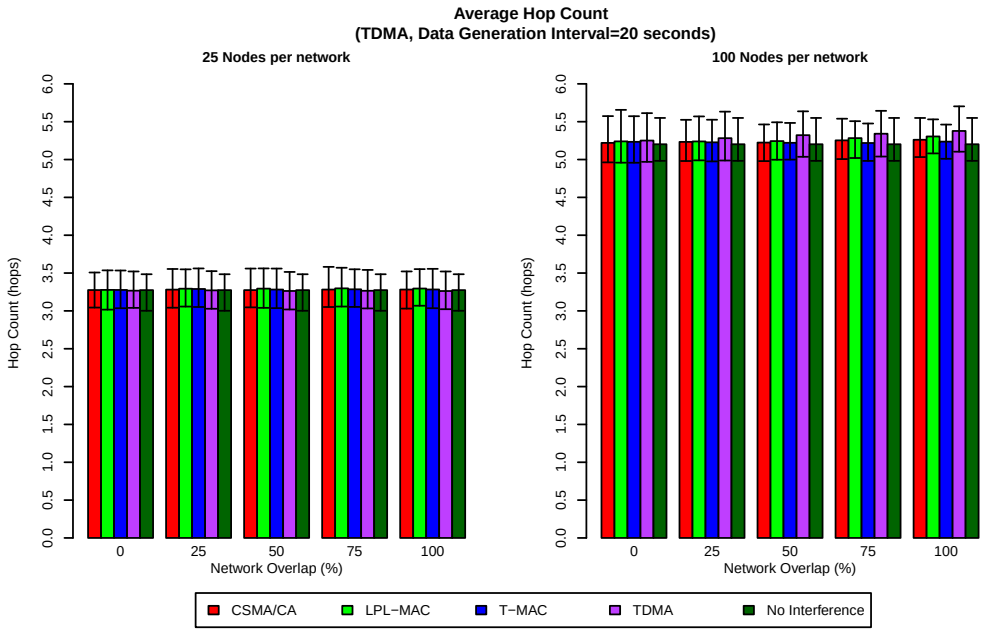


Figure 2.40: Average, 5- and 95%-tile hop count of TDMA for the random-flows scenario in the presence of interference for a varying amount of network overlap.

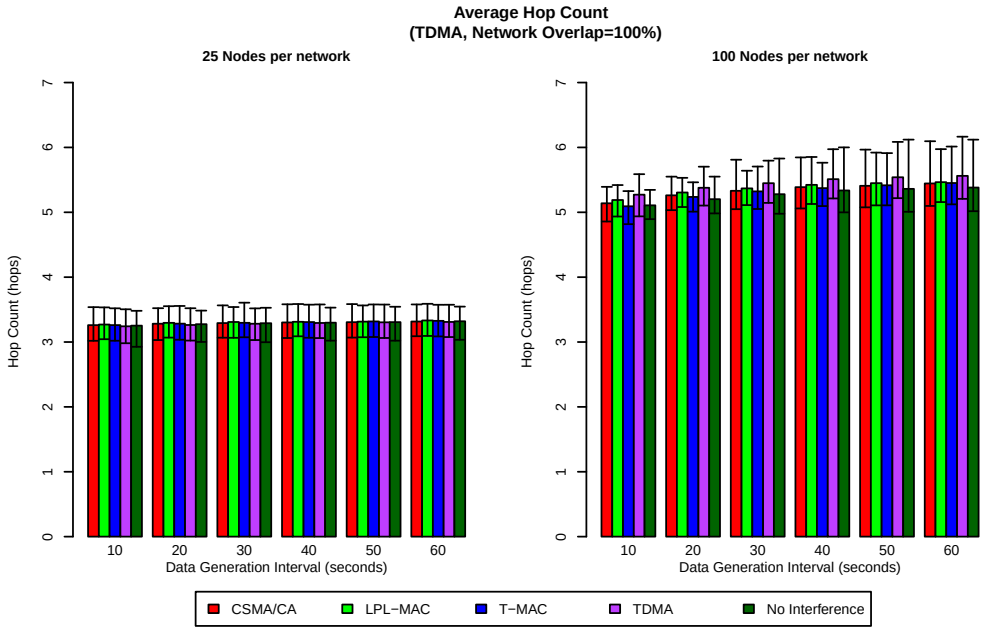


Figure 2.41: Average, 5- and 95%-tile hop count of TDMA for the random-flows scenario in the presence of interference for different data generation intervals.

2.4.3 Influence on the Reliability

2.4.3.1 Reliability of CSMA/CA

Figures 2.42 and 2.43 show the average, 5- and 95-percentile of the reliability measured for a network using the CSMA/CA MAC protocol. The first thing to be noted is that once again the reliability of CSMA/CA is not solely decided by the interference it receives from another MAC protocol. As with the node-to-sink scenario, the 'baseline' reliability of CSMA/CA will drop slightly when the data generation interval is reduced but this effect is less significant in the random-flows scenario than it is in the node-to-sink scenario. (In the node-to-sink scenario the average 'baseline' reliability drops by at most 16%-points while in the random-flows scenario the reliability is reduced by at most 7.3%-points). When the impact of interference on the reliability is considered, it can be observed that CSMA/CA reacts in much the same way to interference in the random-flows scenario as it did in the node-to-sink scenario. As with the node-to-sink scenario the relative difference between the 'baseline' reliability of CSMA/CA and the reliability measured under interference conditions rises with the amount of overlap between the networks and when the data generation interval is reduced. Moreover, it also varies with the interfering MAC protocol used, with LPL-MAC and T-MAC having respectively the largest and the smallest impact on the reliability of CSMA/CA. In addition, the individual performance figures measured for the random-flows scenario are very similar to those measured for the node-to-sink scenario.

When the amount of overlap between the networks is varied between 0% and 100%, the

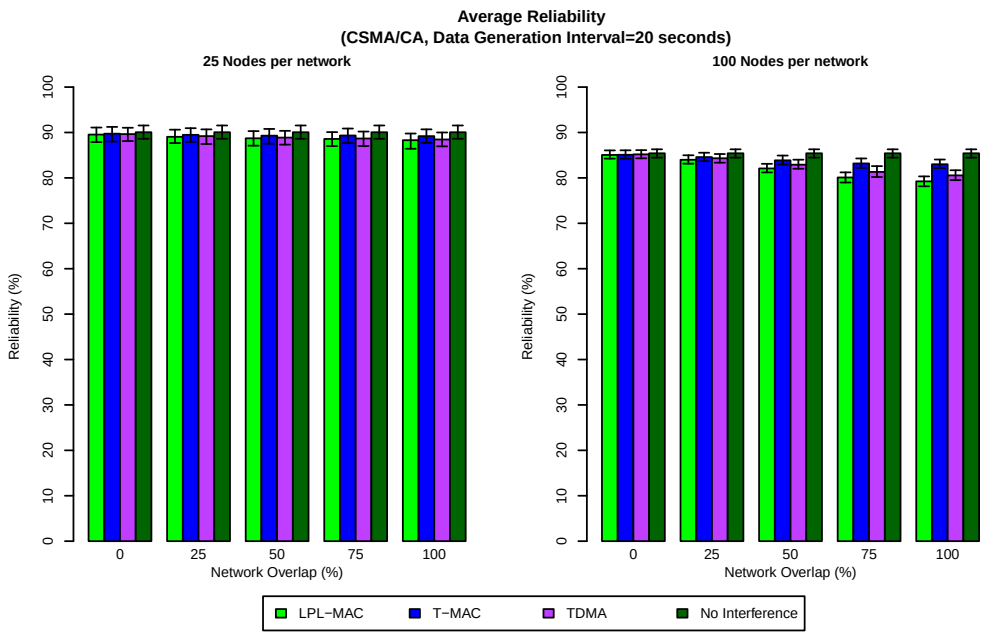


Figure 2.42: Average, 5- and 95%-tile reliability of CSMA/CA for the random-flows scenario in the presence of interference for a varying amount of network overlap.

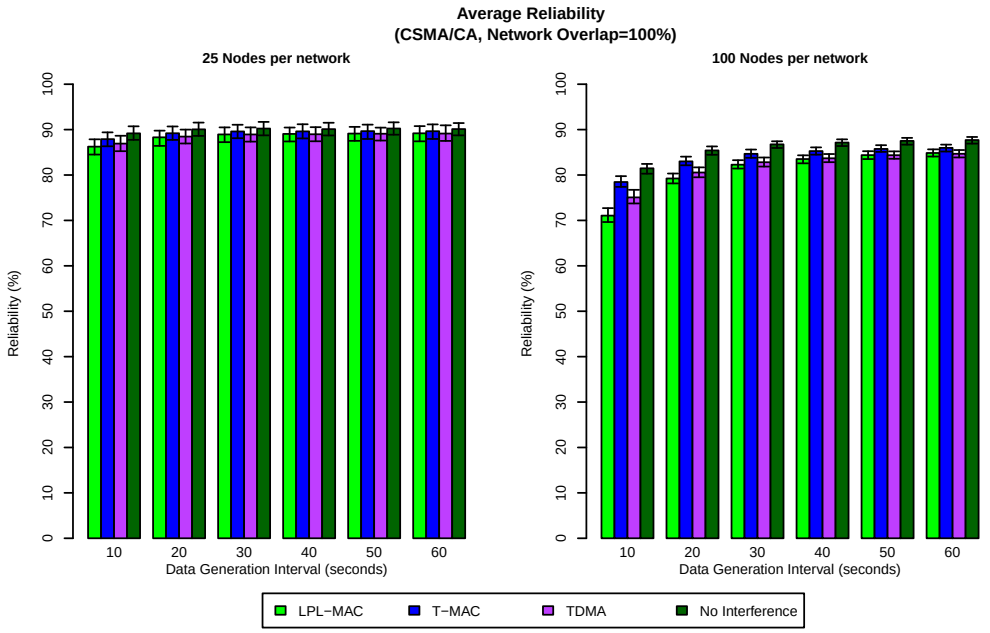


Figure 2.43: Average, 5- and 95%-tile reliability of CSMA/CA for the random-flows scenario in the presence of interference for different data generation intervals.

reliability is at most reduced by between 0 and 1.9%-points (from 90.1% to 88.3%) for the 5x5 deployment, versus 0 and 1.5%-points in the node-to-sink scenario. For the 10x10 deployment the reliability is reduced by between 0 and 6.9%-points (from 85.4% to 80.1%) versus 0 and 6%-points in the node-to-sink scenario. A similar observation can be made for the data generation interval. When this parameter is varied, inter-MAC interference will cause the reliability to drop by between 0.5%-points and 3.3%-points (from 89.2% to 86.3%) for the 5x5 deployment, versus between 0 and 2.7%-points for the node to sink scenario. For the 10x10 deployment the reliability drops by between 1.76%-points and 12.8%-points (from 81.5% to 71%), versus between 1.3 and 9.4%-points in the node-to-sink scenario.

2.4.3.2 Reliability of LPL-MAC

Figures 2.44 and 2.45 show the average, 5- and 95-percentile of the reliabilities measured for a network using the LPL-MAC protocol. From these figures it is clear that, like the CSMA/CA MAC protocol, LPL-MAC also reacts in much the same way to interference in the random-flows scenario as it did in the node-to-sink scenario. As before, the impact of interference rises with the amount of overlap between the networks and decreases when the data generation interval is reduced.

Figures 2.44 and 2.45 also show that, despite these variations in reliability, LPL-MAC is still extremely resilient to the effects of interference: over the entire range of considered test parameters, inter-MAC interference will cause the average reliability of LPL-MAC to drop by at most 3%-points (from 88.7% to 86%). In addition, the reliability of LPL-

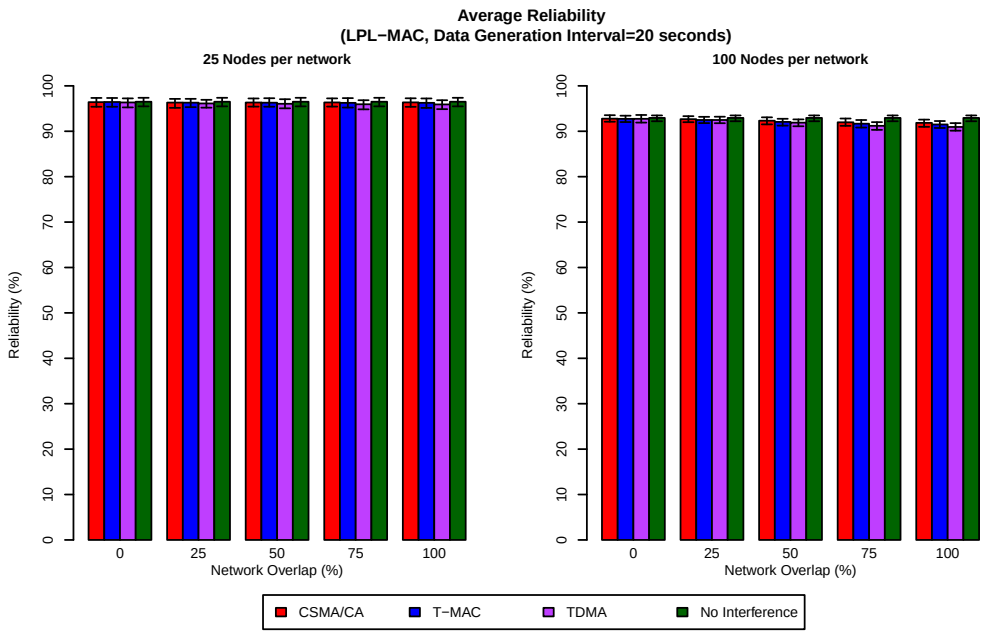


Figure 2.44: Average, 5- and 95%-tile reliability of LPL-MAC for the random-flows scenario in the presence of interference for a varying amount of network overlap.

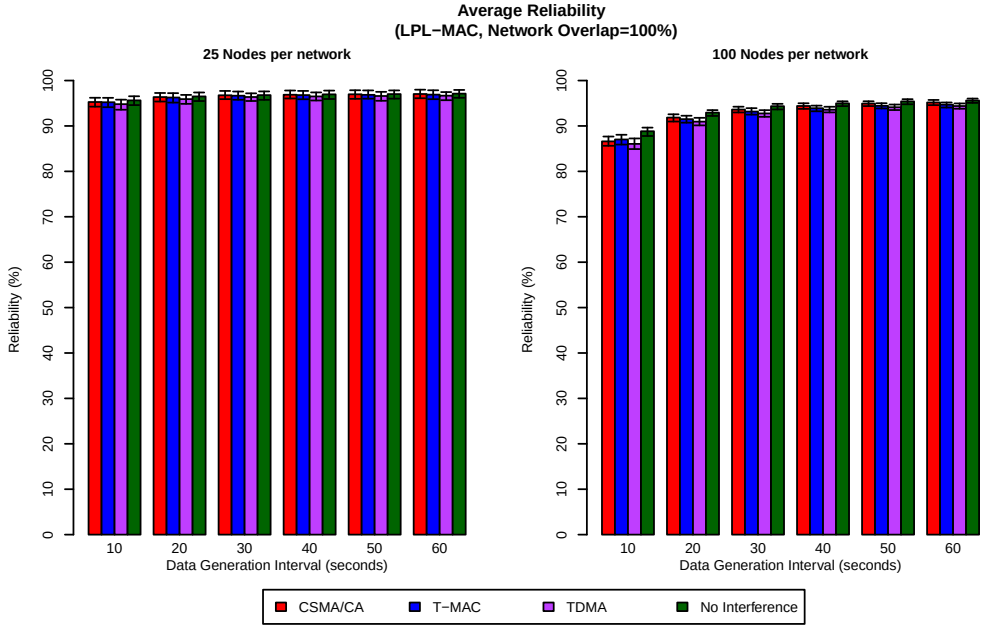


Figure 2.45: Average, 5- and 95%-tile reliability of LPL-MAC for the random-flows scenario in the presence of interference for different data generation intervals.

MAC also varies with the data generation interval even if no other network is present in the wireless environment but this effect is less pronounced in the random-flows than in the node-to-sink scenario (7.1%-points versus 10%-points).

2.4.3.3 Reliability of T-MAC

Figures 2.46 and 2.47 show the average, 5- and 95-percentile of the reliabilities measured for a network using the T-MAC protocol. From these figures it is immediately clear that T-MAC suffers from the same scalability-issues as it did in the node-to-sink scenario and that these issues severely affect the reliability in both the 5x5 and the 10x10 deployment. For the 5x5 deployment the average 'baseline' reliability of T-MAC drops from 78.5% to a little over 58% when the data generation interval is reduced from 60 seconds to 10 seconds. For the 10x10 deployment the reliability of T-MAC is at best 46.4% and can drop down to 30.6%. When the 'baseline' reliability of T-MAC is compared between the two scenarios it becomes clear that for the 5x5 deployment the 'baseline' reliability of T-MAC is higher in the random-flows than in the node-to-sink scenario. This is most likely because in the random-flows scenario the different traffic flows are more evenly spread out over the wireless environment, which means that the T-MAC protocol does not suffer from the extremely high contention around the sink area that occurs in the node-to-sink scenario. For the 10x10 deployment this is only the case when a data generation interval of 10 seconds is used. Otherwise, the baseline reliability is higher for the node-to-sink than for the random-flows scenario. This different behaviour is most likely caused by the fact that in the random-flows scenario the traffic is significantly more 'bursty' than in the node-to-sink scenario. When a sufficiently large number of nodes is used, these bursts can cause temporary peaks in the contention between the nodes which in turn affects the reliability of the network (this is the case even when a large data generation interval is used). The reason that this does not result in a lower reliability for a data generation interval of 10 seconds is most likely because in that case the effect of these temporary peaks in contention is overshadowed by the effect of the continuous high level of contention that occurs in the immediate area around the sink node in the node-to-sink scenario.

When the effect of inter-MAC interference on the reliability of T-MAC is considered it is clear that, as with the node-to-sink scenario, the reliability of T-MAC is more affected by interference when there is a large rather than a small amount of overlap between the networks. In addition, the reliability is a bit more affected by interference in the random-flows scenario than it was in the node-to-sink scenario. For the 5x5 deployment, the reliability of T-MAC drops by between 1.9%-points (from 69.4% to 68.1%) and 6.8%-points (from 69.4% to 64.7%) depending on the amount of overlap between the networks, versus a drop in reliability of between 0 and 5%-points in the node-to-sink scenario. For the 10x10 deployment, the reliability drops by between 1.5%-points (from 35.8% to 35.2%) and 14.6%-points (from 35.8% to 30.6%), versus 0 and 10.3%-points in the node-to-sink scenario. When the effect of interference is instead considered for a varying data generation interval, it is clear that for the 5x5 deployment the effect of interference increases when the data generation interval is reduced and that the reliability is once again a bit more affected in the random-flows scenario than it is in the node-to-sink scenario. When the data generation interval is reduced from 60 to 10 seconds, the performance overhead resulting from interference increases from 4.7%-points (from 78.5% to 74.7%)

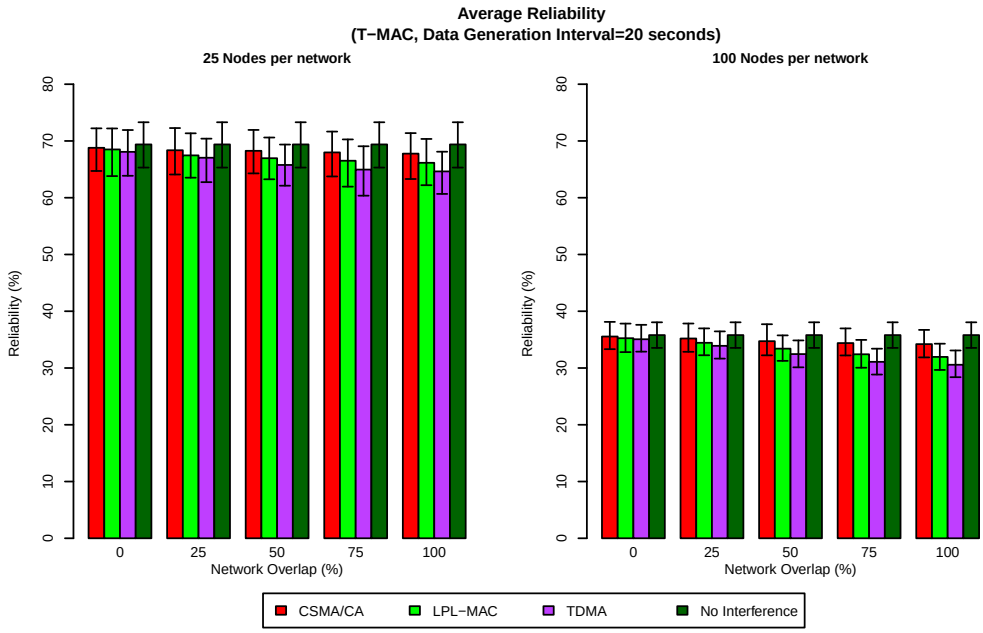


Figure 2.46: Average, 5- and 95%-tile reliability of T-MAC for the random-flows scenario in the presence of interference for a varying amount of network overlap.

to 8.4%-points (from 58.3% to 53.4%), versus 3.2 to 7.36%-points for the node-to-sink scenario.

A similar observation can be made for the 10x10 deployment except in the case that T-MAC is combined with TDMA. When these two MAC protocols are combined, the reliability of T-MAC is slightly more affected when a large rather than a small data generation interval is used. Given that in this case altering the data generation interval only has a small effect on the overhead resulting from interference (the relative difference in reliability varies between 15 and 17%-points), this different behaviour may be attributed to the fact that the 'synchronisation traffic' of TDMA will have a proportionally higher impact on the reliability when the data generation interval is large rather than small. When T-MAC receives interference from a contention-based MAC protocol, the same behaviour as in the 5x5 deployment is observed except that in the worst case the reliability is less affected by interference in the random-flows than it is in the node-to-sink scenario. In the random-flows scenario the reliability of T-MAC is reduced by between 3.7%-points (from 46.4% to 44.7%) and 13.6%-points (from 30.6% to 26.4%), versus between 2 and 16%-points for the node-to-sink scenario.

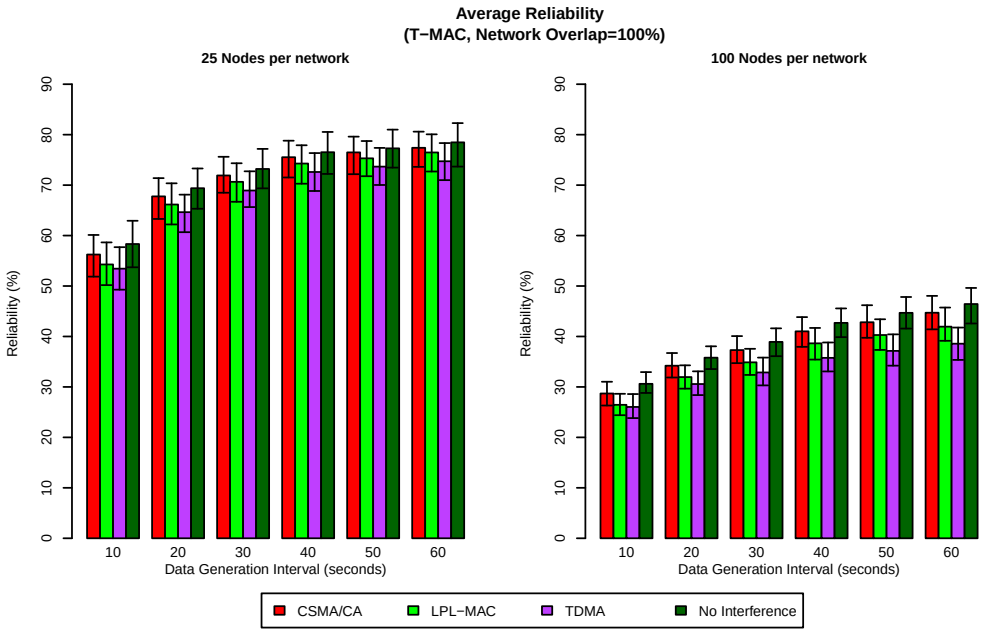


Figure 2.47: Average, 5- and 95%-tile reliability of T-MAC for the random-flows scenario in the presence of interference for different data generation intervals.

2.4.3.4 Reliability of TDMA

The reliability of the TDMA MAC protocol is shown in figures 2.48 and 2.49 for respectively a varying amount of overlap between the networks and a varying data generation interval. These figures show that, as with the node-to-sink scenario, there is a lot more variation in the reliability measured for individual test runs than for the other considered MAC protocols. In addition, it should also be noted that even when TDMA is not under the influence of inter-MAC interference, the reliability will suffer from the fact that under high network-load conditions certain nodes are not assigned enough 'tx-slots' to transmit all packets in time. In contrast to the node-to-sink scenario however, the reliability of TDMA is only affected when a data generation interval of 10 seconds is used for the 10x10 deployment and even then the reliability is only reduced by 7.5%-points (from 76% to 70.8%) which is about the same as for the CSMA/CA MAC protocol (see above). This reduced sensitivity to the network load is caused by the fact that in the random-flows scenario the traffic is more evenly spread out over the network.

When considering the influence of inter-MAC interference, it can be observed that the effect of this interference on the reliability of TDMA increases both when the amount of overlap between the networks is increased and when the data generation interval is reduced. While this is the same behaviour as observed in the node-to-sink the scenario, figures 2.26 and 2.48 make it plain that the reliability is more affected by inter-MAC interference in the random-flows scenario than in the node-to-sink scenario. When there is 0% overlap between the networks, inter-MAC interference will cause the reliability to drop by at most 2.5% (from 82.7% to 80.7%) for the 5x5 deployment while for the

10x10 deployment the reliability is reduced by 3.4%-points (from 76.6% to 73.9%). When the amount of overlap is increased to 100% however, the reliability is reduced by 9.3%-points (82.7% to 75%) and 34.4% (76.6% to 53.3%) for respectively the 5x5 and 10x10 deployments. For the 5x5 deployment the impact of interference is not that much higher than in the node-to-sink scenario (where the performance overhead varies from 0 to 8.9%-points), but for the 10x10 deployment the difference between the scenarios is much more dramatic (for that deployment the performance overhead only varies between 0 and 13.9%-points in the node-to-sink scenario). A similar observation can be made when the effect of interference is considered for a varying data generation interval (figure 2.49). For the 5x5 deployment the reliability is reduced by between 1.7%-points (from 82.7% to 81.3%) and 15.3%-points (from 83.4% to 70.7%) depending on the MAC protocol and data generation interval used. (In the node-to-sink scenario the performance overhead varies between 0 and 15%-points). For the 10x10 deployment the reliability is reduced by between 9.7%-points (from 74.3% to 67%) and 37.8%-points (from 70.8% to 44.1%) which is significantly more than in the node-to-sink scenario (for which the performance overhead varies between 1.2 and 18%-points).

From these performance figures it is clear that TDMA is less resilient to inter-MAC interference in the random-flows scenario than in the node-to-sink scenario. This is due to the fact that the sensor network applications used in these two scenarios create different 'traffic patterns' in the network. In the node-to-sink scenario all traffic flows in the network are directed to a single sink node located at the edge of the network. Since the sink nodes used by the respective networks are located at opposite sides of the wireless environment this effectively means that, for each network involved, the area of the network that is most affected by interference is also the area where the least amount of transmissions of the network itself are taking place. Conversely, in the area of the network where most of the transmissions are taking place (near the sink) the interference received from the other network will be comparatively lower. This has the effect of somewhat reducing the effect of interference on the reliability (especially for the 10x10 deployment). In the random-flows scenario however, the destination nodes are semi-randomly selected, which means that the traffic flows are more evenly spread out over the wireless environment. Because of this, there is a higher probability that interfering transmissions will collide with one another and as a result inter-MAC interference will have a larger effect on the reliability of the networks involved.

When the impact of interference is compared between the different interfering MAC protocols it is clear that, as in the node-to-sink scenario, interference from the LPL-MAC protocol has a much more significant impact on the reliability of TDMA than the other two contention-based MAC protocols. When LPL-MAC is used to generate interference, the reliability of TDMA drops by at most 37.8%-points (from 70.8% to 44.1%) as a result of interference while for CSMA/CA and T-MAC the average reliability drops by at most 23.3%-points (from 70.8% to 54.3%). For the case where both networks in the wireless environment use the TDMA MAC protocol, the interference between the networks only has a negligible effect on the reliability for the 5x5 deployment. As with the node-to-sink scenario however, earlier tests revealed that swapping the dynamic route selection mechanism with one that uses pre-calculated static routes causes the reliability of TDMA to drop dramatically. As before, this indicates that the relatively minor impact of interference observed for the 5x5 deployment is not due to the properties of the TDMA

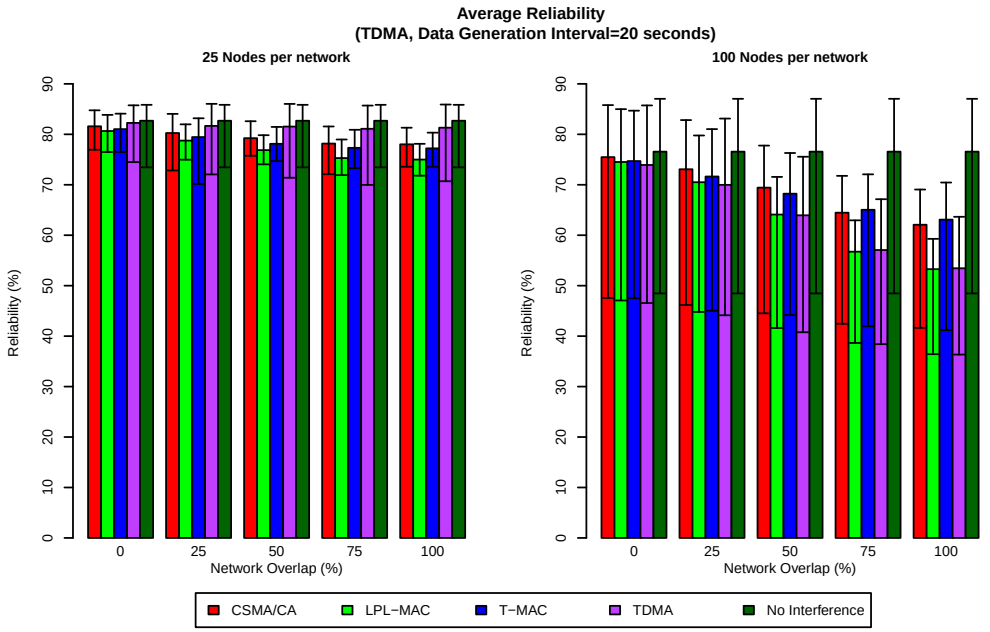


Figure 2.48: Average, 5- and 95%-tile reliability of TDMA for the random-flows scenario in the presence of interference for a varying amount of network overlap.

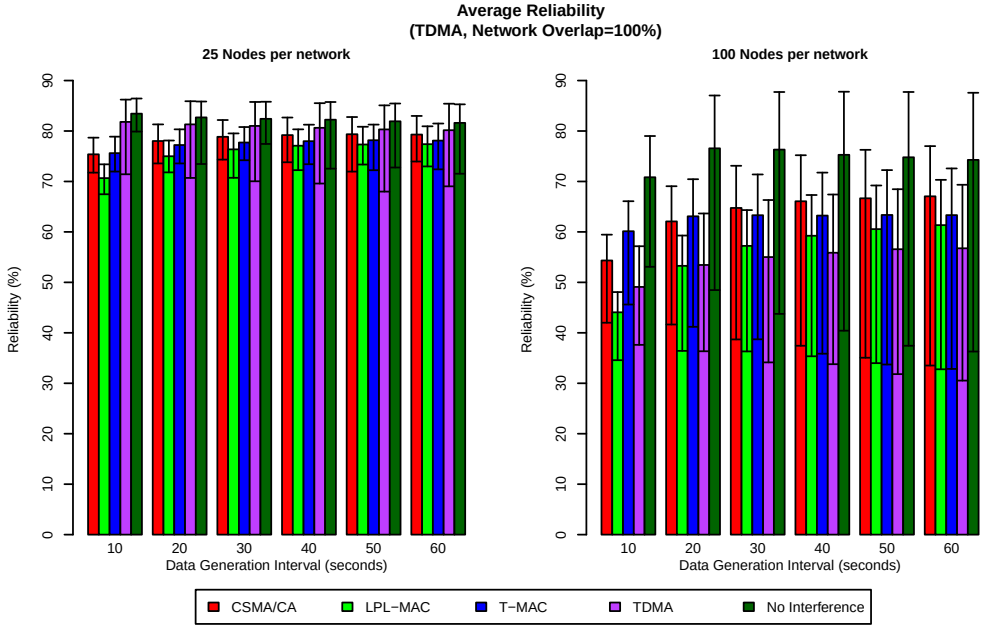


Figure 2.49: Average, 5- and 95%-tile reliability of TDMA for the random-flows scenario in the presence of interference for different data generation intervals.

MAC protocol itself but is instead the result of the route selection mechanism selecting paths that do not interfere with one another. In contrast to the node-to-sink scenario, combining two TDMA MAC protocols does however have a very significant effect on the reliability for the 10x10 deployment. In that case, the average reliability can drop by more than 30%-points as a result of interference. This larger susceptibility to interference is most likely caused by the fact that, in the random-flows scenario, the interfering traffic flows are evenly spread out over the same wireless environment rather than being aimed in opposite directions. Because of this, the route selection mechanism has fewer (if any) options for avoiding links that interfere with one another and as a result the reliability will be more affected by inter-MAC interference.

2.5 Conclusion

This chapter investigated how the network performance is affected by interference when two MAC-heterogeneous sensor networks are deployed in the same wireless environment. After first providing a summary of currently existing MAC protocols (section 2.1) and subsequently discussing the simulator and test setup used (section 2.2), the effects of inter-MAC interference were investigated separately for both the node-to-sink and the random-flows scenario.

From the performance evaluation in sections 2.3 and 2.4, it is clear that inter-MAC interference can have a significant impact on the performance of the networks involved but that the extent to which the performance is affected depends on a number of factors. Both the traffic rate of the networks, as well as the amount of overlap between the geographical areas in which the networks are deployed, significantly affect the extent to which the network performance is reduced as a result of inter-MAC interference. When the traffic rate is low and there is little to no overlap between the networks, the performance of the networks is, in general, only minimally affected by interference. Unsurprisingly, when the traffic rate and amount of overlap are increased, the effect of interference on the network performance grows as well.

When the impact of interference is considered for the individual performance metrics, it turns out that inter-MAC interference mostly affects the duty cycle and the end-to-end reliability. The average hop count of the networks is in most cases only minimally affected by inter-MAC interference (max 2% difference) and even in those cases where the effect of inter-MAC interference is more noticeable the average hop count is altered by, at the very most, 7.8%.

When the impact of interference is considered for the duty cycle metric it is clear that, with the exception of CSMA/CA, TDMA is the least affected by inter-MAC interference. Over the entire range of considered parameters the average duty cycle rises by at most 5.2%-points. For the T-MAC protocol, the effects of inter-MAC interference are more significant, with the duty cycle rising by at most 17%-points as a result of interference. For the LPL-MAC protocol the situation is even worse, with the duty cycle rising by as much as 30%-points for the 5x5 deployment and 50%-points for the 10x10 deployment. While a performance overhead this large only occurs under worst case conditions (i.e., 100% overlap and a data generation interval of 10 seconds), LPL-MAC is also quite susceptible to inter-MAC interference when a lower amount of network overlap or a lower

traffic rate is used. While this in itself does not hamper the operation of the LPL-MAC network, it does mean that sensor network developers (or administrators) need to be aware that even under low to moderate interference conditions, inter-MAC interference will have a significant effect on the energy consumption of nodes using the LPL-MAC protocol.

When the impact of interference is considered for the reliability metric it quickly becomes clear that, in contrast to the duty cycle metric, LPL-MAC is extremely resilient towards inter-MAC interference, with the reliability dropping by at most 3.2%-points as a result of interference. The CSMA/CA and T-MAC protocol are significantly less resilient to interference with the reliability dropping at most by respectively 12.8%-points and 15.8%-points because of interference. Although, as expected, TDMA is the most susceptible to inter-MAC interference of all considered MAC protocols, the reliability of TDMA is less affected by interference than originally anticipated. When TDMA is combined with either LPL-MAC, T-MAC or CSMA/CA, this can be attributed to the CCA (and back-off) mechanisms used by these contention-based MAC protocols. In the case that TDMA is combined with another TDMA MAC protocol this can, as discussed in sections 2.3.3 and 2.4.3, be attributed to the routing layer avoiding links that interfere with one another. All-in-all, this means that in most cases the performance overhead resulting from interference is significant (at most 17%-points), but not overly dramatic. The only exception is when the random-flows scenario is used for the 10x10 deployment in which case inter-MAC interference can cause the reliability of TDMA to drop by nearly 38%-points.

When the impact of interference is compared between the node-to-sink and the random-flows scenario, it can be observed that, although for some cases interference has a larger effect on the network performance in one scenario than in the other one, the MAC protocols still react in mostly the same way to variations in either the amount of overlap between the networks, the traffic rate or the specific interfering MAC protocol used. The only noteworthy exception is when two TDMA MAC protocols are combined for the 10x10 deployment in which case the reliability of the TDMA MAC protocol is much more affected by inter-MAC interference in the random-flows than in the node-to-sink scenario. (As discussed in section 2.4.3, this is due to the difference in traffic patterns between these scenarios).

Although the effects of interference thus vary with the overlap, data rate and exact combination of MAC protocols used, the graphs shown in sections 2.3 and 2.4 also make it plain that of all the considered parameters, the *number of nodes present in each network* has by far the most profound effect on how the performance changes as a result of interference. When each network uses 25 nodes (the 5x5 deployment), the performance drops at most by around 15% as a result of interference (the only exception to this is the duty cycle of LPL-MAC). While in this case the performance overhead incurred is not insignificant it can still be considered to be within acceptable bounds for most sensor network applications. When 100 nodes are used per network however (10x10 deployment), this is no longer the case given that for this deployment the performance of the networks is much more impacted and regularly drops by more than 30% as a result of interference.

While the results discussed in sections 2.3 and 2.4 thus make it plain that there are a number of cases in which the effects of interference would have a dramatic effect on the network performance, these results also show that under low to moderate traffic (and

interference) conditions the impact of inter-MAC interference is small enough to allow them to co-exist in the same wireless environment without much issue. Given that, as discussed above, co-existence is a pre-requisite for communication, this also paves the way for enabling communication between the MAC-heterogeneous sensor networks involved. Since the remaining chapters all deal with the mechanisms required to enable communication between these networks, all (simulation-based) performance tests discussed in these chapters will use a test-setup for which the effects of inter-MAC interference have been deemed low enough, based on the results of this chapter, to allow them to co-exist and thus, potentially, communicate with one another. More specifically, a network size of 25 nodes per network is used with 50% overlap between the networks and a data generation interval of 20 seconds.

CHAPTER 3

Virtual Gateways

The previous chapter investigated the effects of interference between MAC-heterogeneous sensor networks and showed that, under low to moderate interference conditions, the impact of inter-MAC interference on the performance of the networks is sufficiently small to allow them to co-exist in the same wireless environment. This chapter considers the problem of how to enable communication between these networks. As one might expect, the major issue with doing so is that the MAC protocols used by the respective networks are not compatible with one another and that there is thus no interoperability between the networks. The traditional manner to achieve interoperability between wireless networks is to enforce the use of a standardised PHY- and MAC-layer in case that all networks are based on the same wireless technology and to deploy gateway nodes equipped with multiple radio interfaces to enable communication between nodes based on entirely different technologies (such as for instance between WiFi and cellular networks).

In the case of sensor networks, the standardisation approach only offers a partial solution. The PHY-layer of IEEE 802.15.4 has been relatively well adopted but the specific MAC protocol used, tends to vary with the requirements of the use case for which the sensor network has been developed. As previously discussed in section 2.1, this is because the requirements imposed by the various sensor network use cases are in general too diverse to allow them to be met by a single (standardised) MAC protocol. Another issue with the standardisation approach is that it can only be used to achieve interoperability for 'future' networks. It cannot be used to enable communication between sensor networks that already exist. The second approach of introducing gateways into the network seems to be a more viable solution since it would allow each network to continue using its own MAC protocol. Unfortunately, the deployment of gateway nodes equipped with multiple radio interfaces would require the development of specialised hardware. As a result, this approach is infeasible for most scenarios.

This chapter therefore investigates an alternative approach for providing MAC-layer interoperability between heterogeneous sensor networks. This approach is based on the fact that sensor network MAC protocols generally operate on top of a single, standardised PHY-layer: the one specified in IEEE 802.15.4. Instead of using regular gateways to enable communication between MAC-heterogeneous sensor networks, it is proposed to use so-called *Virtual Gateways*. On a regular gateway node each MAC protocol operates on top of a separate radio interface. On a *Virtual Gateway* all MAC protocols make use of a single, shared radio interface. Virtual gateways have a number of advantages compared to normal gateways. Firstly, they only require a single radio interface and therefore do not require the development of specialised hardware. Secondly, since every node in a sensor network is already equipped with a radio interface, it can be configured as a virtual gateway by performing a software upgrade. There is no need to add new nodes. Moreover, these nodes can also dynamically enable or disable specific MAC protocols depending on the requirements of the nodes and the traffic conditions in the wireless environment.

To investigate the feasibility of this approach, a network stack capable of running multiple MAC protocols simultaneously was developed. This '*MultiMAC*' network stack was developed for, and evaluated on, real sensor nodes. The reason for using real sensor nodes rather than simulations is that (wireless) network simulators tend to have a somewhat simplistic view of the CPU-time required to process packets at the different layers of the network stack. This in turn causes them to be rather optimistic about the processing power needed to allow the network stack to run smoothly. The Castalia simulator discussed in section 2.2 is no exception. For the other chapters in this thesis this is not an issue since the performance tests in those chapters are mainly concerned with the behaviour of MAC protocols on a network-wide scale and not with the processing requirements of the individual nodes. Within the scope of this chapter however, the hardware requirements (and thus the processing overhead) of this MultiMAC stack will be an important factor in deciding whether or not virtual gateways are a viable method for enabling communication between MAC-heterogeneous sensor networks and as a result the virtual gateway approach needs to be evaluated using real sensor nodes instead of simulations.

The architecture and implementation of the *MultiMAC* stack are discussed in more detail in section 3.1 while in section 3.2 the performance of this network stack is evaluated. In addition, a number of initial tests were also performed on the w-iLab.t [140] testbed to get an idea of how MAC-heterogeneous sensor networks interact with one another in the presence of virtual gateways. These tests are further discussed in section 3.3. Finally, the conclusions of the feasibility study are presented in section 3.4.

3.1 The MultiMAC network stack

The MultiMAC stack was developed from scratch for the Tmote Sky platform [50] in TinyOS 2.1.0. In order for virtual gateways to be a feasible solution for enabling interconnectivity between MAC-heterogeneous networks, the MultiMAC network stack must be able to meet the following requirements:

Minimal overhead: In sensor networks, energy efficiency is in most cases significantly

more important than interoperability. This means that the cost of introducing interoperability between networks should be as small as possible in order for the benefits of interoperability to outweigh the costs, which in turn means that the performance overhead of supporting multiple MAC protocols should be minimal. Moreover, to ensure that the virtual gateway approach can be used with a sufficiently wide selection of sensor nodes, the (minimum) hardware requirements of the MultiMAC stack should also be sufficiently low. In this respect the Tmote Sky sensor nodes used here can be considered to be a good benchmark since they only contain a 4MHz microcontroller with 10K RAM and 48K flash and are thus extremely resource constrained. If the MultiMAC stack can be made to work on these nodes, it should also work on a wide range of more recent and more powerful hardware.

Flexibility: To ensure that the virtual gateway approach is generally usable, the MultiMAC network stack should be flexible enough to accommodate a wide range of MAC protocols. On low-power sensor nodes, the range of MAC protocols that can be supported is mainly dependent on how well both the hardware and the operating system are able to meet the timing requirements of the different MAC protocols. Given that contention-based MAC protocols generally have less stringent timing-requirements than TDMA-based MAC protocols it should come as no surprise to find the former class of MAC protocols to be better supported than the latter (both TinyOS and Contiki for instance only provide contention-based MAC protocols by default). Given however that TDMA-based MAC protocols represent a significant portion of all currently available sensor network MAC protocols, the MultiMAC stack is required to support both contention-based and TDMA MAC protocols within the scope of this thesis.

Extensibility: Virtual gateways would not be a feasible solution for enabling interoperability if the entire network stack would need to be rewritten from scratch each time a MAC protocol is added or removed from the code base. Consequently, developers should be able to easily extend the MultiMAC network stack with new protocols. Moreover, the protocols in the MultiMAC stack must be isolated from each other as much as possible to ensure that MAC developers do not need to take the presence of other MAC protocols into account.

3.1.1 Packet Format

One of the prerequisites for using virtual gateways is that nodes must be able to distinguish between packets sent by different MAC protocols. This is currently not possible due to the wide variety of sensor network MAC protocols that are available and the low level of standardisation between the packet formats used. To circumvent this problem, the MultiMAC stack imposes a number of limitations on the packet formats that can be used. Firstly, MAC protocols are required to use a packet format that is compatible with IEEE 802.15.4. This requirement should not pose too great a problem since most sensor network MAC protocols use a packet format that is, mostly due to technical limitations, already largely compatible with that of IEEE 802.15.4. Moreover, this packet format is very flexible and does not even require any addressing information to be present in the packet. Secondly each MAC protocol is assigned a unique *MAC-id*. As shown in figure 3.1, this MAC-id is stored in one of the reserved fields of the MAC header. Because of this, including the MAC-id in the packet does not incur any additional overhead. While, due

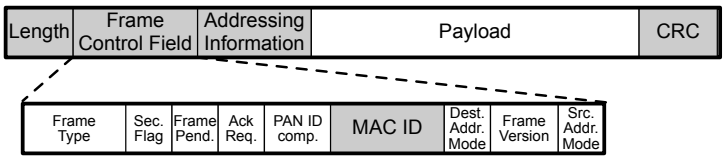


Figure 3.1: The packet format used by the MultiMAC network stack based on the IEEE 802.15.4 packet format. The *MAC-id* of the MAC protocol is stored in one of the reserved fields of the IEEE 802.15.4 MAC header.

to the size of the MAC-id field, at most 8 different MAC protocols can be distinguished at one time, it should be noted that the MAC-id does not have to be globally unique. It must only be unique within the wireless environment in which the virtual gateways reside. Given that the use of virtual gateways requires some sort of agreement to be made between the administrators of the respective sensor networks, the MAC-ids to use can be selected at that time as well. In the specific case of symbiotic networks, the MAC-ids to use can be decided as part of the negotiation process between the networks (see section 1.3.3).

3.1.2 Architecture

The architecture of the MultiMAC network stack is outlined in Figure 3.2.

The *CC2420 Driver* is responsible for all interactions with the CC2420 radio chip. It manages the transmission and reception of data packets and is also responsible for turning the radio on and off. One of the most important features of this component is that it allows MAC protocols to control certain aspects of how frames are transmitted and received. Not only can MAC protocols enable or disable Clear Channel Assessments (CCA) when

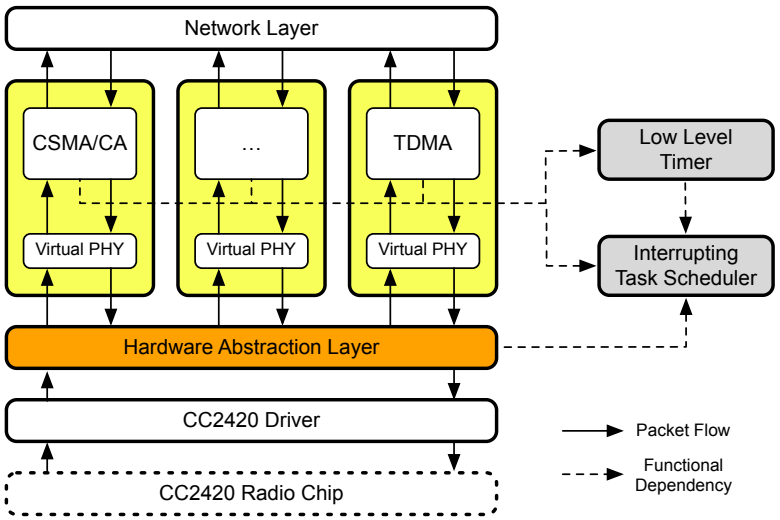


Figure 3.2: The Architecture of the MultiMAC network stack.

sending a frame, but they are also able to decide whether or not the transmission of a frame may be delayed in order to allow an incoming frame to be received. This flexibility is needed to support both contention-based and time-division based MAC protocols. Contention-based MAC protocols will usually require CCA-checks to be performed and can generally allow the transmission of a frame to be delayed. Time division-based MAC protocols on the other hand have very strict timing requirements and will therefore usually require the frame to be transmitted without any delay. Depending on the specific TDMA MAC protocol used, it may also be necessary to send packets without performing a CCA-check beforehand.

The *Hardware Abstraction Layer* (HAL) operates on top of the CC2420 Driver and is responsible for 'multiplexing' the different MAC protocols in the network stack on top of the radio interface. When a packet is received by the HAL, it will dispatch the packet to the correct MAC protocol based on the MAC-id stored in the packet. When a MAC protocol wishes to transmit a packet, the HAL will only pass that packet on to the CC2420 Driver if no other transmission is currently in progress. Otherwise, the HAL reports to the MAC protocol that the channel was busy. Although in this case one of the MAC protocols is prevented from transmitting its packet, this is still more preferable to what would have happened if both MAC protocols had accessed the same channel using separate radio interfaces. In that case either the CCA-check would have failed in one of the two radio interfaces or both packets would have been transmitted and would have consequently collided with each other. The only advantage of using two radio interfaces is that these radio interfaces could be configured to operate on different channels. Using two radios would however consume significantly more energy than multiplexing multiple MAC protocols on top of a single interface. A similar mechanism is used for enabling or disabling the radio interface. Each MAC protocol can enable or disable the radio through the Virtual PHY interface, but the radio will only be actually disabled if all MAC protocols have disabled the radio.

The HAL is also capable of performing address recognition and sending automatic acknowledgements for each MAC protocol individually. Normally, the CC2420 radio chip is able to perform these functions in hardware and, as a result, the delay between the reception of a correct MAC packet and the transmission of the acknowledgement is only 128μs. Unfortunately, the MAC-id of these hardware generated acknowledgements is always equal to zero, which means that the HAL is not able to discern between acknowledgements of different MAC protocols. As a result, hardware acknowledgements cannot be used in the MultiMAC stack. By transmitting acknowledgements from the HAL rather than from the MAC implementation itself, it was possible to reduce the ACK-turnaround time to less than 800μs. This increase in ACK-turnaround time is mostly due to the time that is needed to transfer the packets between the microcontroller and the radio chip. On other platforms that are equipped with a microcontroller with an embedded radio chip, such as the STM32W, it should be possible to reduce this additional delay even further.

As mentioned before, the MAC protocols in the network stack are each presented with a *Virtual PHY* interface that can be used to perform a number of PHY-layer operations such as sending and receiving packets and turning the radio on and off. To ensure that the different MAC protocols remain isolated from one another, they can only access the HAL through this interface. Although the Virtual PHY interface provides most operations

required for implementing a MAC protocol, some operations have been excluded from this interface on purpose. MAC protocols are not able to alter the transmission power or change the channel of the radio interface, since these operations affect all the other MAC protocols in the network stack. The available operations are, as described above, 'multiplexed' by the HAL so isolation between the MAC protocols is preserved.

In addition to the components that are directly involved with packet processing, the MultiMAC stack also provides an *Interrupting Task Scheduler* as an alternative to the default task scheduler provided by TinyOS. The reason for including this scheduler is that TinyOS does not have any support for real-time processing. By default, TinyOS uses a scheduling mechanism whereby tasks are executed on a first-come-first-served basis and can only be interrupted by interrupt handlers. For time-critical MAC protocols this mechanism does not suffice, since there is no upper bound on the execution time of tasks and as a result time-critical tasks may have a large queuing delay. Moreover, running time critical tasks from an interrupt handler to ensure timeliness is not a valid approach. Doing so would prevent other interrupt handlers from being executed, which is detrimental to the correct operation of the hardware drivers.

In contrast to the standard task scheduler of TinyOS, the interrupting task scheduler allows so-called 'Interrupting Tasks' to be scheduled with a number of different priorities. These tasks are able to interrupt both TinyOS tasks and other interrupting tasks with a lower priority. To achieve this, the currently running task is suspended by triggering a software interrupt. The interrupting task scheduler then selects the task with the highest priority and executes it directly from the interrupt handler of the software interrupt. Prior to starting the new task however, the interrupts of the microcontroller, which were automatically disabled when the software interrupt was triggered, are re-enabled. This ensures that interrupting tasks can not only be interrupted by other interrupting tasks with a higher priority but also by genuine hardware interrupts. This task scheduler thus allows MAC protocols to execute time-critical tasks without disrupting the hardware drivers. In addition, this task scheduler is also used by the HAL of the MultiMAC stack. By default, the HAL uses an 'Interrupting Task' to relay events (such as the reception of a packet) to the different MAC protocols. This not only minimises the amount of processing performed in interrupt-context by the HAL itself, but also means that MAC-developers have to be less concerned about minimising the amount of processing performed in the event-handlers called by the HAL.

The addition of the interrupting task scheduler to the MultiMAC stack also prompted a redesign of the interface to the hardware timers. The reason for this is that the default TinyOS timers can only process events either from the interrupt handler of the underlying hardware timer or by using the standard TinyOS task scheduler. As discussed above, neither of these options is well suited for MAC-layer processing. Although it would have been possible to adapt the TinyOS timers to also use the interrupting task scheduler, it should be noted that these timers are used in a wide range of components beside the network stack. As a result, this would have required these components to be altered as well. Another problem with the default timer implementation is that it is quite complex which means that, even when processing events in the interrupt handler itself, the time between the hardware interrupt being triggered and the scheduled event being processed is simply too large for these timers to be used by time-critical MAC protocols. Instead of altering the existing timer-implementation, a new, smaller, simpler and more efficient

interface to the hardware timers was therefore developed and added to the codebase. This *Low Level Timer* implementation not only has a lower latency than the original TinyOS implementation but since it is capable of processing events using the interrupting task scheduler, it allows the priority of scheduled events to be tuned to the requirements of the specific MAC protocol. It should also be noted that this low-level timer interface is not intended to replace the one provided by TinyOS. Instead, the two timer implementations can be used side by side but each one is sourced from a different hardware timer.

3.1.3 Implemented MAC protocols

In order to meet the *Flexibility* and *Extensibility* requirements outlined above, the MultiMAC stack must be able to support a wide range of MAC protocols. Given the number of sensor network MAC protocols currently available (see section 2.1), it is not possible to implement all of them. Similar to the tests performed in chapter 2, only a limited number of 'representative' MAC protocols are therefore implemented. More specifically, a CSMA/CA, an LPL-MAC and a TDMA MAC protocol were implemented for the MultiMAC stack. Due to the high implementation cost involved, the T-MAC protocol is not considered in this chapter. As far as channel access is concerned however, this protocol has around the same timing requirements as CSMA/CA and, as far as synchronisation is concerned, the timing requirements of T-MAC are less stringent than that of TDMA. This means that if the MultiMAC stack can support both the CSMA/CA and TDMA MAC protocols, it should also support the T-MAC protocol.

The CSMA/CA MAC protocol developed for the MultiMAC stack is based on the one provided by TinyOS but has been implemented from scratch. The reason for doing so, is that the original implementation provided by TinyOS is quite complex, spread out over different modules, and is partially integrated with the radio driver. As a result, it is not a good starting point for building a clean, simple and efficient implementation of the CSMA/CA MAC protocol. Once the *Virtual PHY* interface had been fully defined, this newly constructed CSMA/CA implementation was then used as a simple test protocol during the remainder of the development of the MultiMAC stack.

After the MultiMAC stack had been completed, the LPL-MAC and TDMA MAC protocols were implemented. As with the CSMA/CA MAC protocol, the LPL-MAC protocol considered in this chapter uses the same channel access mechanism as the one provided by TinyOS but does not share any code with the TinyOS implementation. In order to preserve the isolation between the MAC protocols, the 'MultiMAC' LPL-MAC implementation was, like CSMA/CA, also written from scratch as a 'standalone' module. This is in contrast to the TinyOS version of LPL-MAC which is implemented as a separate layer on top of CSMA/CA.

The TDMA MAC protocol developed for the MultiMAC stack is similar to the TDMA MAC protocol used in chapter 2, but for simplicity some changes have been made. Firstly, the TDMA MAC protocol considered here treats SYNC-slots in exactly the same manner as DATA-slots, which means that all time slots have the same length. These time slots are organised into superframes of 167 time slots so every superframe is around one second long. In addition, the 'early sleeping'-mechanism discussed in section 2.1.7 is not implemented which means that the TDMA MAC protocol considered here will keep the radio enabled for an entire RX-slot even if no transmission is received. Instead of using the median-

based synchronisation technique described in section 2.1.4, nodes are synchronised using a simple 'synchronisation tree'. (This, in essence, means that every node is assigned a specific 'synchronisation parent' to whose SYNC-messages it will synchronise its clock and that a single 'schedule master' node is selected to be the root of the synchronisation tree.) While, as discussed in section 2.1.4, this method of synchronisation can cause the synchronisation error to accumulate with the number of hops between the schedule master and the node, this turned out not to be an issue for the specific performance tests considered in this chapter.

From a MAC-developer point of view, implementing the CSMA/CA and LPL-MAC protocol proved to be a relatively simple exercise that did not reveal any flexibility or extensibility issues with the MultiMAC network stack. Although implementing the TDMA MAC protocol was initially expected to be more troublesome, this turned out not to be the case and instead the development of the TDMA MAC protocol also proved to be relatively straightforward. This is most likely due to the functionality provided by the MultiMAC stack to support the development of time-critical MAC protocols. The fact that implementing the different MAC protocols thus proved to be quite simple shows that the MultiMAC stack is sufficiently *flexible* to support a wide range of MAC protocols and that it is *extensible* enough to allow MAC protocols to be easily implemented. The *Overhead* of the MultiMAC stack is discussed below.

3.2 Performance evaluation of the MultiMAC stack

The performance of the MultiMAC stack (and thus its performance overhead) is investigated using both 'single MAC' and 'multiple MAC' performance tests. As the name suggests, the 'single MAC' performance tests investigate the overhead of the MultiMAC stack in the case that only a single MAC protocol is used whereas the 'multiple MAC' performance tests investigate the performance of the MultiMAC stack when the node in question is being used as a virtual gateway.

3.2.1 Single MAC protocol performance

To determine the 'single MAC' performance overhead of the MultiMAC stack, its performance is compared to that of the TinyOS and 'Passthrough' network stack. The 'Passthrough' stack is derived from the MultiMAC stack by replacing the HAL component with a component that passes the calls from the MAC protocol directly to the CC2420 Driver and vice versa. Unlike the TinyOS network stack, the Passthrough network stack thus has exactly the same codebase as the MultiMAC stack, apart from the different HAL. Any difference in performance between these stacks is therefore the result of the multiplexing that is performed by the MultiMAC HAL.

Three different metrics are considered in these performance tests: throughput, delay (round trip time) and duty cycle. As in the previous chapter, duty cycle is used as a measure for the energy consumption but in this chapter it is used to measure the energy efficiency of the network stack rather than the energy consumption of the MAC protocol. Given that the performance measured for these three metrics also depends on how the MAC protocol is configured (for LPL-MAC the length of the listen and sleep interval

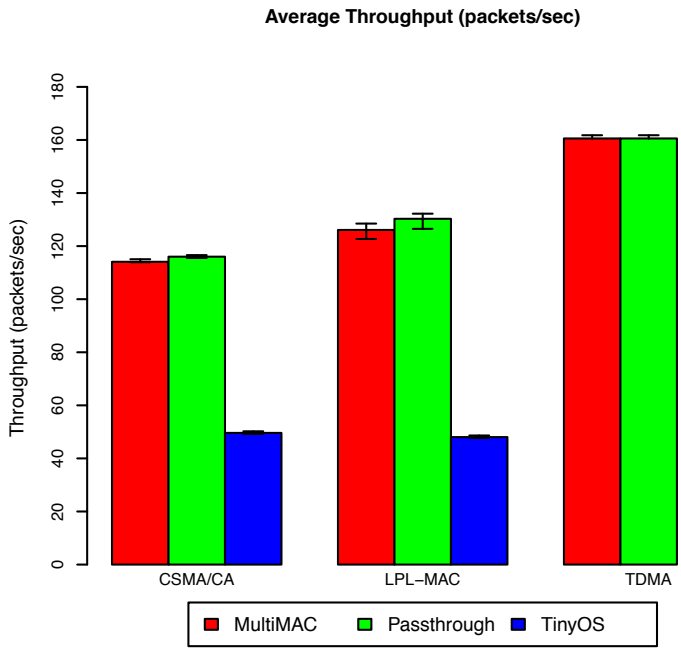


Figure 3.3: Throughput measured over a single link with different MAC protocols and network stacks.

will for instance have a significant impact on both the duty cycle and the delay), the configuration of each MAC protocol is fixed over all test cases.

Throughput is measured by continuously sending maximum-sized data frames from one node to another over a single link, for 60 seconds. The number of successfully transmitted packets is recorded at the end of each test run and this test is repeated 20 times. Delay is measured by sending 'ping' and 'ping-reply' messages between two nodes once every four seconds and recording the round trip time. This test is repeated 300 times. While measuring the round trip time, the duty cycles of both nodes are also measured. To this end both nodes are connected to a logic analyser which is able to capture the precise moments that the radio is turned on and off. After measuring the round-trip time for a specific test scenario, the duty cycle over the entire test run is calculated.

In figure 3.3 the single-hop throughput measured between two nodes running either the TinyOS, Passthrough or MultiMAC stack is shown for the CSMA/CA, LPL- and TDMA MAC protocols. As with the remaining figures in this chapter, the bars represent the average over all performed test runs while the whiskers display the 5- and 95-percentile of the collected measurements. When the CSMA/CA and LPL-MAC protocols are considered, it is immediately clear that both the MultiMAC and the Passthrough network stack are able to achieve a significantly higher throughput than the TinyOS network stack. The TinyOS network stack only manages to process at most around 50 packets per second while the MultiMAC and Passthrough network stack are able to process packets at more than twice the rate. When the throughput of the MultiMAC stack is compared to that of the Passthrough network stack, it is shown that, for the CSMA/CA and LPL-MAC

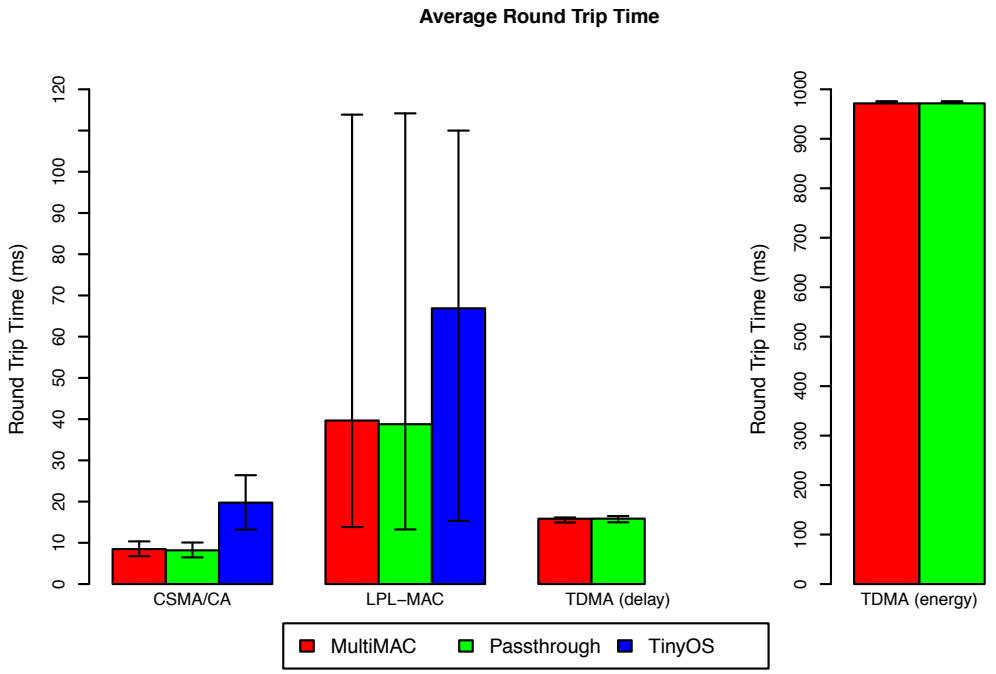


Figure 3.4: Round trip time measured over a single link with different MAC protocols and network stacks. For an optimal view of the data obtained, the measurements are shown in two separate graphs with a different scale on the Y-axis.

protocols, the throughput of the Passthrough network stack is only marginally larger than that of the MultiMAC stack. The difference in performance is 1.65% and 3.32% for respectively the CSMA/CA and LPL-MAC protocols. For the TDMA MAC protocol the throughput achieved using the MultiMAC stack is exactly the same as the one achieved using the Passthrough network stack. This however is hardly surprising given that for this MAC protocol the number of packets that can be sent is determined by the assigned slot allocation and not by the performance of the network stack.

Figure 3.4 shows the round trip time measured between two nodes using either the CSMA/CA, LPL-MAC or TDMA MAC protocol. For the TDMA MAC protocol, the round trip time is highly dependent on which slot allocation is used. Therefore the round trip time and duty cycle were measured both with a slot allocation optimised for minimal delay and a slot allocation optimised for minimal energy consumption. When the round trip time is considered for the MultiMAC and Passthrough network stack, it is clear that for all three considered MAC protocols, the round trip time is slightly higher for the MultiMAC stack than for the Passthrough network stack but that the difference between the two is very small (3.5% for CSMA/CA, 2.5% for LPL-MAC and less than 0.1% for TDMA). Moreover, both the MultiMAC and the Passthrough network stack perform significantly better than the TinyOS network stack, which has a round trip time that is on average 133% larger than that of the MultiMAC stack when the CSMA/CA MAC protocol is used while for the LPL-MAC protocol the round trip time of the TinyOS network stack is 68% higher than that of the MultiMAC stack. It should also be noted that for the

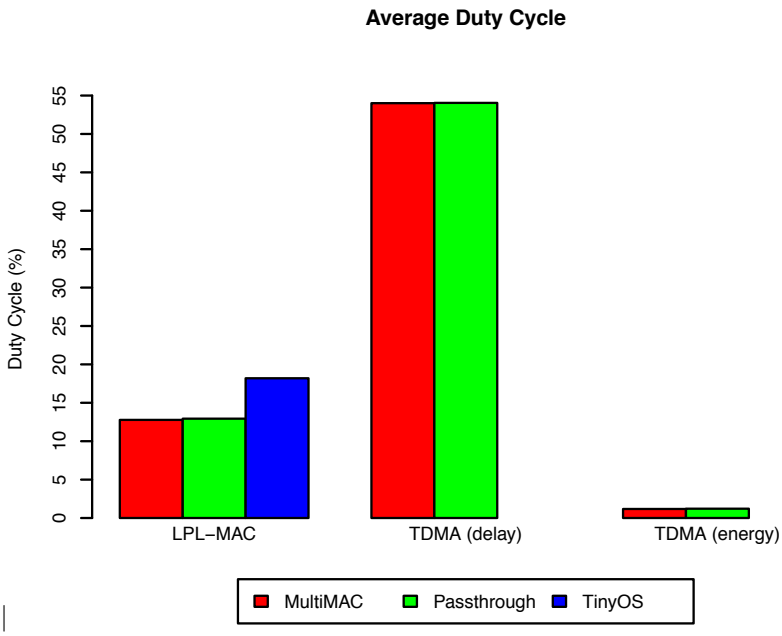


Figure 3.5: Average Duty Cycle measured on two nodes with different MAC protocols and network stacks.

LPL-MAC protocol, there is a lot more variation in round trip time than for the other MAC protocols. This is because, when using the LPL-MAC protocol, nodes disable their radios for most of the time and it takes a variable amount of time to wake these nodes up.

The duty cycle of the nodes is shown in Figure 3.5. Since the duty cycle could only be measured over the entire test run of all round trip time measurements, only a single data point was collected for each test scenario and as a result there are no whiskers shown in figure 3.5. Despite this, the true duty cycle should be accurately reflected since these values were measured over a relatively long time period (20 minutes). The duty cycle for the CSMA/CA MAC protocol is always 100% and is therefore not shown. As with the round trip time measurements, the duty cycle of the MultiMAC stack is slightly higher than that of the Passthrough stack but the difference is once again very small. For the LPL-MAC protocol the difference between the two is only 1.3%-points while for the TDMA MAC protocol the difference is at most 3.2%-points (1.17% versus 1.21%). Moreover, both network stacks significantly outperform the TinyOS network stack when only the duty cycle of the LPL-MAC protocol is considered. In that case, the duty cycle of the TinyOS network stack is more than 40%-points higher (18.2% versus 12.8%) than that of the MultiMAC network stack.

3.2.2 Multiple MAC protocol performance

To measure the impact of running multiple MAC protocols on a single node, a test setup with three nodes is used whereby a single ‘router node’ routes packets between two ‘end

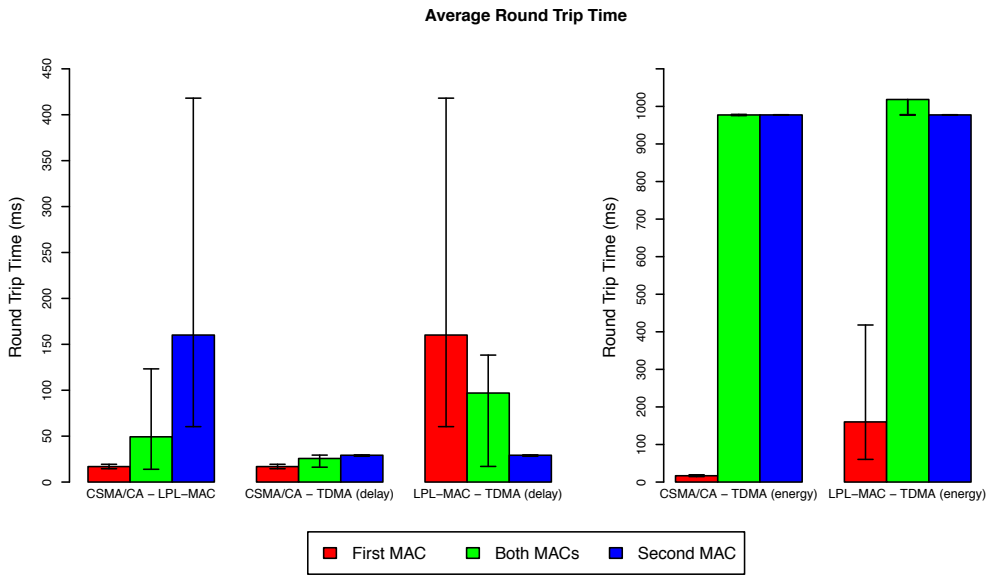


Figure 3.6: Round trip time measured over two links with the same and different MAC protocols. For an optimal view of the data obtained, the measurements are shown in two separate graphs with a different scale on the Y-axis.

nodes'. When both end nodes use the same MAC protocol, the router node is equipped with the same MAC protocol and acts as a regular router. When the end nodes use different MAC protocols, the router node is equipped with both MAC protocols and acts as a virtual gateway. For these tests the round trip time is measured between the two end nodes. As with the 'single MAC' tests discussed above, this is done by sending 'ping' and 'ping-reply' messages between the two end nodes once every 4 seconds. At the same time, the duty cycle of the *router node itself* is also measured. By comparing the measurements for the case where only a single MAC protocol is used to the case where two MAC protocols are used, the overhead of running multiple MAC protocols can be determined.

The measured round trip times are shown in figure 3.6. For each category, the first and third value show the round trip time when all three nodes are using either the first or the second MAC protocol. The middle value shows the round trip time when both MAC protocols are being used at the same time. As with the 'single MAC' tests, the round trip time and duty cycle of the TDMA MAC protocol are highly dependent on which slot allocation is used. Therefore, these metrics were measured both with a slot allocation optimised for minimal delay and a slot allocation optimised for minimal energy consumption. This figure shows that for most combinations of MAC protocols, the round trip measured when using a virtual gateway lies somewhere between the round trip times measured when only a single MAC protocol is used. This behaviour is to be expected given that, when using a virtual gateway, each MAC protocol covers a part of the routing path between the two end nodes and that the delay of the packets sent over this path thus depends on the properties of both of these MAC protocols. Since in the test setup used here, each MAC protocol is used for exactly one of the two links between the two

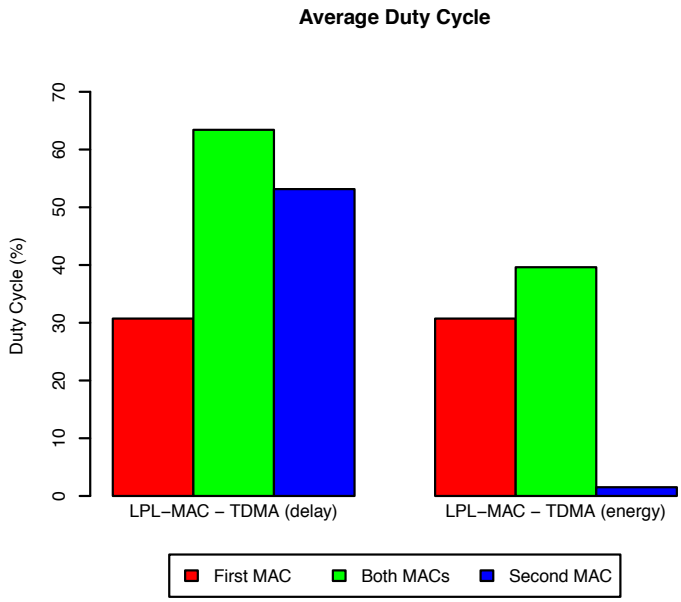


Figure 3.7: Duty Cycle of a Virtual gateway running both a single and two different MAC protocols.

end nodes one might moreover expect the round trip time measured when using a virtual gateway to be pretty close to the average of the round trip times measured for each of the individual MAC protocols. This however is only the case when LPL-MAC is combined with TDMA and a slot allocation optimised for delay is used. In all other cases the round trip time measured when using virtual gateways is noticeably closer to the round trip time measured for one MAC protocol than it is to the round trip time measured for the other one. This shows that the round trip time when using virtual gateways not only depends on the specific MAC protocols used but also on how these specific MAC protocols interact with one another.

The only case for which the round trip time is larger when using a virtual gateway than when using a single MAC protocol is when the LPL-MAC protocol and TDMA MAC protocol (optimised for energy consumption) are combined. In that case, this higher round trip time can be attributed to the fact that, because of the high delay incurred by the energy-optimised slot allocation, the originator of the 'ping' packet (running LPL-MAC) will already have gone back to sleep by the time the 'ping-reply' packet arrives at the virtual gateway. As a result, this node needs to be woken up before the 'ping-reply' packet can be delivered which incurs an additional delay. Even so, the round trip time when using a virtual gateway is only 4% higher than when using solely TDMA, which is not negligible but still very small.

The measured duty cycle is shown in figure 3.7. As with the 'single MAC' tests, the duty cycle is always 100% when using CSMA/CA and as a result these measurements are not shown. From the measurements show in this figure it is clear that using more than one MAC protocol on a single node has a noticeable effect on the duty cycle. This

increase in duty cycle is to be expected since there is more than one MAC protocol that may want to keep the radio enabled. When using a slot-assignment optimised for delay, the duty cycle is around 19%-points higher when using a virtual gateway than when using only the least energy efficient MAC protocol (63.4% versus 53.1%). When using a slot-assignment optimised for energy consumption, the difference is even higher with the duty cycle measured for the virtual gateway setup being 29%-points higher (39.6% versus 30.7%) than the duty cycle measured for the least energy efficient MAC protocol. Given that the *absolute* difference in duty cycle is actually slightly lower for the energy-optimised than the delay-optimised slot allocation (8,9% instead of 10,3%), this larger *relative* difference is partially due to 'baseline' the duty cycle being lower in this case.

Although for both slot allocations the difference in duty cycle is thus not insignificant, it should be noted that this difference is not solely due to the 'overhead' of running multiple MAC protocols. As discussed in chapter 2, even without any virtual gateways present the duty cycle of the nodes is negatively affected by the interference that exists between the heterogeneous MAC protocols. This issue however not only affects the virtual gateways used here. If a 'regular' gateway (with two radio interfaces) were used instead of a virtual one, that 'regular' gateway would suffer from the same issue. Given moreover that such a 'regular' gateway would need to power two radio interfaces and would therefore consume at the very least the sum of the energy used by each individual MAC protocol, the increase in duty cycle measured in these tests can still be considered to be quite modest. While the duty cycle wise cost of running a virtual gateway thus needs to be taken into account when deciding where (any how many) virtual gateways to deploy, it does not make the use of virtual gateways infeasible.

3.3 Network-wide performance evaluation

In order to get an indication of the effect that virtual gateways can have on the network-wide performance of the sensor networks connected through these gateways, a number of large-scale experiments were performed on a sensor network testbed. It should be noted that the effect of using virtual gateways largely depends on the application scenario, the deployment of the nodes and the resulting communication patterns of the networks. Deploying virtual gateways will, for instance, have little effect on the performance if they are not actually used to transmit packets between nodes of different networks. For the random-flows scenario introduced in section 2.2.2 this means that introducing virtual gateways only makes sense if they are needed to enable communication between 'client' and 'server' nodes located in different networks. Given that in that case the use of virtual gateways is *required* for the correct operation of these networks, it is not possible to directly compare the performance of the networks with and without these virtual gateways present. In contrast, for the node-to-sink scenario the use of virtual gateways may optimise, but is not required for, the correct operation of the networks since in this scenario each network is equipped with its own sink node. Given that the node-to-sink scenario thus allows for an easier comparison between the performance of the networks with and without the use of virtual gateways, the initial tests discussed in this section focus solely on this scenario.

For these initial experiments, two sensor networks are deployed in the same wireless

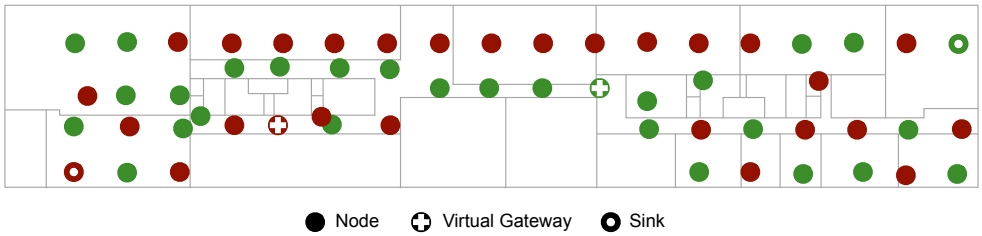


Figure 3.8: The node deployment for the sensor testbed experiments.

environment and two virtual gateway nodes are installed so packets can be routed over nodes of the other network. This allows the path length from the source nodes to the sink to be reduced. Two separate cases are considered. In the first case, packets can be routed over nodes of the other network but must still be sent to the original sink of the network itself. In the second case, nodes can also send their packets to the sink of the other network which allows the path length to be reduced even further. Both cases are compared to the case where no virtual gateways are available and data can only be routed over nodes of the network itself. In this case packets can only be sent to the original sink of the network.

All experiments were performed on the w-iLab.t wireless testbed [140] which contains several Tmote Sky sensor nodes deployed in an office building. For these initial experiments 54 sensor nodes deployed on a single floor were used. These nodes were separated into two networks of 27 nodes so each sensor network covers the entire floor. The sink nodes were placed at opposite ends of the building and the two virtual gateway nodes were chosen manually to be no more than two hops away from the sink nodes and to be maximally connected to nodes of the other network. This node deployment is shown in figure 3.8.

In these tests, the three different MAC protocols discussed in the previous section are considered: CSMA/CA, LPL-MAC and TDMA. Network performance is measured using two metrics. As with the tests discussed in section 3.2, the duty cycle of the nodes is used as a measure for the energy consumption. In addition, the end-to-end reliability (the percentage of generated packets which are correctly delivered to the sink) is also considered. Throughput is not measured because having all nodes transmit at the maximum data-rate would quickly overload the network and thus produce meaningless results. In addition, the stringent energy requirements usually found in large-scale sensor networks causes the network load to be relatively low and as a result throughput usually not that relevant. Although it would have been interesting to also consider the end-to-end delay in these tests, this was unfortunately not possible due to technical limitations.

To measure the end-to-end reliability, test runs of 1 hour are performed. During these test runs nodes send maximum-sized packets to the sink every 10 seconds. As discussed above, the sink used by each node depends on the scenario being tested. To ensure that the collected measurements are not affected by any variations in the route-topology of the networks, the routes used to relay these packets to the sink are pre-calculated based on the quality of the links (which was measured beforehand) and programmed into the nodes prior to the start of the test run. During the test run each node keeps track of

how many packets were sent and received. After the test run this information is then used to calculate the end-to-end reliability for both networks. During these reliability-tests, the duty cycle of the different nodes is also measured. After each test run, the duty cycle of the entire network is then calculated as the average duty cycle of all the nodes in the network. Since for these tests it was not possible to attach a logic analyser to each of the 54 sensor nodes, the CC2420 driver was modified to allow each node to keep track of its own duty cycle. Unfortunately, this means that the duty cycle can only be measured with the clock available on the sensor node itself and as a result the granularity of the measured radio on/off times is reduced from 16MHz to 32KHz. Despite this, the duty cycle measurements should still be sufficiently accurate since the granularity of the clock remains relatively large compared to the measured radio on/off intervals (which are typically between 6 and 100ms). Moreover, as with the tests discussed in section 3.2, the duty cycle was measured over a prolonged period of time (1 hour).

3.3.1 End-to-end Reliability

Figure 3.9 shows the delivery ratio of the two networks when using either the CSMA/CA and LPL-MAC protocol, the LPL-MAC and TDMA protocol or the CSMA/CA and TDMA protocol. For each test case, two test runs are performed to cancel out any asymmetries between the networks. For instance, in the CSMA/CA - LPL-MAC case, during the first test run one network uses the CSMA/CA protocol and the other one uses the LPL-MAC protocol while during the second test run the MAC protocols are swapped. The values shown are the averages over the two test runs.

For the CSMA/CA - LPL-MAC case, there is a significant difference in reliability between the MAC protocols used. When no virtual gateways are present, the reliability when using the CSMA/CA protocol is only around 62% whereas the reliability of the LPL-MAC protocol is almost 100%. This difference is caused by the 'wakeup' mechanism used by LPL-MAC. As discussed in section 2.1.3, LPL-MAC wakes up sleeping nodes by retransmitting the same data packet either until an acknowledgement is received or until a timeout occurs. This high amount of re-transmissions is not only responsible for the high reliability of the LPL-MAC protocol but also creates a substantial amount of interference for less 'greedy' MAC protocols, such as in this case the CSMA/CA protocol. When virtual gateways are added, this has a positive effect on the reliability of packets originating from CSMA/CA nodes whereas the opposite is true for packets originating from LPL-MAC nodes. For CSMA/CA the reliability is increased to 67% when data is routed to the original sink and 73% when data can also be sent to the sink of the LPL-MAC network. For LPL-MAC the reliability is reduced to 95% when routing data to the original sink and 84% when also routing data to the sink of the CSMA/CA network. This is caused by the fact that virtual gateways only enable communication between MAC-heterogeneous sensor networks. They do not eliminate the interference that exists between the different MAC protocols of these networks. As a result, the reliability of CSMA/CA is still impacted by the interference from the LPL-MAC protocol. The decreased reliability for the LPL-MAC network is therefore a direct result of the fact that packets originating from the LPL-MAC network are being sent over nodes of the less reliable CSMA/CA network. Likewise, the measured increase in reliability for the CSMA/CA network is caused by CSMA/CA originated packets being forwarded over nodes using the more reliable LPL-MAC protocol.

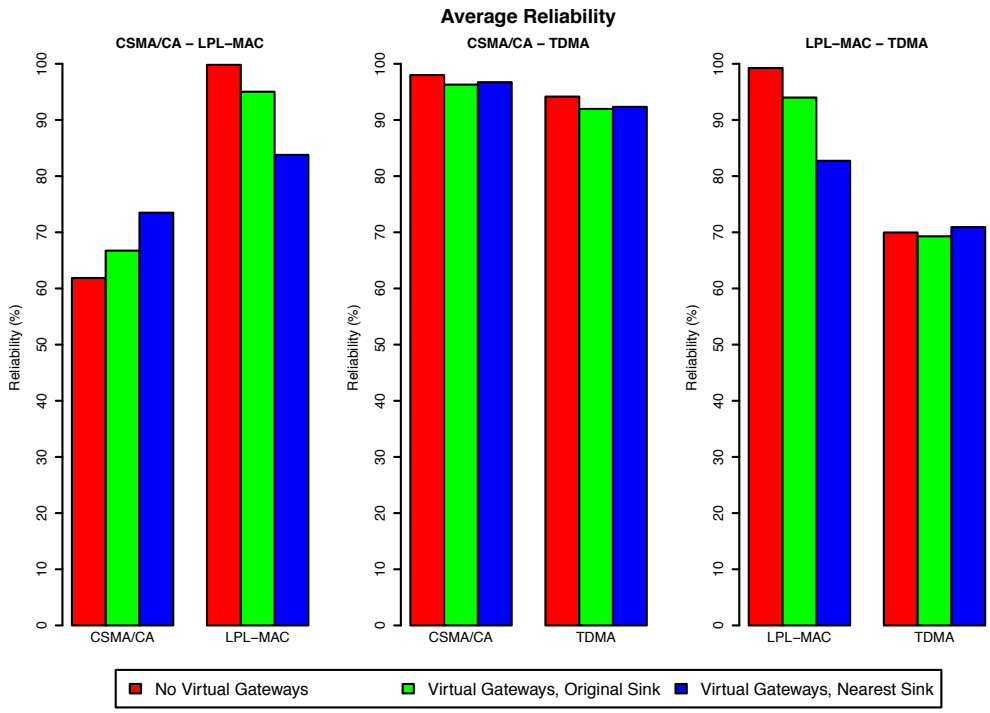


Figure 3.9: End-to-end reliability measured for two networks deployed in the same wireless environment. Each graph shows the reliability measured for a different set of MAC protocols used.

Similar observations can be made for the LPL-MAC - TDMA case. Without any virtual gateways in the wireless environment, the LPL-MAC protocol itself has a very high reliability (99%) while having a significant impact on the reliability of the TDMA MAC protocol, which only has a reliability of 70%. When virtual gateways are installed, packets originating from the LPL-MAC network suffer a reduction in reliability similar to the LPL-MAC - CSMA/CA case. When packets are routed to the original sink of the LPL-MAC network a reliability of 95% is achieved. When packets are also routed to the sink of the TDMA network the reliability drops to 84%. The TDMA protocol however does not react to the presence of virtual gateways in the same way that the CSMA/CA protocol does. When using virtual gateways, the reliability is only slightly different from the case where no virtual gateways are used. The reliability drops to 69% when routing packets to the original sink of the TDMA network and increases to 71% when the sink of the LPL-MAC network is also used. This different reaction of the TDMA protocol to the interference caused by the LPL-MAC protocol may be due to the specific node deployment used. As shown in figure 3.9, the sink nodes are placed at opposite ends of the office floor while the nodes of both networks span the entire floor. As a result, the TDMA nodes located nearest to the sink of the LPL-MAC have the most to gain from the deployment of virtual gateways since they are the furthest away from the TDMA sink. Given the fact that more packets are being transmitted in the immediate vicinity of the sink node than in the remainder of the network, these TDMA nodes are also the most affected by the

interference from the LPL-MAC network. Consequently, packets sent from these nodes to the sink are more likely to be lost at the beginning of the routing path, than further along the routing path. This means that most packets are lost before they reach the virtual gateway and as a result the addition of virtual gateways to the wireless environment only marginally affects the reliability of the TDMA network. Since all test cases use the same node deployment, this is also true for the LPL-MAC - CSMA/CA case. The main difference being that while CSMA/CA does have some mechanisms to cope with outside interference, such as clear channel assessments and acknowledgements, the TDMA MAC protocol used here has none. As a result, more packets reach the virtual gateway when CSMA/CA is used which results in a higher increase in reliability when enabling virtual gateways.

The measurements obtained for the CSMA/CA - TDMA case show that these protocols are much more reliable when used together than in combination with the LPL-MAC protocol. When no virtual gateways are present, the CSMA/CA network achieves a reliability of 98% while the TDMA network has a reliability of 94%. This is caused by the fact that these protocols are less greedy than the LPL-MAC protocol and thus cause less interference to one another. When virtual gateways are installed the reliability of both networks is lower than when no virtual gateways are used. This is most likely caused by the fact that for these experiments only two virtual gateways were installed. This means that all traffic that is passed between the two networks is routed over these two nodes and as a result there is more interference between the two MAC protocols in the neighbourhood of the virtual gateways. Despite this, the reliability only decreases to 96% for packets originating from the CSMA/CA network and 92% for packets originating from the TDMA network when packets are routed to the original sinks of the networks. When packets are sent to the nearest sink this loss of reliability is partially compensated by the fact that shorter routing paths are available. This results in a reliability of 96.5% for packets originating from the CSMA/CA network and 92.5% for packet originating from the TDMA network.

3.3.2 Duty Cycle

The duty cycles measured are shown in figure 3.10. As with figure 3.9, each value shown is the average over two individual test runs.

In the LPL-MAC - CSMA/CA case, the duty cycle of the CSMA/CA network is 100% regardless of whether virtual gateways are used or not. This is because this MAC protocol never disables the radio. When the duty cycles measured for the LPL-MAC network are considered, it is clear that the addition of virtual gateways has a negative effect on the energy efficiency. The average duty cycle in the LPL-MAC network rises from 34% when no virtual gateways are present to 37,5% when virtual gateways are used in combination with the original sinks of the networks. As discussed above, the limited number of virtual gateways used for these experiments creates a number of ‘hot spots’ in which there is an increased amount of traffic and interference between the MAC protocols. Since the LPL-MAC protocol is sensitive to the amount of activity on the channel when it comes to energy consumption, these hot spots have a negative effect on the overall duty cycle of the LPL-MAC network. When packets are routed to the nearest sink instead of the original sink however the duty cycle of the LPL-MAC network drops to 35%. In this case the

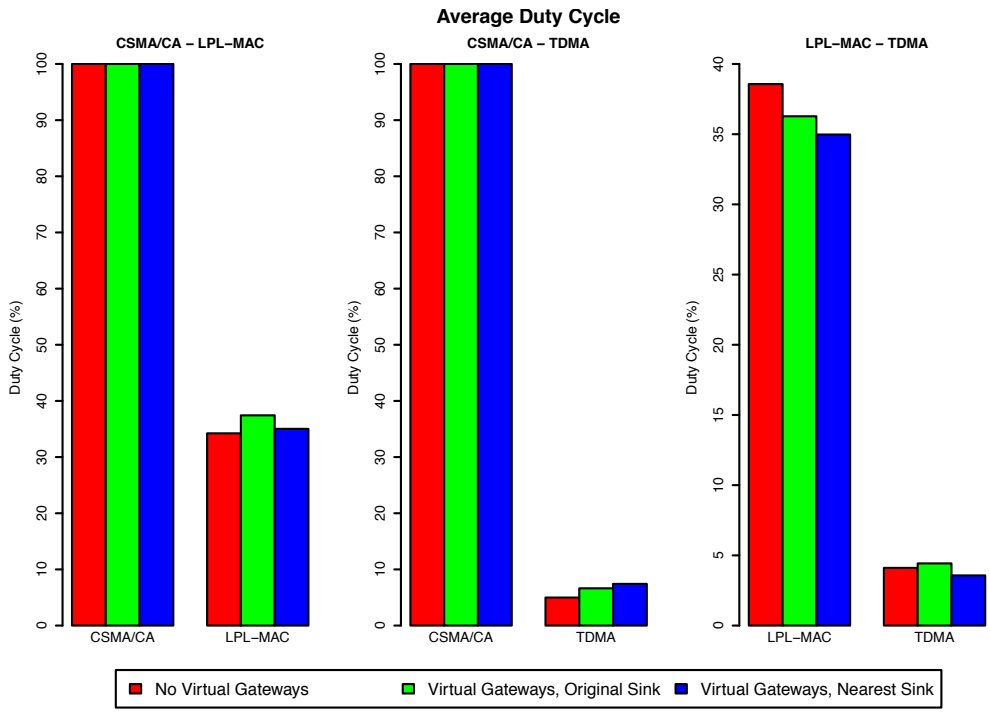


Figure 3.10: Duty cycle measured for two networks deployed in the same wireless environment. Each graph shows the duty cycles measured for a different set of MAC protocols used.

shorter routing paths used partially compensate for the increase in energy consumption caused by routing over only two virtual gateways.

When the CSMA/CA and TDMA protocols are used, the duty cycle of the CSMA/CA network is, again, 100% in all cases while the duty cycle for the TDMA MAC protocol increases from 5% when no virtual gateways are present to 6.5% when virtual gateways are used in combination with the original sinks and 7.5% when packets are also routed to the other sink. The increase in energy consumption for the TDMA network is a direct result of the fact that additional slots are allocated to allow the additional traffic from the CSMA/CA network to be routed over nodes in the TDMA network.

Finally, when the LPL-MAC protocol is used together with the TDMA MAC protocol it is clear that, as shown in figure 3.10, virtual gateways seems to have a positive effect on the energy consumption of both networks. The LPL-MAC network has a duty cycle of 38.5% in the case that no virtual gateways are present. When virtual gateways are installed the duty cycle drops to 36% in the case that packets are routed to the original sink and 35% if packets are also routed to the other sink. For the TDMA network the duty cycle increases from 4% to 4.5% when packets are routed over the virtual gateways to the original sink, but when packets are routed to the nearest sink the duty cycle drops down to 3.5%. This reduction in energy consumption however comes at a cost. As discussed in section 3.3.1, for this scenario the reliability of the networks is also reduced when virtual gateways are installed. Since a lower reliability results in less traffic, the duty cycle is also

reduced.

3.3.3 Result Analysis

The main result of the performed experiments is that although virtual gateways enable the communication between MAC heterogeneous sensor networks, this does not automatically lead to a more optimal operation of the networks involved. For the specific application scenario and node deployment investigated here, virtual gateways do allow for shorter paths to be used (a reduction in path length of respectively 4% and 24% was observed depending on whether packets were sent to the original sink or to the nearest sink) but this reduced path length comes at a reliability- and duty cycle-wise cost. Depending on the specific combination of MAC protocols used, the end-to-end reliability is reduced by up to 17%-points and for most test scenarios the use of virtual gateways also increased the energy usage. This is mostly due to the fact that the application scenario and node deployment chosen for these experiments did not take into account the effects of the interference that exists between nodes using different MAC protocols. Given, for instance, the effect that the LPL-MAC protocol has on the reliability of the CSMA/CA and TDMA protocols, choosing routes which avoid inter-network interference as much as possible could have resulted in a better performance. Instead a node deployment was used in which the nodes of the networks are evenly distributed over the same office floor, which guarantees that each node suffers from the effects of interference regardless of which routes are chosen. Moreover, only a limited number of gateway nodes were used since, as discussed in section 3.2.2, using a node as a virtual gateway does incur a certain performance overhead. In hindsight, it would have been better to deploy more virtual gateways since limiting the number of virtual gateways created 'hot spots' in the wireless environment in which interference between the MAC protocols is increased even further. The lesson to be learned from these experiments therefore is that virtual gateways do not remove the interference that exists between different MAC protocols and that, although significant optimisations may be achieved depending which performance metrics are of importance to the respective networks, great care must be taken in determining both the location and the amount of virtual gateways used.

3.4 Conclusion

This chapter investigated the feasibility of using *Virtual Gateways* as a means to enable communication between MAC heterogeneous sensor networks. To this end a network stack capable of running multiple MAC protocols (MultiMAC) was developed in TinyOS for the Tmote Sky platform. As stated in section 3.1, this network stack needed to combine a *flexible* and *extensible* architecture with a low performance *overhead* in order for virtual gateways to be considered a viable solution for enabling inter-MAC connectivity.

From the discussion in section 3.1.3 it is clear that this stack is *flexible* and *extensible* enough to support a wide variety of MAC protocols. Moreover, the performance tests with a single MAC protocol discussed in section 3.2.1 make it plain that the overhead of supporting multiple MAC protocols is almost negligible. The tests performed with the 'Passthrough' stack show that even when all other components (except the HAL) in the network stack are the same, the performance overhead of supporting multiple MAC

protocols is so small as to be nearly negligible (around 3% at the very most). Despite this 'multiplexing overhead' however, the MultiMAC network stack manages to outperform the TinyOS network stack by more than a factor of 2 in terms of throughput and round-trip time and by more than 40%-points in terms of duty cycle. This shows that the performance overhead of supporting multiple MAC protocols is insignificant compared to the performance benefits that can be achieved by being performance-conscious both in the design and implementation of the network stack.

Section 3.2.2 investigated the performance of the MultiMAC stack when two MAC protocols are used simultaneously. These performance tests show that while, as far as delay is concerned, virtual gateways perform in most cases no worse than the least performant of the individual MAC protocols used, they also show that there is a significant duty-cycle wise cost associated with configuring a sensor node as a virtual gateway. While, as discussed in section 3.2.2, this is partially due to the interference that exists between the MAC protocols, this does mean that this duty cycle wise cost needs to be taken into account when choosing the (number of) virtual gateway to use.

This need to be careful in the selection of the virtual gateway nodes is also made clear by the large-scale experiments performed on the w-iLab.t testbed. These tests not only show that the performance of the networks can suffer if the selected virtual gateways do not fit the communication patterns in and between the networks, but also illustrate that there are a significant number of factors to consider in the selection of the virtual gateway nodes. Firstly, the decision as to which nodes to configure as a virtual gateway requires a trade-off to be made between different and possibly conflicting performance metrics. (As discussed in section 3.3.3, using too few virtual gateways had a negative effect on the reliability; using more virtual gateways however would have increased the duty cycle of the networks). For the most optimal results, the extent to which the different performance metrics are of importance to the respective networks should therefore be considered in the selection of the virtual gateway nodes. In addition, the manner in which these performance metrics are affected by the introduction of virtual gateways also varies with the specific combination of MAC protocols used. The use of virtual gateways for instance reduced the duty cycle for the 'LPL-MAC - TDMA' case but not for the cases where one of the networks used the CSMA/CA MAC protocol. To complicate matters even more, it is moreover not guaranteed that all the networks involved react in the same manner to the introduction of virtual gateways and, as illustrated by the end-to-end reliability measurements collected for the 'CSMA/CA - LPL-MAC' case, it is for instance perfectly possible for virtual gateways to let one network benefit at the expense of the other. Given all the different factors to consider, it is clear that although virtual gateways are a viable method of enabling connectivity between MAC-heterogeneous networks, the problem of selecting the virtual gateways to use based on the requirements of the individual networks is a complex issue that warrants further investigation. For this reason, the virtual gateway selection problem is further investigated in the remaining chapters of this thesis.

Before moving on to the next chapter it should be noted that the remaining chapters in this thesis do not use the w-iLab.t testbed as a test platform. The main reason for this is that the virtual gateway selection problem is simply too complex to allow it to be researched using real-life sensor nodes alone. The tests discussed in section 3.3 already took 18 hours to run, spread over 3 nights to avoid interference from WiFi-transmitters. Given that for a proper investigation into the virtual gateway selection problem a much

wider range of parameters and a much larger set of possible virtual gateway combinations need to be considered, doing so using only the w-iLab.t testbed as a test platform is simply not feasible. In addition, it should also be noted that since performing the tests discussed in section 3.3, the w-iLab.t testbed has been torn down permanently, which means that even using this testbed as part of a larger test platform is not an option.

For these reasons the work discussed in chapters 4, 5 and 6 is performed using the Castalia simulator introduced in section 2.2. To support the use of virtual gateways, the Multi-MAC architecture has been incorporated into the Castalia network stack but since the mechanisms used to support the use of multiple MAC protocols are the same as discussed in section 3.1.2, there is no need to discuss the Castalia implementation any further. It is worth mentioning however that, although the tendency of wireless network simulators to overestimate the processing capabilities of the nodes meant that the feasibility of the virtual gateway approach had to be investigated using real sensor nodes, this limitation does not pose a problem for the work presented in the remainder of this thesis. After all, the feasibility study presented in this chapter has shown that even extremely resource constrained sensor nodes (such as Tmote Skys) can be used as a virtual gateway and that the computational requirements of the virtual gateway approach are therefore not an issue. As a result, the fact that the Castalia simulator is somewhat optimistic about the processing capabilities of the nodes is of no consequence for the investigation of the virtual gateway placement problem.

CHAPTER 4

Virtual Gateway Selection Part 1: Prediction Algorithm

The previous chapter introduced the concept of a Virtual Gateway: i.e., a regular sensor node running multiple MAC protocols simultaneously. Although these virtual gateways offer a viable means to enable communication between MAC-heterogeneous sensor networks, the initial large-scale tests discussed in section 3.3 make it plain that great care should be taken in choosing the exact set of sensor nodes to be used as a virtual gateway. Selecting the optimal set of nodes to use as a virtual gateway however is not a simple task since it requires finding a balance between the different and possibly conflicting requirements of each individual network. (One network may for instance prioritise duty cycle over hop count while another network may do the opposite.) Moreover, the optimal location of the virtual gateways depends heavily on the routing paths between the networks and these routing paths are in turn influenced by the location of the virtual gateways.

This thesis therefore introduces IRVG (Iterative Removal of Virtual Gateways). IRVG is a heuristic mechanism for selecting the optimal set of virtual gateway nodes to enable interoperability between two MAC heterogeneous sensor networks. Unlike existing gateway placement techniques (see section 4.1.1) this mechanism does not rely on a predefined model to decide on the selection of virtual gateway nodes. Instead, virtual gateways are selected by iteratively applying multiple virtual gateway configurations to the sensor networks and thus evaluating the effect of each configuration on their overall performance. As the name implies, IRVG starts from a large set of virtual gateway nodes and then removes one or more virtual gateways per iteration.

The approach used by IRVG requires two main research problems to be addressed. First, a prediction algorithm needs to be developed that, based solely on past measurements, can accurately predict the network performance resulting from the removal of a specific

set of virtual gateways. Next, a selection algorithm is needed to select the next set of virtual gateways to be removed based on the results of the prediction algorithm.

Since these two algorithms are two separate aspects of the proposed IRVG mechanism, these algorithms are discussed in two separate chapters. This chapter first discusses the design considerations and the approach used by IRVG. Afterwards the prediction algorithm itself is discussed in section 4.2 and its accuracy is investigated in section 4.3. The selection algorithm on the other hand is discussed in chapter 5. It should also be noted that, although IRVG has been designed to be usable for a wide range of application scenario's, the performance analysis in sections 4.3 and 5.2, focus on the random-flows scenario introduced in section 2.2.2. Chapter 6 further investigates IRVG for the node-to-sink use case.

4.1 Iterative Removal of Virtual Gateways

4.1.1 Related Work

Gateway placement algorithms have been studied extensively in literature both for sensor networks and for wireless mesh networks. Despite the fact that these types of networks have very different properties, the underlying problem addressed by the different gateway placement algorithms is remarkably similar. In both cases, gateway placement effectively involves finding entry and exit points in the network that allow for the most optimal routing paths to (and from) these points. The two most noticeable differences between the algorithms for sensor networks and those for mesh networks are the scale (number of nodes) of the network and of course the metrics used to evaluate a particular gateway placement.

For sensor networks 'network lifetime' and 'hop count' are two of the most common metrics used, although 'hop count' is sometimes also expressed as latency to the nearest gateway. [141] for instance proposes a number of Integer Linear Programming (ILP) models to minimise the number of required gateways given specific delay or hop count constraints. Inversely they also allow the worst case delay or hop count to be minimised given a maximum number of gateway nodes. [142] uses ILP in combination with a flow-based routing protocol to iteratively reposition mobile gateways to maximise the network lifetime. [143] combines ILP with an approximation approach to optimise the network lifetime for the 'single gateway' case. ILP is used to estimate the network lifetime for a number of possible gateway locations. The list of considered gateway locations is generated by discretising the search space. The level of discretisation is chosen based on the required approximation bounds, which allows network designers to make a tradeoff between the quality of the approximation and the number of gateway locations to consider. ILP is of course not the only technique used to select the optimal gateway placement. [144] for instance use a greedy heuristic to minimise the number required gateway nodes given a maximum number of hops from the sensor nodes to the nearest gateway while [145] uses genetic algorithms to minimise the average latency in the network. It should be noted that for sensor networks, gateway placement is very similar to the problem of determining where the 'sink' node(s) of the network should be placed and as a result the terms 'gateway', 'sink' and 'basestation' are used interchangeably in literature.

Gateway placement algorithms for mesh networks use different criteria to determine the optimal gateway placement. [146] and [147] for instance both attempt to minimise the number of gateways given specific constraints on the capacity of the gateways and the maximum hop count. In both cases, interference between nodes and links is expressed as a constraint on the load of individual links. [148] directly considers interference between nodes in the selection of the gateway nodes. The selection algorithm itself relies on 'brute-force' tactics to find the optimal selection and therefore has limited scalability. [149] proposes a heuristic gateway selection algorithm that only considers the overall interference between the different gateway nodes, rather than interference between individual nodes. As with the work of [148], interference between individual nodes is resolved by using TDMA and providing an interference-free slot allocation.

Finally, [150] investigates the case where a wireless mesh network is used as a backbone network for a sensor network. In that case, the location of the gateway nodes is not only subject to optimisation concerns in the sensor network but also to design constraints of the backbone network. The authors therefore propose an ILP model and a number of heuristic algorithms to jointly optimise the energy efficiency of the sensor network and minimise the installation cost of the backbone network.

Although there are quite a number of algorithms for gateway placement currently available, none of these can be used to determine the optimal location of virtual gateways in the network. The main reason for this is that there are a number of fundamental differences between the 'traditional' gateways considered by previous work and the *virtual* gateways considered here. Firstly, as discussed above, existing algorithms consider gateway nodes to be exactly the same as sink nodes, which means that these nodes are assumed to be mains powered and relatively powerful. Because of this, existing algorithms do not take into account the energy and resource constraints of the gateway nodes. Virtual gateways however are very nearly identical to regular sensor nodes. They have the same hardware capabilities as regular nodes and are subject to the same resource and energy constraints. The only difference is that virtual gateways, in addition to the tasks of regular sensor nodes, also have to run multiple MAC protocols simultaneously. A second problem lies in the fact that 'traditional' gateway placement algorithms only need to consider the traffic flows from the sensor nodes to the gateways. The reason for this is that once traffic from the sensor network reaches the gateway node, it is forwarded to its final destination over a backbone network. This means that once traffic reaches the gateway node it no longer influences other transmissions in the network and can therefore be said to be 'removed' from the wireless environment of the sensor network. This is however not the case when virtual gateways are used, since they do not 'remove' packets routed over them from the wireless environment. Instead, the MAC protocol used to transmit the packet is altered after which it is reinserted into the wireless environment. As a result, the gateway placement algorithm also needs to consider the traffic leaving the gateway nodes.

Moreover, the effects of inter-node interference also need to be taken into account when deciding on the set of virtual gateway nodes to use. This is because virtual gateways are only used when multiple MAC protocols are active in the same wireless environment at the same time. In that case, these MAC protocols will interfere with each other since they use incompatible channel access mechanisms. As discussed in section 2.5, most MAC protocols are able to cope relatively well with a small amount of interference from other MAC protocols. When the load in the network increases however, the resulting

interference can have a very significant effect on both the reliability and the energy usage in both networks.

The current gateway placement algorithms however, operate on models of the sensor network that drastically simplify the characteristics of the wireless channel and the behaviour of the MAC protocol. The energy cost of sending and receiving packets is often assumed to be constant [141, 144] or is assumed to be only dependent on the distance between sender and receiver [143, 142]. Likewise, link reliability is also greatly simplified. [141] and [142] for instance assume that nodes have a fixed transmission range while [143] and [144] operate on a connectivity graph and implicitly assume all links in the network to be lossless. Moreover, interference between nodes is not considered at all by gateway placement algorithms for sensor networks while for mesh networks, interference is mostly modelled as constraints on the load of individual nodes [146, 147] or resolved by calculating an interference-free transmission schedule for the entire network. To the best of the author's knowledge there are currently no gateway placement algorithms that directly consider interference between nodes when deciding on the gateways to use.

4.1.2 Design considerations

The Iterative Removal of Virtual Gateways (IRVG) mechanism is intended to provide a general solution to the virtual gateway placement problem. This means that this mechanism should be usable for a wide range of application scenarios and not just for a single use case. To this end IRVG should adhere to the following requirements:

Independent from the MAC protocols: To be generally usable, IRVG must be independent from the used MAC protocols. This means that no a-priori assumptions can be made about the nature of the MAC protocols used (such as for instance the fairness-property of CSMA/CA).

Minimal restrictions on the routing protocol: Ideally IRVG should be completely independent from the routing protocol and routing metrics used in the sensor networks. Unfortunately, this is not possible due to the fact that the optimal set of virtual gateway nodes depends on the routing paths used and that these routing paths are in turn influenced by the location of the virtual gateway nodes. IRVG therefore assumes that:

1. the routing protocols and metrics used are 'sufficiently compatible' with each other. This means that once one or more virtual gateways are enabled and link-level connectivity is established between the networks, the routing protocols should be able to exchange routing information and establish routing paths over and between nodes of different networks.

2. the used routing protocols dynamically choose the routes that are most optimal for the applications running in the network. This in essence means that the routing metrics used should reflect the application requirements of the networks.

It should be noted that (1) is not a restriction imposed by IRVG. It is in fact a vital prerequisite for multi-hop communication between MAC-heterogeneous sensor networks which is merely being exploited by IRVG. Moreover, IRVG does not impose any restriction on the specific type of routing metrics used or the protocols used to disseminate routing information in the networks.

Tuneable to the Application Requirements: Given the fact that sensor networks are being used for an increasingly wide array of sensor network applications, it is imperative that the gateway selection mechanism is able to take the specific application requirements of the networks into account. As will be further discussed in chapter 5, this is accomplished by the use of a reward function that allows the administrators of the individual networks to specify 'goals' and 'weights' for the performance metrics gathered for each network. Within the scope of this work three performance metrics are considered: duty cycle, reliability and hop count.

It should be noted that this thesis only considers the case where there are only two MAC-heterogeneous sensor networks in the wireless environment. It is moreover assumed that all nodes have a network specific 'default' MAC protocol that is used for intra-network communication and that virtual gateway functionality is therefore only used for inter-network links. Extending the IRVG mechanism to the n-network case is beyond the scope of this work but should be relatively straightforward.

4.1.3 Approach

The approach used by IRVG to determine the set of virtual gateway nodes to use, differs greatly from the one used by many existing gateway placement algorithms. In order to allow them to be used as network planning tools (i.e., to help decide where each node will be installed during deployment) these existing algorithms operate solely on a predefined model of the sensor network. In the case of virtual gateways however such a model would have to be able to accurately model the effects of interference between MAC heterogeneous sensor nodes. Given that this depends on how exactly the MAC protocols interact with the wireless environment, such a model would require intimate knowledge of, and would therefore be dependent on, the MAC protocols used by the sensor networks. Since IRVG must be independent from the MAC protocols used, using such a model is not feasible.

Unlike these existing algorithms however, IRVG is not designed to be a network planning tool. Instead, it is a tool for enabling communication between MAC-heterogeneous sensor networks *that have already been deployed*. Because of this, the performance of a virtual gateway configuration can be evaluated by applying it to the sensor networks and subsequently measuring the network performance. In contrast to existing gateway placement algorithms, IRVG therefore operates directly on the sensor networks to determine the virtual gateway configuration to use. In this work the centralised architecture shown in figure 4.1 is used to allow IRVG to interact with the sensor networks. In this architecture, IRVG is executed on a central controller which is connected to the sensor networks through the local endpoints. The local endpoints are responsible for applying the virtual gateway configurations in their respective networks and collecting performance measurements and routing information.

To determine the final set of virtual gateway nodes, IRVG iteratively applies multiple virtual gateway configurations to the MAC-heterogeneous sensor networks. After each configuration has been applied, the following information is gathered from the networks to allow IRVG to evaluate the applied configuration and to allow the next configuration to be calculated.

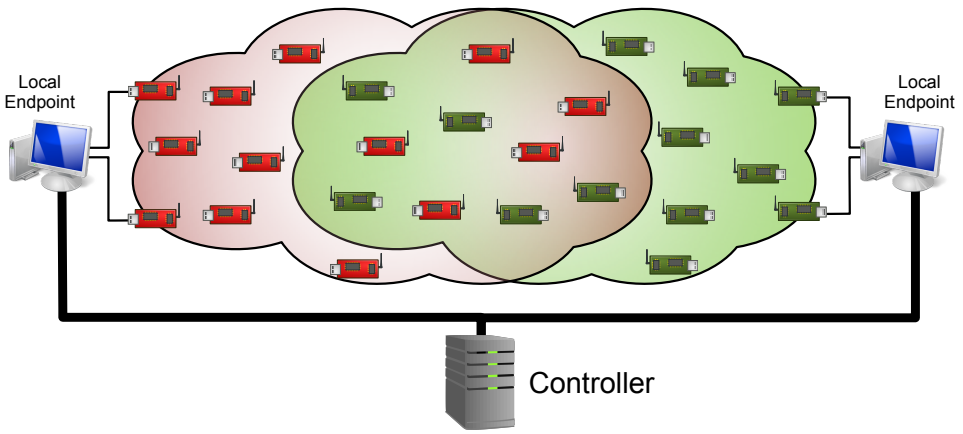


Figure 4.1: Supporting Architecture for IRVG.

Node Duty Cycle: The duty cycle is measured for each individual node in the network and is used to estimate the energy-cost of configuring a node as a virtual gateway.

Neighbour Tables: The neighbour tables of the individual nodes are collected. This not only includes the actual list of neighbours but also the link reliability to each of those neighbours. Although this does require nodes to build and maintain a neighbour table, it should be noted that "*...neighbourhood management* [is] *essential to and tightly coupled with reliable routing in sensor networks*" [128] and that most routing protocols will therefore already maintain a neighbour table. This information is only used to estimate the effect of removing a gateway on the overall reliability of the networks. IRVG is still able to function if this information is not available but in that case reliability will not be taken into account.

End-to-end flow information: To estimate the effect of removing a gateway on the routing paths in the networks, IRVG first needs to know which routing paths are being used and how much traffic flows over each of them. Since no assumptions are made about the type of applications running in the network it is not known in advance which nodes will communicate with each other. As discussed above, no assumptions are made about the specific routing mechanisms and metrics used and as a direct result the traffic patterns in the networks are collected from the networks during each iteration.

When measuring the traffic flowing over the routing paths, this traffic should moreover be tracked for each network individually. The reason for this is that, despite being 'joined' by one or more virtual gateways, the individual sensor networks may have very different requirements for the duty cycle, reliability and hop count experienced by the nodes in the respective networks. To allow IRVG to evaluate and optimise the performance of traffic flows of different networks using different metrics (eg: reliability for one network and hop count for the other), these traffic flows should be tracked for each network individually. Moreover, feedback from the application layer is needed to properly distinguish between traffic benefitting the application(s) of the individual networks.

Why this is the case is best explained by an example. Consider for instance the random-

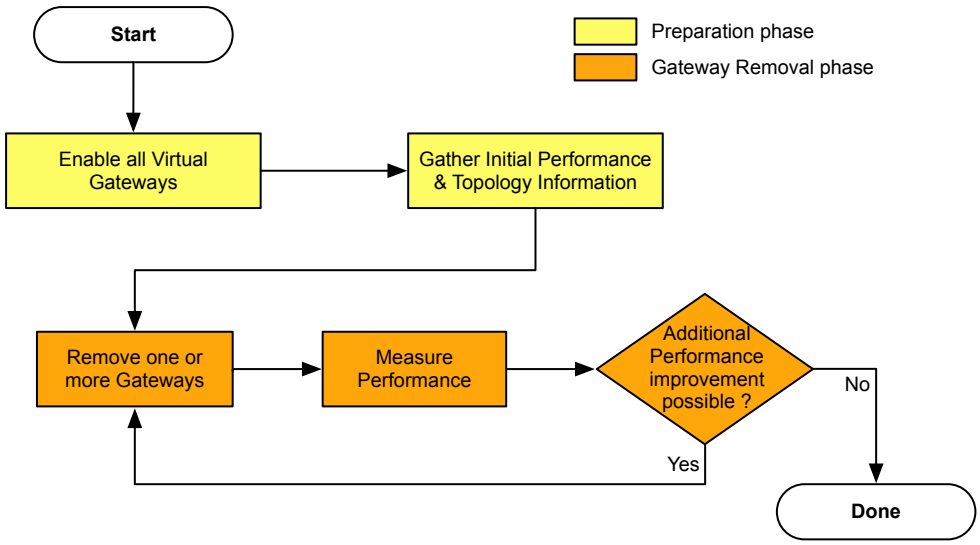


Figure 4.2: High-level summary of the IRVG selection mechanism.

flows scenario introduced in section 2.2.2 in which nodes regularly send 'query'-messages to other nodes to request sensor readings. Nodes receiving such 'query'-messages respond with a 'response'-message containing the sensor reading. In this case both the 'query'- and the 'response'-message are only important to the node that sent the original query since it is the one interested in the sensor readings. The 'responding' node on the other hand does not benefit from the traffic it receives from the 'querying' node. In this case, both the 'query'- and the 'response'-message should therefore be counted towards the network of the 'querying'-node. As is clear from this example, the network to which a traffic flow belongs cannot be determined unambiguously from the source and the destination node alone and therefore feedback from the application layer is needed to decide which traffic to account to which network.

As shown in figure 4.2, IRVG consist of a *preparation* and a *gateway removal* phase. During the preparation phase, all virtual gateway nodes are enabled in both networks after which initial topology information and performance measurements are gathered from these networks. During the *removal* phase, IRVG iteratively removes unnecessary gateways from the set of virtual gateway nodes until no further improvement is possible. The advantage of this heuristic 'removal only' strategy is that it minimises the restrictions imposed on the routing protocols. By using a 'removal only' strategy IRVG is able to 'learn' the routing paths used in the networks and only has to account for routing paths that might be broken by the removal of a virtual gateway when calculating the next virtual gateway configuration to apply. Conversely, if virtual gateways were added to the virtual gateway configuration, IRVG would also have to predict which new routes are created by the addition of a virtual gateway. This would require much more intimate knowledge of, and therefore impose much wider restrictions on, the routing protocols and routing metrics used.

Two algorithms are used to determine during each iteration which virtual gateway config-

uration to apply next: a prediction and a selection algorithm. The prediction algorithm predicts the effect of removing a set of virtual gateways on both the topology and the performance of the network. The selection algorithm then decides which gateways to remove next based on these predictions. Both algorithms operate solely on routing information and routing measurements gathered from the networks.

The selection algorithm is further discussed in chapter 5, the prediction algorithm is discussed in the remainder of this chapter.

4.2 Prediction Algorithm

The prediction algorithm of IRVG estimates the effect of removing a set of virtual gateways on the duty cycle of the nodes and on the reliability and hop count in the networks. To do so the prediction algorithm first predicts the effect of removing said gateways on the *topology* of the networks and subsequently calculates the reliability, hop count and duty cycle resulting from this topology. In order to predict how the removal of a virtual gateway affects the topology of the network however, the measurements collected after each iteration of the selection algorithm do not suffice. Two additional virtual gateway configurations need to be evaluated before any predictions can be made. IRVG therefore first applies a so-called *maximum* and *minimum* gateway configuration to the networks before the selection algorithm takes over.

The *maximum* gateway configuration consists of all nodes in both networks capable of being configured as a virtual gateway. Applying this configuration to the networks should allow the preferred routes of the routing protocols to be discovered since this configuration allows all conceivable inter-network links to be used. The *minimum* gateway configuration consists of those specific virtual gateways that are deemed to be essential for communication between the two networks (i.e., these virtual gateways are never disabled). This configuration is applied in the networks for two reasons. Firstly, it allows the 'fallback routes', that is the routes used when no other gateways are available, to be discovered. Secondly, by comparing the duty cycles of the nodes for the maximum and the minimum gateway configuration, the energy cost of configuring a particular node as a virtual gateway can be estimated. Since the virtual gateways of the minimum configuration are never disabled by the selection algorithm the energy cost of configuring these nodes as virtual gateways therefore doesn't need to be known. Although this does limit the possible set of virtual gateway configurations, it should be noted that, depending on the use case, the minimum gateway configuration is allowed to be empty. This will for instance be the case for the node-to-sink scenario introduced in section 2.2.2. For the random-flows scenario considered here however, the minimum gateway configuration always consists of a single node: the most used virtual gateway from the *maximum* gateway configuration.

In what follows, the topology of the network resulting from the removal of the selected gateway is called the *predicted* topology. This topology is calculated from the *current* topology (from which the gateways are to be removed) and from the *minimum* and *maximum* topologies for which the performance was measured at the very beginning.

4.2.1 Mathematical Notations

V is the set of nodes in both networks. Since the prediction algorithm does not distinguish between nodes of different networks the union of the individual networks is referred to as 'the network' in this section.

M is the set of all MAC protocols and $M_d(i)$ denotes the default MAC protocol of node $i \in V$.

GW_x is the set of virtual gateway nodes for network topology x, where x is one of min, max, c or p and refers to either the *minimum*, *maximum*, *current* or *predicted* topology.

$LINK_x = \left(V, E_x^L\right)$ is the link graph of topology x and is constructed from the neighbour tables collected from the nodes in the network. Since multiple links can exist between two nodes (one for each available MAC protocol), $LINK_x$ is a directed multigraph and all edges are labelled with the MAC protocol used for the corresponding link. More formally:

$$E_x^L = \{\{v_i \to v_j, m\} | v_i, v_j \in V, m \in M, \\ v_i \text{ is in the neighbour table of } v_j \text{ for MAC protocol } m\}$$

For simplicity's sake $\{v_i \to v_j, m\}$ is written as $v_i \overset{m}{\to} v_j$.

$PATHS_x$ is the set of routing paths used in the network for topology x and is constructed from the end-to-end path information collected from the nodes. Each path consists of a set of consecutive links connecting the source node to the target node. Each link in a path is associated with a single MAC protocol and must be an edge in the link graph. More formally:

$$PATHS_x = \{p | p = \{v_1 \overset{m_1}{\to} v_2, v_2 \overset{m_2}{\to} v_3, \ldots, v_n \overset{m_n}{\to} v_{n+1}\}, p \subset E_x^L\}$$

The source and destination nodes of a path p are denoted as $SRC(p)$ and $DST(p)$.

The paths in $PATHS_x$ are logically divided into multiple flows. Within the scope of this work a flow is defined as the set of all routing paths sharing a common source and a common destination node:

$$FLOW_x(i,j) = \{p \in PATHS_x | SRC(p) = i, DST(p) = j\}$$

with $FLOW_x(i,j)$ denoting the flow from node i to node j. $FLOWS_x$ is the set of all flows for topology x.

The route graph for topology x is denoted as

$$ROUTE_x = \left(V, E_x^R\right)$$

and is constructed from the paths in $PATHS_x$:

$$E_x^R = \{v_i \overset{m}{\to} v_j | \exists p \in PATHS_x : v_i \overset{m}{\to} v_j \in p\}$$

$TX_x(p,i)$ is the number of packets transmitted over path $p \in PATHS_x$ for network i during the iteration where topology x was active. For simplicity's sake it is assumed that

gateway configurations are active for the same amount of time and that all iterations therefore have the same length.

$DC_x(i)$ is the duty cycle of node i for a network topology x.

$R_x^R(e)$ and $R_x^L(e)$ are two separate measures for the reliability of link e. $R_x^L(e)$ is the reliability of link e obtained through the neighbour discovery mechanisms used in the network and is hereafter referred to as the '*link reliability*'. In essence, the link reliability refers to the reliability-measurements that are stored in the *neighbour tables* of the different nodes and which are used by the routing protocol to decide the route topology of the network. R_x^R is the reliability of the link measured as the fraction of data packets that are successfully forwarded over this link and is referred to as the '*route reliability*'. These two reliability measures are not necessarily equal since they are measured using entirely different mechanisms. In general, $R_x^R(e)$ is the most accurate measurement of the two since it takes into account any reliability enhancing features of the MAC protocol, such as acknowledgements and retransmissions, but it is only available for links that are actually used in routing paths (i.e., links in E_x^R). R_x^L is available for all links in E_x^L, but its accuracy depends largely on the specific reliability estimation and neighbourhood management techniques used by the nodes and is therefore used as a 'rough' estimation of the reliability.

It should be noted that although $R_x^R(e)$ and $R_x^L(e)$ are two separate reliability metrics for a single link e, this distinction is not made by the prediction algorithm. Instead, the prediction algorithm considers each link e to have a 'reliability' property that is dependent on the graph in which the link resides. R_x^R is the reliability of the links in the route graph $ROUTE_x$ while R_x^L is the reliability of the links in the link graph $LINK_x$.

4.2.2 Topology prediction

Predicting the topology resulting from the removal of one or more gateways, corresponds to constructing $DC_p(i)$, E_p^L, E_p^R, $PATHS_p$, TX_p, R_p^L and R_p^R from the current topology and the set of remaining virtual gateways GW_p.

To calculate $DC_p(i)$, it is assumed that the duty cycle-wise cost of running as a virtual gateway does not vary from one iteration to another. This allows the duty cycle to be predicted based on the duty cycle measurements collected for the minimum and maximum topology. $DC_p(i)$ is thus calculated as follows:

$$limit(x) = min\,(1, max\,(0, x))$$

$$DC_p(i) = \begin{cases} limit\,(DC_c(i) - (DC_{max}(i) - DC_{min}(i))) & \text{if } i \in GW_c \setminus GW_p \\ DC_c(i) & \text{otherwise} \end{cases}$$

For every node i that is a gateway in the current but not the predicted topology (i.e., the gateway is being removed), the difference in duty cycle between the maximum and minimum topology is used to estimate the *cost* of running as a virtual gateway for that specific node. The new duty cycle of the node is then predicted by subtracting this cost from the current duty cycle of the node ($DC_c(i)$). This result is then limited to the interval $(0, 1)$ to prevent $DC_p(i)$ from becoming invalid as a result of inaccuracies in the estimation of the duty cycle-wise cost.

To calculate E_p^L, E_p^R, $PATHS_p$, TX_p, R_p^L and R_p^R, the prediction algorithm assumes that when a virtual gateway is disabled, only those paths from the current topology which pass through the gateway are affected. All other paths are assumed to be unaffected. Paths are processed iteratively to build the predicted topology. Paths that are unaffected by the removal of the gateways are added to the predicted topology without any alterations. For each of the paths broken by the removal of the gateways, the prediction algorithm first tries to repair the path by replacing the broken links with equivalent links using another MAC protocol. If this is not possible, the path is removed from the flow to which it belongs. Flows that no longer contain any paths after all paths have been processed are deemed to be 'broken'. For each of these, the prediction algorithm tries to find a set of alternative paths. This is done by iteratively applying multiple 'replacement policies'. If no replacement paths can be found, the flow remains broken and all traffic sent over it is assumed to be lost. The total amount of packets sent over broken flows is stored in the $TX_{broken}(i)$ variable for each individual network i and is later used to predict the average reliability in the network.

4.2.2.1 Formal Definition of the Algorithm

Algorithm 1 details the operation of the topology prediction algorithm. To keep the description of the algorithm as minimal and as clear as possible, not all details of the prediction algorithm are considered. The traffic flowing over a path is not considered separately for each network since this distinction is not made by the prediction algorithm and all traffic is considered equally. Because of this $TX_{broken}(i)$ and $TX_x(p, i)$ are abbreviated as TX_{broken} and $TX_x(p)$. Moreover, the construction of R_p^R and R_p^L are also not considered. As discussed in section 4.2.1, R_x^R and R_x^L denote the graph-specific reliability of the links in the route graph $ROUTE_x$ and link graph $LINK_x$. This is not only the case for the predicted topology but also for the topologies from which the links are sourced. Moreover, the prediction algorithm does not alter the reliability of a link when it is added to the predicted topology. This means that, as long as it is unambiguously specified from which exact graph the links added to $ROUTE_p$ and $LINK_p$ are sourced, the reliabilities R_p^R and R_p^L are also known.

Lines 2 to 9 first initialise a number of variables. E^{rm} is the set of links broken by the removal of the gateways. Lines 5 to 8 initialise $FLOW_p(i, j)$, $TXF_{br}(i, j)$ and $TXF_{nbr}(i, j)$. $TXF_{br}(i, j)$ holds the total traffic flowing over the broken paths in the flow from node i to node j. Likewise, $TXF_{nbr}(i, j)$ holds the total traffic flowing over all non-broken paths in the flow. Line 9 initialises the link and route graph of the predicted topology.

From line 10 on, each path in the original (current) topology is examined individually. On line 11 the new path, which will be added to $PATHS_p$ later on, is initialised. Next all links in the current path are processed individually (lines 12 to 26). For each link, the algorithm first checks whether the link is still valid and, if so, adds it to the new path without alteration (lines 13 and 14). If the link was broken by the removal of a virtual gateway the algorithm tries to find a replacement link, between the same nodes but using an alternative MAC protocol (lines 16 to 24). This is done in two separate stages since links from the route graph are preferred over those from the link graph (lines 16 and 17). If a replacement is found, it is added to $path_{new}$ instead of the original link (lines 18 to 19). If not, $path_{new}$ is cleared to signal that the path is broken (line 21) and the traffic

flowing over this path is added to the 'broken' traffic of the flow to which the path belongs (line 22). After this, the remaining links of the path are skipped. If, after all links of the paths have been processed, the new path is not deemed to be broken (line 27), it is added to the predicted topology. On line 28 the predicted amount of traffic sent over the new path is initialised to the amount of traffic sent over the original path. On line 29 this traffic is also added tot the total (non-broken) traffic recorded for the flow to which the path belongs. As will be further discussed below, this is done to allow the traffic sent over the broken paths in a flow to be distributed over the remaining (non-broken) paths

ALGORITHM 1: Topology prediction

1 **Begin**
2 ReplacementPolicies $\leftarrow [\text{Policy}_1, \text{Policy}_2, \ldots, \text{Policy}_n]$;
3 $TX_{broken} \leftarrow 0$; $FLOWS_p \leftarrow \emptyset$; $PATHS_p \leftarrow \emptyset$;
4 $E^{rm} \leftarrow \{i \xrightarrow{m} j \in E_c^L | m \neq M_d(i), i \notin GW_p\} \cup \{i \xrightarrow{m} j \in E_c^L | m \neq M_d(j), j \notin GW_p\}$;
5 **for** $FLOW_c(i,j) \in FLOWS_c$ **do**
6 $FLOW_p(i,j) \leftarrow \emptyset$;
7 $TXF_{br}(i,j) \leftarrow 0$; $TXF_{nbr}(i,j) \leftarrow 0$;
8 **end**
9 $E_p^L \leftarrow E_c^L \setminus E^{rm}$; $E_p^R \leftarrow \emptyset$;
10 **for** $path \in PATHS_c$ **do**
11 $path_{new} \leftarrow \emptyset$;
12 **for** $e \in path$ **do**
13 **if** $e \notin E^{rm}$ **then**
14 $path_{new} \leftarrow path_{new} \cup e$;
15 **else**
16 $E_{repl} \leftarrow \{v_i \xrightarrow{m_r} v_j \in E_p^R | \exists m_b \in M : v_i \xrightarrow{m_b} v_j = e\}$;
17 **if** $E_{repl} = \emptyset$ **then** $E_{repl} \leftarrow \{v_i \xrightarrow{m_r} v_j \in E_p^L | \exists m_b \in M : v_i \xrightarrow{m_b} v_j = e\}$;
18 **if** $E_{repl} \neq \emptyset$ **then**
19 $path_{new} \leftarrow path_{new} \cup E_{repl}$;
20 **else**
21 $path_{new} \leftarrow \emptyset$;
22 $TXF_{br}(SRC(path), DST(path)) \leftarrow$
$TXF_{br}(SRC(path), DST(path)) + TX_c(path)$;
23 **break** ;
24 **end**
25 **end**
26 **end**
27 **if** $path_{new} \neq \emptyset$ **then**
28 $TX_p(path_{new}) \leftarrow TX_c(path)$;
29 $TXF_{nbr}(i,j) \leftarrow TXF_{nbr}(i,j) + TX_c(path)$;
30 $PATHS_p \leftarrow PATHS_p \cup \{path_{new}\}$;
31 $FLOW_p(SRC(p), DST(p)) \leftarrow FLOW_p(SRC(p), DST(p)) \cup \{path_{new}\}$;
32 $E_p^R \leftarrow E_p^R \cup path_{new}$;
33 **end**
34 **end**
35 **for** $p \in PATHS_p$ **do**
$TX_p(p) \leftarrow TX_p(p) + \frac{TX_p(p)}{TXF_{nbr}(SRC(p), DST(p))} TXF_{br}(SRC(p), DST(p))$;

```
36  |  FLOWS_broken ← {FLOW_p(i,j) ∈ FLOWS_p | FLOW_p(i,j) = ∅} ;
37  |  for FLOW_c(i,j) ∈ FLOWS_broken do
38  |  |  success ← false;
39  |  |  for k ← 1 to n do
40  |  |  |  (PATHS_r, TX_r, E_r^L, E_r^R) ← ReplacementPolicies[k](i, j, TXF_br(i,j)) ;
41  |  |  |  if PATHS_r ≠ ∅ then
42  |  |  |  |  success ← true ;
43  |  |  |  |  E_p^L ← E_p^L ∪ E_r^L ; E_p^R ← E_p^R ∪ E_r^R ;
44  |  |  |  |  for path ∈ PATHS_r do
45  |  |  |  |  |  if path ∈ PATHS_p then
46  |  |  |  |  |  |  TX_p(path) ← TX_p(path) + TX_r(path)
47  |  |  |  |  |  else
48  |  |  |  |  |  |  TX_p(path) ← TX_r(path) ;
49  |  |  |  |  |  |  PATHS_p ← PATHS_p ∪ {path} ;
50  |  |  |  |  |  |  FLOW_p(SRC(path), DST(path)) ←
    |  |  |  |  |  |    FLOW_p(SRC(path), DST(path)) ∪ {path} ;
51  |  |  |  |  |  end
52  |  |  |  |  end
53  |  |  |  |  break;
54  |  |  |  end
55  |  |  end
56  |  |  if success = false then TX_broken ← TX_broken + TXF_br(i,j) ;
57  |  end
58  |  FLOWS_p ← ⋃_{∀i,j:FLOW_p(i,j)≠∅} FLOW_p(i,j) ;
59  end
```

in that flow. Next, the path itself is added to $PATHS_p$ (line 30) and to the correct flow in the predicted topology (line 31). Lastly the different links in the new path are also added to the route graph (E_p^R) of the predicted topology (line 32).

Once all paths have been processed, the traffic flowing over each path in the predicted topology ($TX_p(p)$) is calculated. $TX_p(p)$ was originally initialised to the traffic flowing over the corresponding path in the original topology ($TX_c(p)$) on line 28, but this initial prediction did not account for the traffic flowing over paths that are broken by the removal of the virtual gateways. Rather than discarding this traffic outright, the prediction algorithm assumes that when a path is broken by the removal of a virtual gateway, the routing protocol will automatically select an alternative route from the same flow. This means that the traffic flowing over the broken paths of the original flow can be redistributed over the remaining paths in the predicted flow. Secondly it is assumed that the traffic distribution over the remaining paths in the flow is not altered. As shown in line 35, the traffic flowing over the broken paths in a flow is therefore distributed over the remaining paths in that flow proportional to the amount of traffic flowing over these paths in the current topology.

Next, the set of broken flows ($FLOWS_{broken}$) is calculated (line 36). These are flows for which all paths were broken by the removal of the virtual gateways. For each of these flows, the different replacement policies are iteratively applied in order to find a suitable

set of replacement flows. (lines 39 to 55). As shown in line 40, each replacement policy returns a tuple of values. $PATHS_r$ is the set of paths that replace the paths in the broken flow. If no replacements could be found, this set is empty. $TX_r(p)$ denotes the predicted amount of traffic sent over each path $p \in PATHS_r$. E_r^L and E_r^R are the sets of links to be added to E_p^L and E_p^R.

Unlike the route repair mechanism, path replacement policies are allowed to use links that are not in E_p^L. This allows them to reuse paths and links that are known to exist (based on historical data) but that are not used in the current configuration and are therefore not present in E_c^L and E_p^L. In addition, the source and destination nodes of the paths generated by these path replacement policies are not required to match the source and destination nodes of the broken flow. This allows path replacement policies to, for instance, re-route traffic to another destination node based on their knowledge of the application running on these nodes. (The 'Equivalent Flow Replacement Policy' introduced in chapter 6 is a good example of this). If the paths of the flow have been successfully replaced, the information returned by the replacement policy is merged into the predicted topology (lines 43 to 52). Line 43 first adds the links generated by the replacement policy to the topology. Next, lines 44 to 52 add the different paths in $PATHS_r$ to the predicted topology. Lines 45 and 46 cover the corner case that a path in $PATHS_r$ is already known in the predicted topology. If the path is not yet known, it is added to the topology on lines 48 to 50. If no replacement paths were found by the replacement policies, the traffic flowing over the broken flow is added to TX_{broken} (line 56). The final step in the algorithm is to calculate $FLOWS_p$ as the union of all non-empty flows in the predicted topology (line 59).

4.2.2.2 Computational Complexity of the Algorithm

To give an idea of the computational complexity of the topology prediction algorithm discussed above, it should first be noted that this algorithm in essence consists of two parts: the first part (lines 2 to 35) tries to repair paths that are broken by the removal of the virtual gateways while the second part (starting on line 36) applies the different replacement policies to replace those flows for which, even after performing local repair, no unbroken paths remain.

The complexity of the first part of the algorithm can be expressed as follows:

$$\mathcal{O}\left(|E_c^L| log\left(|E_c^L|\right) + |PATHS_c| Pmax_c log(|E_c^L|) + |PATHS_c| log(|PATHS_c|)\right)$$

This assumes that all edge and path sets (i.e., E_x^L, E_x^R, $PATHS_x$, ...) are stored using a tree data structure, which implies a complexity for addition, lookup and removal of a single element of $\mathcal{O}(log(n))$. The above formula consists of three separate terms. The first term is due to cost involved in initialising E^{rm} and E_p^L (lines 4 and 9). In the worst case, both these sets can become as large as $|E_c^L|$ which means that the worst-case complexity of constructing these sets is $\mathcal{O}\left(|E_c^L| log\left(|E_c^L|\right)\right)$.

The second term is due to the algorithm iteratively processing all links of all routing paths in $PATHS_c$. For each of these links, the algorithm performs lookups in E^{rm} (line 13), E_p^R (line 16) and E_p^L (line 17). Given that all three of these sets are smaller than $|E_c^L|$ and with $Pmax_c$ denoting the maximum path length encountered in the *current* topology, the complexity of processing all these links is $\mathcal{O}\left(|PATHS_c| Pmax_c log(|E_c^L|)\right)$.

The final term in the above formula is due to post-processing that is done for each path after all the links have been processed (lines 27 to 33). The largest (complexity-wise) cost of this post-processing is the addition of $path_{new}$ to $PATHS_p$ and given that, at this stage of the algorithm, $|PATHS_p| \leq |PATHS_c|$, the worst case complexity of doing so is $\mathcal{O}\left(|PATHS_c| log(|PATHS_c|)\right)$.

The complexity of the second part of the algorithm (from line 36 onwards) cannot be determined from the listing in algorithm 1 alone. The reason for this is that the topology prediction algorithm invokes (line 40) a number of user-configurable path replacement policies to find replacement paths for broken flows. Since the topology prediction algorithm does not impose any computational restrictions on these replacement policies, the overall complexity of the topology prediction algorithm ultimately depends on the complexity of these replacement policies. For this reason, the computational complexity of the replacement policies considered here is briefly discussed with the replacement policies themselves in section 4.2.3.

In addition to invoking the replacement policies, the topology prediction also performs some post-processing to incorporate the paths ($PATHS_r$) and links (E_r^L, E_r^R) returned by the path replacement policy into the predicted topology (lines 43 to 52). Assuming that, over the course of processing all broken flows, the different replacement policies generate a total of $paths_{tot}$ paths and $links_{tot}$ links, the complexity of this post-processing can be expressed as follows:

$$\mathcal{O}\left(paths_{tot} log(|PATHS_c| + paths_{tot}) + links_{tot} log(|E_c^L| + links_{tot})\right)$$

This is basically the computational complexity of adding respectively $paths_{tot}$ and $links_{tot}$ elements to a tree structure that will, after the elements are added, contain at most $|PATHS_c| + paths_{tot}$ and $|E_c^L| + links_{tot}$ elements. Although the computational complexity of the post-processing performed by the topology prediction algorithm thus depends on the *number* of paths and links generated by the path replacement policies it should be noted that, under normal circumstances, the number of links and paths returned by the path replacement policies are expected to be relatively small and that the computational complexity of incorporating these into the predicted topology is thus expected to be less than the complexity of the replacement policies themselves.

4.2.3 Path replacement policies

When the removal of one or more Virtual Gateways causes all paths in a certain flow to be broken, the prediction algorithm iteratively applies a number of replacement policies to attempt to repair the flow. Each replacement policy makes a number of specific assumptions about the traffic flows in the network in order to build a set of plausible replacement paths based on data from the *minimum* and *maximum* topologies and the already predicted paths in the *predicted* topology. In this work three replacement policies are used that only make relatively general assumptions about the traffic flows in the network to build replacement paths. More specialised replacement policies can easily be added to take full advantage of use case specific network properties. One assumption shared by all three replacement policies is that links which were observed in the past and that are not affected by the removal of any gateway nodes are still available even if they

were not observed during the most recent iteration. This in essence, means that E_p^L may be extended with links from E_{min}^L when necessary.

The 'Reverse Flow Policy' $RFlow_x$ builds replacement paths for a broken flow by assuming that data packets can flow in two directions over a routing path. x denotes the specific topology used to build the replacement paths and, as with the notations defined in section 4.2.1, can refer to either the predicted, current, minimum or maximum topology. To find replacement paths for a flow from node i to node j, it first checks whether a flow from j to i exists in topology x. If so, it reverses all the paths in that flow to obtain a set of replacement paths for the original flow from i to j.

ALGORITHM 2: Reverse Flow Replacement Policy

```
1  Function RFlow_x(i , j , TX_broken)
       Result: (PATHS_r, TX_r, E_r^L, E_r^R)
       // Initialise variables
2      PATHS_r ← ∅ ; E_r^L ← ∅ ; E_r^R ← ∅ ;
3      if FLOW_x(j, i) ≠ ∅ then
4          for path ∈ FLOW_x(j, i) do
5              candidate ← { n_1 -m-> n_2 | n_2 -m-> n_1 ∈ path } ;
6              (path_r, E_l, E_r) = SourceLinks (candidate) ;
7              if path_r ≠ ∅ then
8                  PATHS_r ← PATHS_r ∪ {path_r} ;
9                  E_r^L ← E_r^L ∪ E_l ; E_r^R ← E_r^R ∪ E_r ;
10             end
11         end
12         if PATHS_r ≠ ∅ then
13             TX_tot = Σ_{path∈PATHS_r} TX_x (path) ;
14             for path ∈ PATHS_r do
15                 TX_r(path) ← (TX_x(path) / TX_tot) TX_broken ;
16             end
17         end
18     end
19 end
```

The operation of the 'Reverse Flow Policy' is shown in Algorithm 2. On line 2 first a number of variables are initialised. Next, the replacement policy checks whether a reverse flow exists in the topology (line 3). If so, the paths in the reverse flow are processed individually to build the replacement paths. On line 5 a candidate replacement path is constructed by inversing all the links in the original path. This candidate path is subsequently passed to the *SourceLinks* algorithm. This algorithm checks whether the specified path is a plausible replacement path for the broken flow. This is done by comparing the links in the candidate path to those in the route- and link graph of both the predicted and minimal topology. If all links in the candidate path can be found, the path is deemed to be a plausible replacement. The details of the algorithm are shown in Algorithm 3. The *SourceLinks* algorithm returns a tuple of values (line 6). $path_r$ is the replacement path to be added to $PATHS_r$ if not all links in the candidate path could be found, this path is empty. E_l and E_r are the links to be added to E_r^L and

E_r^R respectively. This is done on line 9. Once all paths of the reverse flow have been processed, the TX_r values are calculated for the individual paths. This is done similarly to the traffic calculations of the topology prediction algorithm.

To give an idea of the computational complexity of the 'Reverse Flow Policy' ($RFlow_x$) it should first be noted that the complexity of the *SourceLinks* algorithm is

$$\mathcal{O}\left(|candidate|(log(|E_p^L|) + log(|E_{min}^L|)\right)$$

This is becaisefor every link in the candidate-path several lookups in the route- and link graphs of the predicted and minimum topology need to happen. Since $RFlow_x$ invokes the *SourceLinks* algorithm once for every path in $FLOW_x(j,i)$ and since all of these paths contain at most $Pmax_x$ links (see section 4.2.2.2), the overall complexity of the 'Reverse Flow Policy' can be expressed as:

$$\mathcal{O}\left(|FLOW_x(j,i)|Pmax_x(log(|E_p^L|) + log(|E_{min}^L|))\right)$$

ALGORITHM 3: SourceLinks algorithm used by the Replacement Policies

```
1  Function SourceLinks(candidate)
       Result: (path, E_l, E_r)
2      path ← ∅ ; E_r ← ∅; E_l ← ∅;
3      for x ∈ {p, min} do
4          R ← (E_x^R ∩ candidate) \ path ;
5          L ← (E_x^L ∩ candidate) \ (path ∪ R) ;
6          path ← path ∪ R ∪ L ; E_r ← E_r ∪ R ; E_l ← E_l ∪ L ;
7      end
8      if |candidate| ≠ |path| then
9          path ← ∅ ; E_r ← ∅ ; E_l ← ∅ ;
10     end
11 end
```

The 'MinGraph Policy' attempts to replace broken paths with equivalent paths from the minimum topology. This is based on the assumption that paths of the 'minimum' topology can be reused regardless of the gateway configuration since $\forall GW_x : GW_{min} \subseteq GW_x$. The MinGraph Policy operates very similarly to the reverse flow policy. The only significant differences are that flows are sourced from $FLOWS_{min}$ instead of $FLOWS_x$ and that paths are not reversed before they are passed to *SourceLinks*. As a result, the complexity of the 'MinGraph Policy' will also be similar to that of the reverse flow policy.

The 'Concatenation Policy' $Concat_x$ replaces broken paths by concatenating sections of paths that are already known to exist in topology x. More specifically, all paths in $PATHS_x$ are scanned for subsections that connect the source and destination node of the broken flow with one of the gateways in GW_p. These 'subpaths' are then concatenated to create a plausible path from the source to the destination node.

The operation of the 'Concatenation Policy' is shown in Algorithm 4. After first initialising a number of variables (line 2), the concatenation policy tries to construct a path from i to j through each of the virtual gateways. As shown on line 3, only gateways from $GW_x \cap GW_p$ are considered. This is done to avoid building paths through gateways that

ALGORITHM 4: Concatenation Policy

1 **Function** $Concat_x(i\ ,\ j\ ,\ TX_{broken})$
  Result: $(PATHS_r, TX_r, E_r^L, E_r^R)$
  `// Initialise variables`
2 $PATHS_r \leftarrow \emptyset$; $E_r^L \leftarrow \emptyset$; $E_r^R \leftarrow \emptyset$;
3 **for** $gw \in (GW_x \cap GW_p)$ **do**
4 $(p_s, TX_s, E_s^L, E_s^R) \leftarrow FindSubPath(i, gw)$;
5 $(p_d, TX_d, E_d^L, E_d^R) \leftarrow FindSubPath(gw, j)$;
6 **if** $p_s \neq \emptyset \wedge p_d \neq \emptyset$ **then**
7 $path \leftarrow p_s \cup p_d$;
8 $PATHS_r \leftarrow PATHS_r \cup \{path\}$;
9 $E_r^L \leftarrow E_r^L \cup E_s^L \cup E_d^L$; $E_r^R \leftarrow E_r^R \cup E_s^R \cup E_d^R$;
10 $TXE(path) \leftarrow TX_s$;
11 **end**
12 **end**
13 $TX_{tot} \leftarrow \sum_{p \in PATHS_r} TXE(p)$;
14 **for** $path \in PATHS_r$ **do** $TX_r(path) \leftarrow \frac{TXE(path)}{TX_{tot}} TX_{broken}$;
15 **end**
16 **Function** $FindSubPath(i\ ,\ j)$
  Result: $(path, TX, E_l, E_r)$
17 $path \leftarrow \emptyset$; $TX \leftarrow 0$; $E_l \leftarrow \emptyset$; $E_r \leftarrow \emptyset$;
18 $P_{sub} \leftarrow \{p = \{i \overset{m_1}{\rightarrow} n_1, n_1 \overset{m_2}{\rightarrow} n2, \ldots, n_k \overset{m_k}{\rightarrow} j\} | \exists q \in PATHS_x : p \subseteq q \wedge SourceLinks(p) \neq \emptyset\}$;
19 **if** $P_{sub} \neq \emptyset$ **then**
20 $path \leftarrow \text{argmax}_{p \in P_{sub}} \sum_{q \in PATHS_x : p \subseteq q} TX_x(q)$;
21 $TX \leftarrow \sum_{q \in PATHS_x : path \subseteq q} TX_x(q)$;
22 $(path, E_l, E_r) \leftarrow SourceLinks(path)$;
23 **end**
24 **end**

are no longer enabled in the predicted topology. For each gateway gw, the $FindSubPath$ algorithm is invoked to find a single suitable path from the source node i to the gateway (line 4). The details of this algorithm are shown in lines 16 - 24 and are discussed further below. The $FindSubPath$ algorithm returns a tuple of values. p_s is the discovered path from the source node to the selected gateway gw. If no path could be found, $p_s = \emptyset$. TX_s is the amount of traffic flowing over this path in topology x. E_s^L and E_s^R are the links to be added to the link graph and route graph. On line 5 the $FindSubPath$ algorithm is invoked again, this time to find a path from the gateway to the destination node. It should be noted that although multiple paths may exist between node i and gw (and likewise gw and j), the $FindSubPath$ algorithm will only return a single path. This is because the total number of paths between i and j rises rapidly with the number of subpaths returned by $FindSubPath$. Since it is highly unlikely that all these possible combinations will actually be used, only the subpath that carries the most traffic is returned. If $FindSubPath$ was able to find a suitable path from i to gw and from gw to j (line 6), these subpaths are concatenated to form a path from i to j (line 7). That path is then added to $PATHS_r$ and E_r^L and E_r^R are updated accordingly (lines 8-9). Finally TX_s is

recorded in the $TXE(path)$ variable (line 10). $TXE(path)$ denotes the relative amount of traffic that is estimated to flow over the given path and is later used to distribute the traffic in the broken flow (TX_{broken}) over the different paths in $PATHS_r$ (lines 13 - 14). TX_s is used, rather than for instance TX_d, to estimate the traffic flowing over a given path. This is because once the routing path has been chosen, the amount of traffic sent over this path is determined by the sender and not by the receiver. The details of the $FindSubPath$ algorithm itself are shown on lines 16 to 24. Given a source node i and a destination node j, this algorithm selects a plausible path between the two nodes based on the paths in $PATH_x$. To this end first the set of all 'subpaths' P_{sub} is selected on line 18. P_{sub} consists of all paths from node i to node j that are part of a larger path $q \in PATHS_x$. Moreover all paths in P_{sub} must be plausible replacement paths, which means that the $SourceLinks$ algorithm may not return an empty path. This is abbreviated in Algorithm 4 as $SourceLinks(p) \neq \emptyset$. The path to be returned is selected on line 20. As discussed above, this is the subpath over which most packets are sent.

To give an idea of the computational complexity of this path replacement policy, the complexity of the $FindSubPath$ algorithm must first be determined. To do so, it is worth noting that the complexity-wise cost of this algorithm is mostly due to the fact that it needs to scan all paths in $PATHS_x$ for a subpath linking nodes i and j and then, once such a path has been found, pass it on to the $SourceLinks$ algorithm to check whether this path is valid. Given that scanning a path for a subpath can be done in linear time, the complexity of processing a single path p can be expressed as

$$\mathcal{O}\left(|p| + |p|(log(|E_p^L|) + log(|E_{min}^L|))\right)$$

Given moreover that

$$\forall p \in PATHS_x : |p| \leq Pmax_x$$
$$\mathcal{O}(|p|) < \mathcal{O}\left(|p|(log(|E_p^L|) + log(|E_{min}^L|))\right)$$

the complexity of the $FindSubPath$ algorithm can be simplified to

$$\mathcal{O}\left(|Pmax_x|(log(|E_p^L|) + log(|E_{min}^L|))\right)$$

Since $Concat_x$ invokes the $FindSubPath$ algorithm twice for each virtual gateway in $GW_x \cap GW_p$ and $|GW_x \cap GW_p| \leq |GW_x|$, this means that the overall complexity of the concatenation policy can be expressed as:

$$\mathcal{O}\left(|GW_x||Pmax_x|(log(|E_p^L|) + log(|E_{min}^L|))\right)$$

It should be noted that the result of the topology prediction algorithm depends on the order in which the different Replacement Policies are applied. This in turn depends on the specific initialisation of the $ReplacementPolicies$ variable in Algorithm 1. Within the scope of this work, this variable is initialised as follows:

$$ReplacementPolicies = \{RFlow_p, MinGraph, RFlow_{min}, Concat_p, Concat_{min}\}$$

4.2.4 Performance Estimation

The final step in the prediction algorithm is to calculate the performance of the *predicted* topology. Three metrics are considered: duty cycle, hop count and reliability and these metrics are calculated for each network individually.

In accordance with the notations that will be introduced in chapter 5, NW denotes the set of all networks and $V(i)$ is the set of all nodes in network i.

$avgDC_p(i)$ is the average node duty cycle for network i and is calculated as follows:

$$avgDC_p(i) = \frac{\sum_{v \in V(i)} DC_p(v)}{|V(i)|}$$

As discussed in section 4.2.1, $DC_p(v)$ is the predicted duty cycle of node v.

The average reliability and average hop count for the individual networks are calculated from the predicted route graph ($ROUTE_p$) and the predicted number of packets sent for each network over the different paths ($TX_p(p,i)$). Since, as discussed in section 4.1.3, traffic is attributed to a network based on which node benefits from the traffic being transmitted, rather than based on the source and destination node, $avgR_p(i)$ and $avgHC_p(i)$ are referred to as the average *node* reliability and average *node* hop count for network i.

$$avgR_p(i) = \frac{\sum\limits_{p \in PATHS_p} R_{path_p}(p) TX_p(p,i)}{\sum\limits_{p \in PATHS_p} TX_p(p,i) + TX_{broken(i)}}$$

$$avgHC_p(i) = \frac{\sum\limits_{p \in PATHS_p} HC_{path_p}(p) TX_p(p,i)}{\sum\limits_{p \in PATHS_p} TX_p(p,i)}$$

In the above equations $R_{path_p}(p)$ and $HC_{path_p}(p)$ are the predicted end-to-end path reliability and predicted path length. Also note the presence of TX_{broken} in the denomintator of the formula for $avgR_p(i)$. This value is present to account for the traffic sent over paths that were broken by the removal of the gateways and for which no plausible replacement path could be predicted. $R_{path_p}(p)$ and $HC_{path_p}(p)$ are calculated as follows:

$$R_{path_p}(p) = \prod_{l \in p} R_p^R(l)$$

$$HC_{path_p}(p) = |p|$$

4.3 Evaluation

The accuracy of the prediction algorithm is evaluated using the simulator introduced in chapter 2. For all tests discussed here, the random-flows scenario is used with 50 nodes deployed along the same two 5x5 random grids as used in chapter 2, an overlap of 50% and an average data generation interval of 20 seconds. All other simulation parameters are as they are described in section 2.2.3. For each set of parameters, 100 independent test runs (using different seeds for the Castalia Simulator) are performed. Since the evaluation of the prediction algorithm requires multiple gateway configurations to be evaluated within a single test run (i.e., different gateway configurations need to be simulated) each simulation within a single test run uses the same seed for the random

number generator. This ensures that the prediction error measured is only due to the prediction algorithm itself. Moreover, the same set of random flows (see section 2.2.2) is also used for all simulations within a single test run. The set of 'random flows' does differ between test runs. In addition, to ensure that a sufficient amount of inter-network traffic is generated, 80% of these flows have endpoints in different networks.

The prediction algorithm is evaluated for each possible combination of the four MAC protocols considered in this thesis. In addition, the case where two (non-synchronized) TDMA MAC protocols are used is also investigated. When doing so, three classes of virtual gateways need to be considered. These classes of virtual gateways differ in how broad their removal will affect the current route topology, how extensive the assumptions that need to be made to predict the resulting topology actually are and subsequently how accurate the resulting prediction is likely to be.

1. **Non-Border gateways:** These are virtual gateways whose removal should not require any change to the current route topology. This, in essence, means that these gateways neither send data to nor receive data from 'foreign' nodes (nodes in a different network). Otherwise, the route topology would be affected by their removal. Since these gateways, despite running multiple MAC protocols, are therefore *not* located on the border between the two networks, the prediction algorithm only needs to predict the duty cycle of the nodes and doesn't need to repair or replace any broken paths.

2. **Redundant gateways:** These are virtual gateways whose removal should not require any change to the current route topology apart from changing the MAC protocol on their inter-network links. This, in essence, means that the gateway is either a non-border gateway or that all of its 'foreign' neighbours are also virtual gateways. It should be noted that if two neighbouring gateways are located in different networks they can only be considered to both be 'redundant' if they are not removed at the same time, since otherwise it wouldn't be possible to maintain the route topology by only changing the MAC protocol on the link between them. Removing a redundant gateway will cause a number of routing paths to be broken but since there is an alternative MAC protocol available for each broken link, the prediction algorithm should be able to repair any 'broken' paths using only the 'local repair' mechanism.

3. **Non-redundant gateways:** These are virtual gateways that communicate with 'foreign' nodes but that do not meet the requirements of a redundant gateway, such as virtual gateways whose foreign neighbours include regular nodes as well as virtual gateway nodes. As a result, removing a non-redundant gateway will cause a number of routing paths to be broken that can't be repaired using the 'local repair' mechanism. Consequently, the prediction algorithm will either have to re-route the traffic of the broken path over the other paths in the flow and may even have to resort to applying one or more replacement policies if the entire flow is broken.

Since the accuracy of the prediction algorithm is expected to vary with the class of the removed virtual gateways, the prediction algorithm is evaluated separately for each class of virtual gateways.

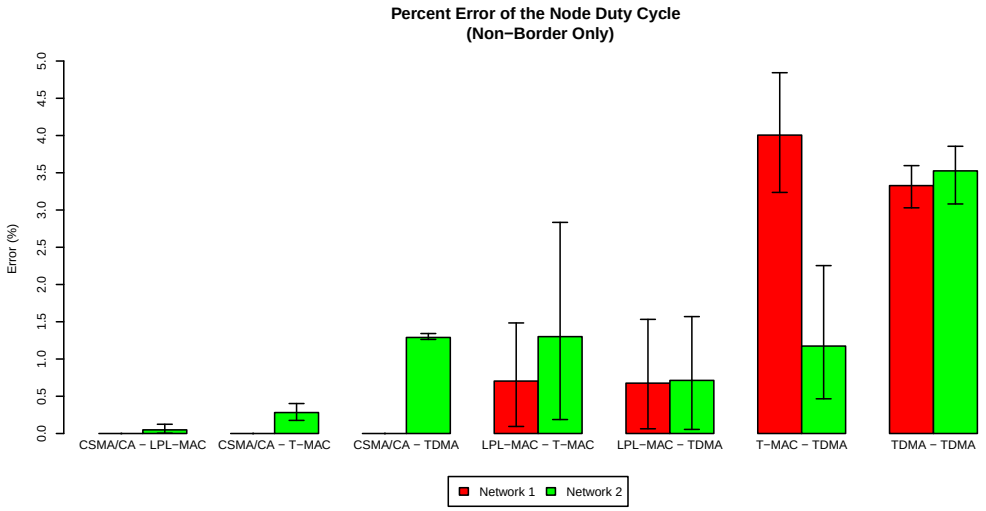


Figure 4.3: Average, 5- and 95%-tile of the percent error of the node duty cycle for the 'Non-Border' scenario.

4.3.1 Removal of non-border gateways

The accuracy of the prediction algorithm for the non-border gateway scenario is evaluated over a number of different test runs. During each test-run the *maximum* and *minumum* gateway configuration are first evaluated, using the Castalia simulator, to obtain the topology information required for the prediction algorithm to operate (see section 4.2). Next, all the non-border gateways in the maximum configuration are identified and the prediction algorithm is applied to predict the duty cycle, reliability and hop count resulting from their removal. Next the non-border gateways are removed from the maximum configuration and the resulting gateway configuration is also evaluated using the Castalia simulator to obtain the actual values for the duty cycle, reliability and hop count metrics. Finally, the *percent error* is calculated as follows for each metric.

$$percent\ error = 100 * \frac{|actual\text{-}predicted|}{actual}$$

In the above formula *actual* is the value of the metric and *predicted* is the value of the metric predicted by the prediction algorithm.

Figure 4.3 shows the average error of the node duty cycle for all combinations of MAC protocols. The 5- and 95-percentile are also included as an indication of the best- and worst case scenario. As for the other considered metrics, the bars are used to display the average error and the whiskers show the 5- and 95-percentile. For most combinations of MAC protocols the prediction error of the node duty cycle is very low with averages around 1.5% and 95-percentiles around 3%. In the case when two (non-synchronized) TDMA MAC protocols are used the average duty cycle is a bit higher (around 3.6%) and when T-MAC is combined with TDMA, the duty cycle error for the T-MAC network is between 4% and 5%. Even for these two scenarios however, the predicted duty cycle is quite accurate given that the prediction is made without any knowledge about the actual

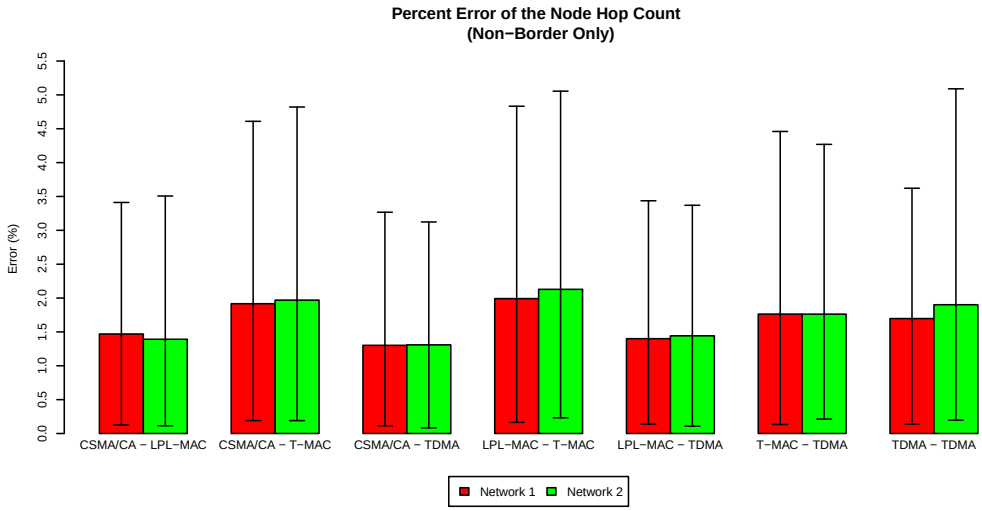


Figure 4.4: Average, 5% and 95%-tile of the percent error of the node hop count for the 'Non-Border' scenario.

MAC protocols used.

The average error of the node hop count is shown in Figure 4.4. For all combinations of MAC protocols the average percent error hovers between 1.5% and 2.2%. The 95-percentile error is a little bit higher and hovers between 3.5% and 5.5%. Initially, it may seem strange that there is any prediction error at all in this test case given that the removal of non-border gateways should, by definition, not require any change to the routing topology. One might therefore expect the routing topology to be unaffected by the removal of these gateways and expect the prediction error in this case to be zero. This is however not necessarily the case. It is true that the removal of non-border gateways indeed does not *require* any changes to be made to the route topology. This however does not prevent the routing protocols of the networks to select more optimal routes if these become available. Moreover, the removal of the non-border gateways will reduce the amount of interference between the heterogeneous MAC protocols, which in turn may allow the routing protocol to select more optimal (i.e., shorter) paths. Since these indirect effects are not taken into account by the prediction algorithm a certain prediction error is expected to occur and given that the prediction algorithm has no prior knowledge about the routing protocols used, the error values shown in Figure 4.4 are actually very low.

As with the node hop count, a certain error is also expected to occur in the prediction of the node reliability. As shown in figure 4.5 however, this error is very small. For most combinations of MAC protocols the average node reliability error is situated between 0.5% and 2% while the 95-percentile error is lower than 4%. The only exception is the case where two (non-synchronised) TDMA MAC protocols are used. In that case, the average error can be as high as 6.5% and the 95-percentile tops out at around 10.5%. These larger errors can be explained by the fact that the TDMA MAC protocol considered in this thesis does not use clear channel assessments or acknowledgements. (As discussed

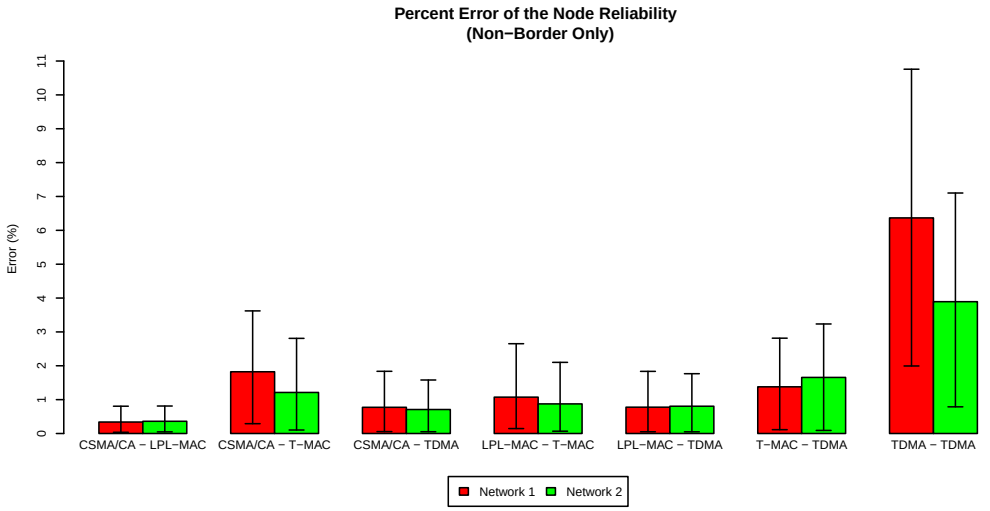


Figure 4.5: Average, 5% and 95%-tile of the percent error of the node reliability for the 'Non-Border' scenario.

in section 2.1.7, this is because these mechanisms are also not considered by most of the existing TDMA-based sensor network MAC protocols). As with all MAC protocols the link reliability depends, at least partially, on the timing mechanisms of the MAC protocol since a packet can only be received correctly if it does not collide with another, simultaneous, transmission. Since TDMA neither tries to avoid (no CCA) nor recover from (no acknowledgements) such collisions, the specific time at which packets are sent has a much more profound effect on the link reliability than for the other considered MAC protocols. The impact of these timing effects is moreover compounded when two TDMA MAC protocols are used since in that case TDMA does not benefit from the CCA-mechanisms employed by the other MAC protocol. To remain independent from the used MAC protocols, these timing effects are not taken into account by the prediction algorithm which results in the somewhat larger prediction error for the TDMA - TDMA case shown in Figure 4.5.

4.3.2 Removal of redundant gateways

As for the non-border gateway case, the accuracy of the prediction algorithm for the redundant-gateways scenario is also evaluated over multiple test runs and during each test-run first the *maximum* and *minimum* gateway configuration are evaluated to obtain the topology information required by the prediction algorithm.

In addition to the set of MAC protocols used, two additional parameters are also considered: the 'initial number of gateways' and the 'number of gateways are removed'. The 'initial number of gateways' is the number of gateways present in the *current* configuration that is passed to the prediction algorithm. This number is varied from 50 to 35 gateways in steps of 5 gateways to investigate how the accuracy of the prediction algorithm evolves when gateways are iteratively removed by the IRVG mechanism. The 'number of gateways removed' is the number of gateways that are removed in a single step

from the current gateway configuration. This number is varied from 1 to 10 gateways to investigate how accurate the predictions of the prediction algorithm are when increasingly larger numbers of gateways removed in a single step. It should be noted that for the case where the 'Initial number of gateways' is 35, the 'number of gateways removed' is at most 5 to ensure that all removed gateways are in fact *redundant* gateways.

For each combination of parameters, first the 'current' topology is generated by removing randomly selected *redundant* gateways from the 'maximum' configuration so the current configuration has the correct 'initial number of gateways'. This configuration is then evaluated using the Castalia Simulator to obtain the performance and topology-information required by the prediction algorithm. Next, the correct number of, again, *redundant* gateways from this configuration are randomly selected for removal and the prediction algorithm is applied to predict the effect of their removal on the node hop count, node duty cycle and node reliability. Finally, the selected gateways are removed, the resulting gateway configuration is evaluated using the Castalia simulator and the prediction error is calculated from the predicted and the actual values of the performance metrics.

4.3.2.1 Duty Cycle

The prediction error for the duty cycle metric is shown in figures 4.6 to 4.12. Each figure shows the average and 95-percentile of the observed prediction error separately for each of the two networks. The average error is shown using continuous lines while the 95-percentile of the error, which is included as an indication of the worst case scenario, is shown using dotted lines. Finally, different line colours are used to differentiate between different values of the 'initial number of gateways' parameter.

Figures 4.6, 4.7 and 4.8 show the prediction error of the duty cycle for the cases where CSMA/CA is used by one of the two networks. For the network using the CSMA/CA protocol, the prediction error is always zero. This is hardly surprising given that CS-

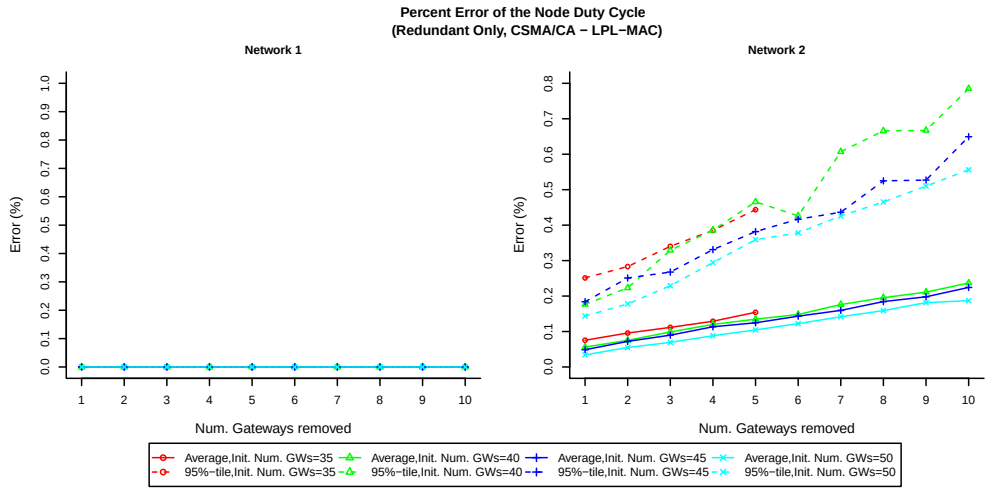


Figure 4.6: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using the CSMA/CA and LPL-MAC protocols.

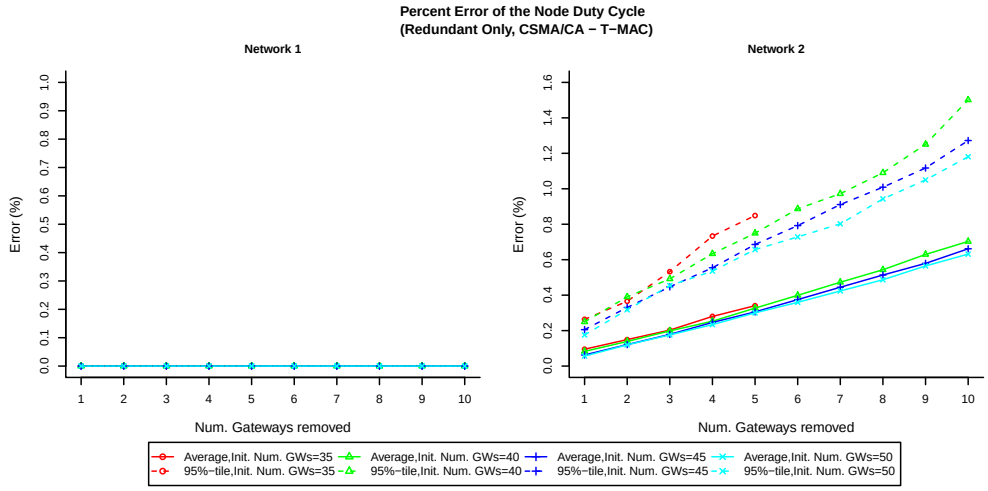


Figure 4.7: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using the CSMA/CA and T-MAC protocols.

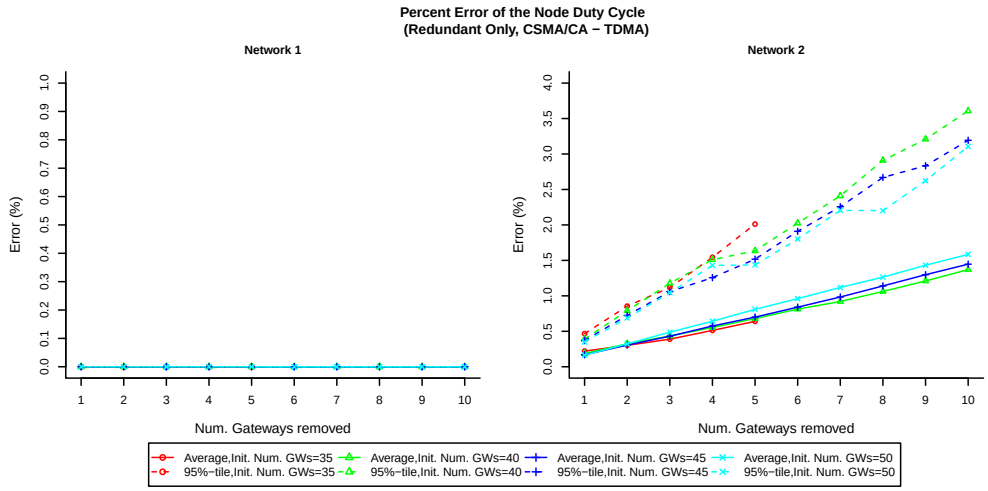


Figure 4.8: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using the CSMA/CA and TDMA MAC protocols.

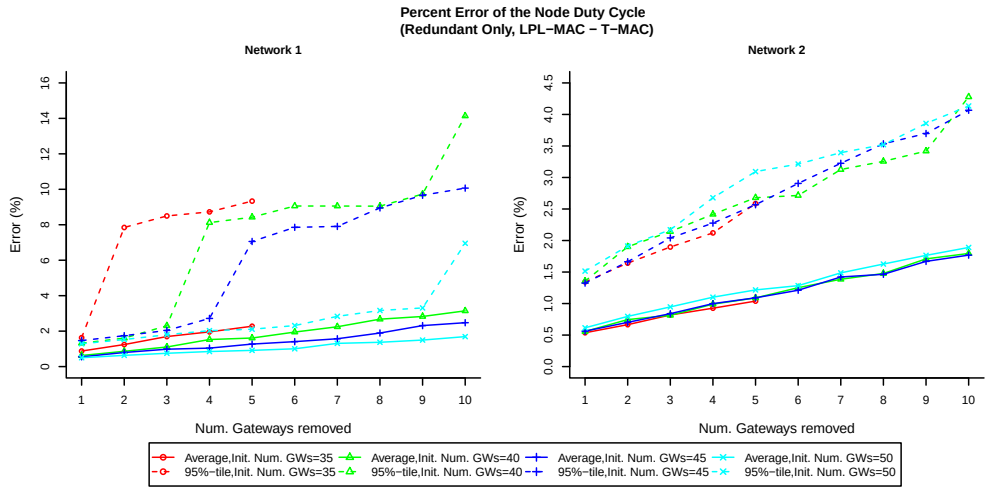


Figure 4.9: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using the LPL-MAC and T-MAC protocols.

MA/CA always keeps the radio enabled and that the duty cycle will therefore always be 100%. For the other network (using either LPL-MAC, T-MAC or TDMA), the prediction error can clearly be seen to rise with the number of gateways that are removed in a single step. This behaviour is expected since the prediction algorithm uses a very simple method to predict the duty cycle of individual nodes and the resulting error accumulates when more gateways are removed in a single step. For the LPL-MAC protocol the 'Initial number of Gateways' used also has an effect on the prediction error as the error is larger when there are fewer gateways to start from. This effect may be explained by the fact that interference between heterogeneous MAC protocols not only affects the duty cycle of the nodes engaged in a specific traffic flow, but also the duty cycle of all surrounding nodes. Consequently, the removal of a virtual gateway also affects the duty cycle of the surrounding nodes, despite this not being taken into account by the prediction algorithm. Moreover, this effect is more profound when fewer initial gateways are used since in that case the removal of a single virtual gateway results in a proportionally larger reduction in interference experienced by the surrounding nodes. This is true for all MAC protocols, but since nodes using the LPL-MAC protocol are woken up by any nearby transmission, the effect is somewhat more visible when CSMA/CA is combined with LPL-MAC than in the case where it is combined with either T-MAC or TDMA. Despite these limitations however, for the three CSMA/CA-cases, the error is still very small with a worst case error lower than 3.6% over the entire range of the considered parameters.

The prediction error of the duty cycle for the cases where LPL-MAC is combined with either T-MAC or TDMA is shown in Figures 4.9 and 4.10. For the T-MAC and TDMA networks, the prediction error follows a similar pattern when these networks are combined with LPL-MAC instead of CSMA/CA. The only difference is that the prediction error is somewhat higher (maximum 4% instead of 1.6% for T-MAC and 6% instead of 4% for TDMA), but still relatively small. For the LPL-MAC network the average prediction error is also relatively small (less than 4%). The 95-percentile of the prediction error

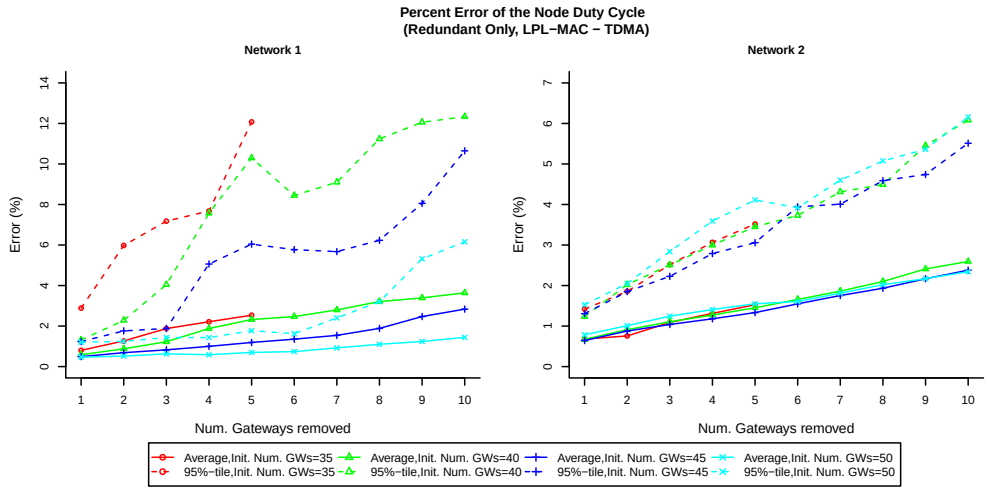


Figure 4.10: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using the LPL-MAC and TDMA MAC protocols.

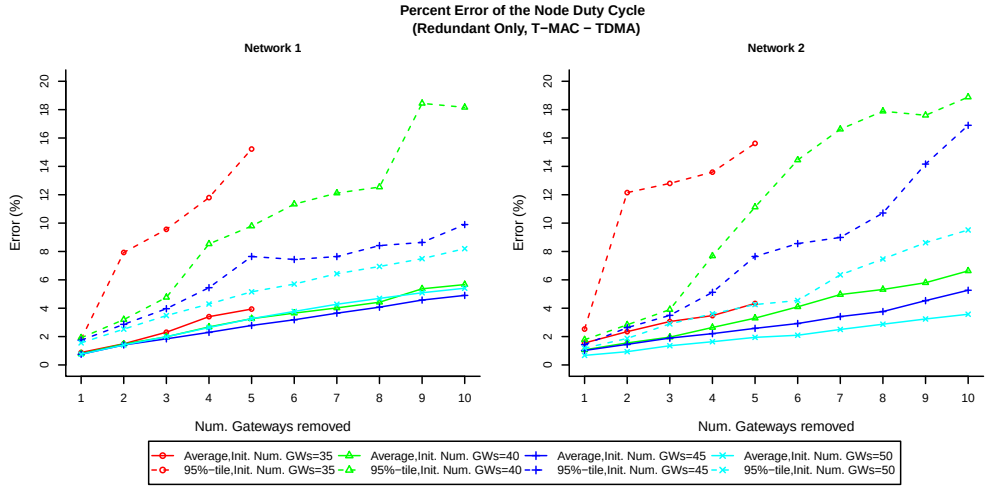


Figure 4.11: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using the T-MAC and TDMA MAC protocols.

however rises much more rapidly than the average error and depending on the number of gateways removed and the initial number of gateways can become as high as 12% when combined with TDMA and as high as 15% when combined with T-MAC.

In the T-MAC - TDMA case (see figure 4.11) the prediction error once again rises both when the initial number of gateways is reduced or when the number of removed gateways is increased. For both networks the average prediction error rises relatively slowly and while it can still be considered to be relatively small when only a few gateways are removed, it can rise to around 6.5% when more gateways are removed. The 95-percentile of the

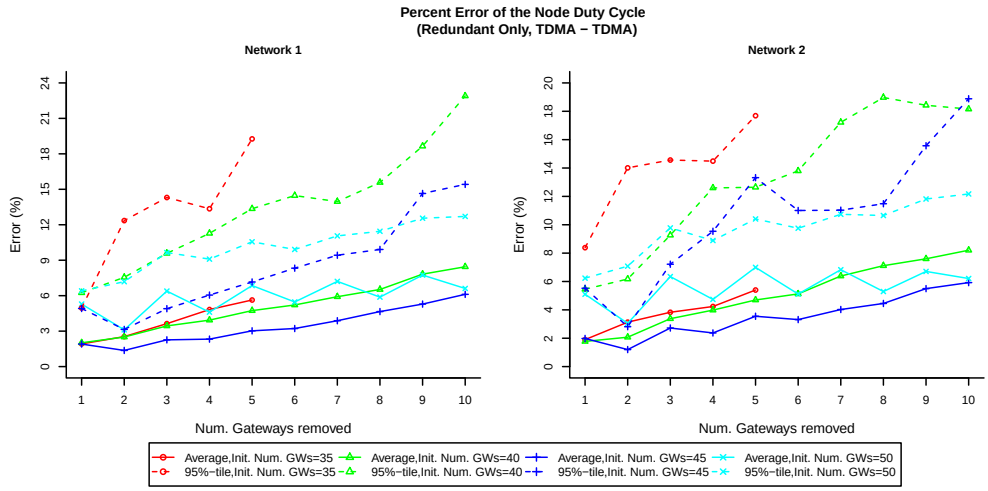


Figure 4.12: Average and 95%-tile of the percent error of the node duty cycle for the 'Redundant' scenario using two TDMA MAC protocols.

prediction error however rises much more rapidly and become as large as 19%.

Although a similar trend can be observed for the TDMA - TDMA case, figure 4.12 also makes it clear that in this case the node duty cycle varies significantly with only minor changes made to either the 'Number of Gateways removed' or the 'Initial number of gateways' parameter. This somewhat more 'erratic' behaviour is most likely caused by the fact that, when both networks use a TDMA MAC protocol, the effect that the interference between these MAC protocols has on the node duty cycle, will for a very significant portion depend on the specific timings and thus the slot allocation used by the nodes. The fluctuations observed in figure 4.12 are thus most likely caused by the fact that these slot allocations depend on the exact number of nodes adhering to the timing of each MAC protocol and thus also on the specific number of virtual gateways used. Figure 4.12 also shows that the prediction errors observed for the TDMA - TDMA case can become relatively high. Depending on the initial number of gateways and the number of gateways removed, the average prediction error can rise to around 8.5% while the 95-percentile error can become as high as 23%.

While, depending on the combination of MAC protocols and the number of gateways used and/or removed the prediction error can thus become quite high, this doesn't necessarily pose a problem for the gateway selection algorithm. After all, these very large prediction errors mostly occur when the number of removed gateways is extremely high and in those cases the prediction error can be reduced significantly by choosing a more conservative value for the 'number of gateways removed'-parameter. As a result, the number of gateways to remove in a single step will be an important consideration in the design of the selection algorithm.

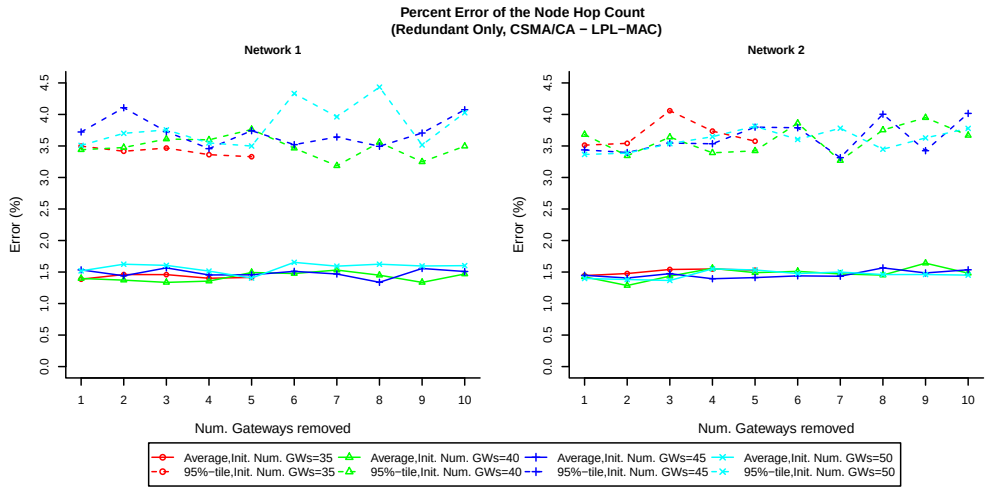


Figure 4.13: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using the CSMA/CA and LPL-MAC protocols.

4.3.2.2 Hop Count

The prediction error for the node hop count metric is shown in Figures 4.13 to 4.19. As these figures show, the prediction error behaves somewhat differently depending on whether or not the T-MAC protocol was used.

In all cases where the T-MAC protocol is used, the prediction error can clearly be seen to rise with the number of removed gateways. This behaviour is not unexpected since, as discussed in section 4.3.1, the reduction in interference resulting from the removal of one or more gateways may cause the routing protocols of the network to establish slightly different paths even if no paths were broken by the removal of these gateways. Given that the interference is expected to drop with the number of removed gateways, the difference between the predicted and the actual route topology will also be larger and as a result the same is true for the prediction error. Although the worst case error observed is nowhere near as high as for the node duty cycle metric, the error can still become quite high with averages topping out at 4.5% and 95-percentiles as high as 10%. Luckily these higher error values only occur when an extremely high number of gateways is removed in a single step and, as with the duty cycle metric, the prediction error can be reduced significantly by removing fewer gateways at the same time.

When the T-MAC protocol is *not* used, the prediction error is stable across all values for the 'Initial number of Gateways' and 'Number of Gateways removed' parameter. Moreover, these error values correspond to the error values observed for the non-border scenario with averages hovering between 1.5 and 2% and 95-percentiles topping out at 5%. The only exception is when 50 initial gateways are used in the TDMA - TDMA scenario, in which case the error observed for 'Network 2' is visibly larger than for the other 'initial number of gateways'-values. This could be caused by the fact that, as explained in section 4.3.1 the behaviour of the TDMA protocol is somewhat harder to predict than that of the other MAC protocols, especially when combined with another TDMA MAC protocol.

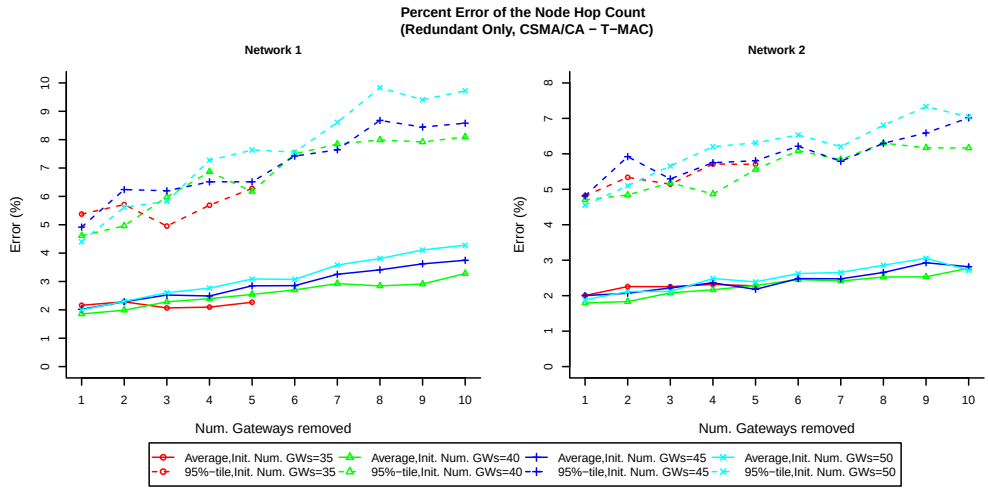


Figure 4.14: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using the CSMA/CA and T-MAC protocols.

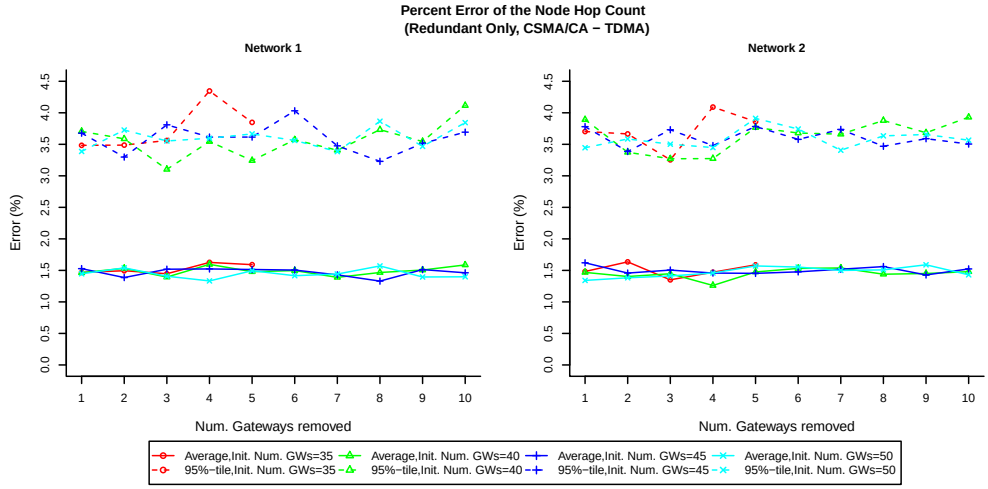


Figure 4.15: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using the CSMA/CA and TDMA protocols.

Moreover, the case where 50 initial gateways are used is also the worst case scenario for interference between two non-synchronized TDMA networks, which might explain why the prediction error is only higher in this particular case. Finally, it should be noted that although, as discussed in section 4.1.3, the traffic flows are tracked separately for each network, each flow still traverses nodes of both networks. Because of this, the difference in prediction error between the two networks, is not necessarily caused by the MAC protocol behaving differently in one network over the other. It only signifies that for this particular case the traffic flows from one network are slightly more affected than the other.

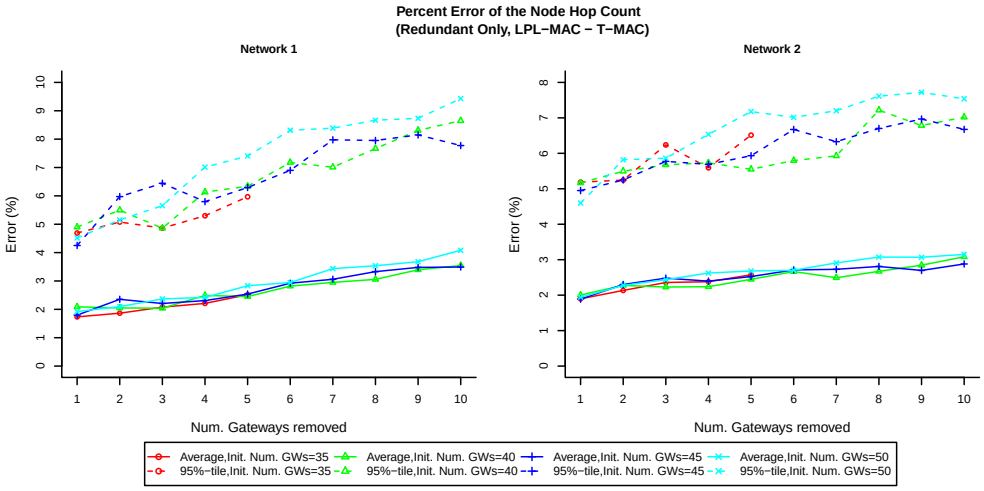


Figure 4.16: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using the LPL-MAC and T-MAC protocols.

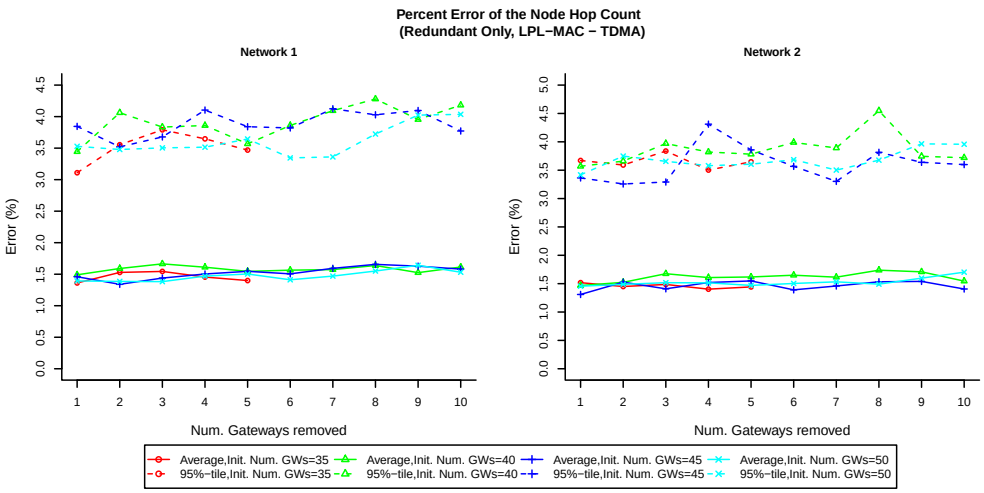


Figure 4.17: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using the LPL-MAC and TDMA protocols.

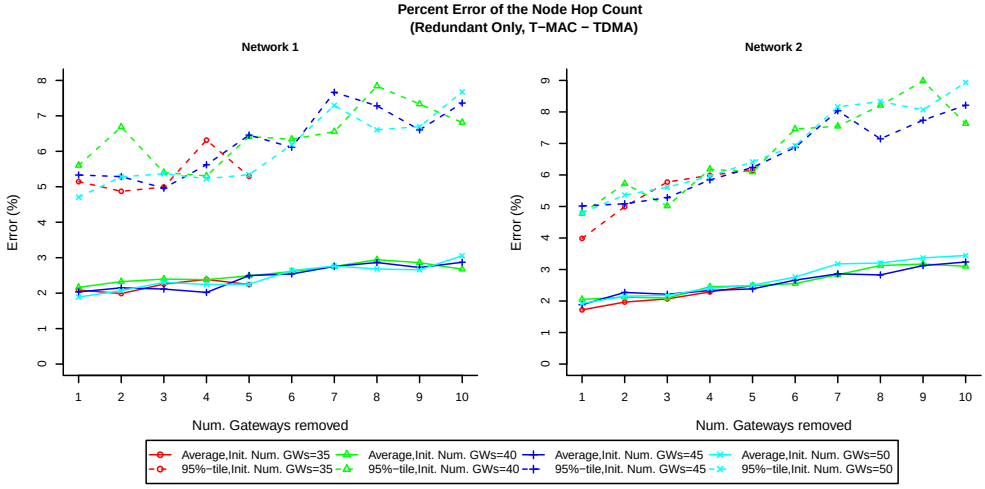


Figure 4.18: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using the T-MAC and TDMA protocols.

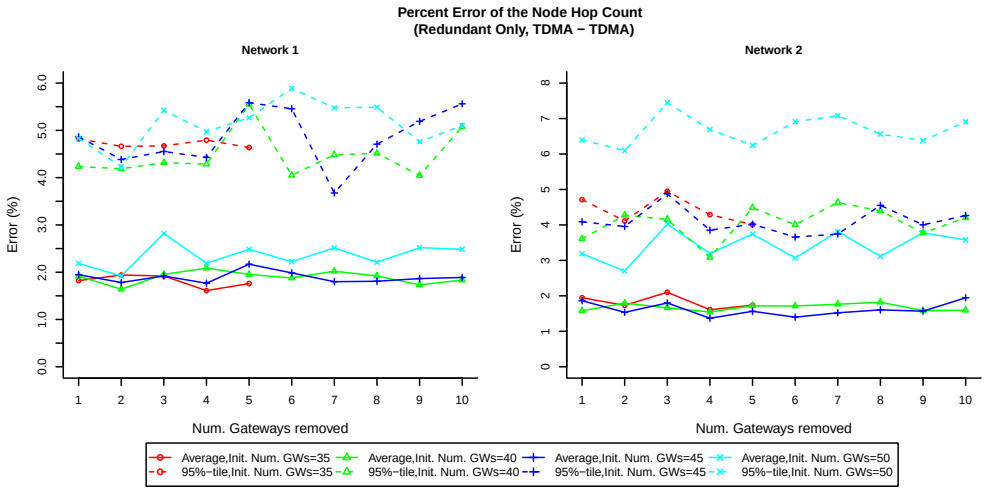


Figure 4.19: Average and 95%-tile of the percent error of the node hop count for the 'Redundant' scenario using two TDMA MAC protocols.

4.3.2.3 Reliability

The prediction error for the reliability metric is shown in Figures 4.20 to 4.26. For most combinations of MAC protocols the prediction error 'behaves' in mostly the same manner (The TDMA - TDMA scenario being a notable exception). As with the other two metrics, the prediction error grows with the number of removed gateways. This is most likely caused by the fact that, although interference between heterogeneous MAC protocols can have a significant impact on the reliability (see chapter 2), the effects of this interference are not considered by the prediction algorithm. Given that the effects of this interference on the link reliability increases with the number of interfering nodes, it is also expected to decrease with the number of removed virtual gateways. Because of this, the difference between the predicted reliability and the actual reliability also increases with the number of virtual gateways that are removed in one step. When the effect of the 'Initial number of gateways' is considered, the prediction error can be seen to grow with the 'Initial number of gateways'. This is in contrast to the node duty cycle and hop count metrics where the prediction error either stays the same or decreases when the initial number of gateways increases. To provide a possible explanation for this behaviour it should first be noted that, as discussed in section 4.3.1, the routing protocols of the network are not bound by the predictions made by the prediction algorithm and that as a result the actual routing topology of the network may differ from the predicted topology even if no paths were broken by the removal of the virtual gateways. Moreover, it should also be noted that the number of inter-network links available to the routing protocols decreases when more and more virtual gateways are disabled. Because of this, the number of alternative inter-network routing paths will also decrease with the 'Initial number of gateways' and as a result the actual route topology is likely to more closely resemble the predicted topology, which in turn results in a lower prediction error.

As with the hop count metric, there is also a significant difference in prediction error

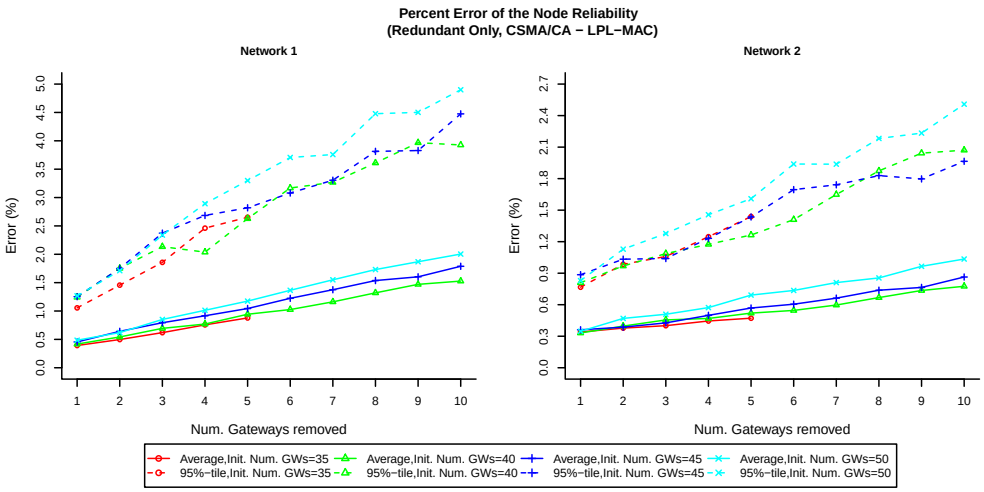


Figure 4.20: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using the CSMA/CA and LPL-MAC protocols.

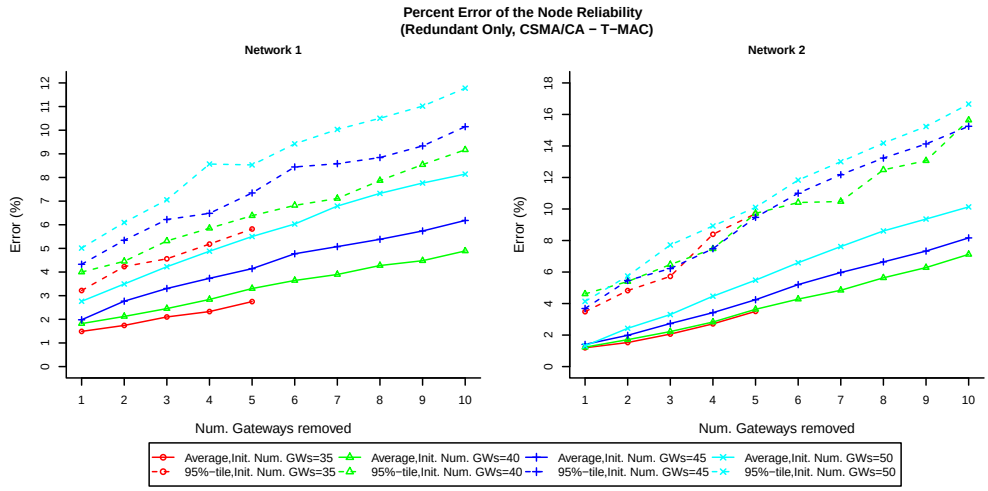


Figure 4.21: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using the CSMA/CA and T-MAC protocols.

depending on whether or not the T-MAC protocol was used. When the T-MAC protocol is *not* used, the prediction error is very low with averages rising to around 2% and 95-percentiles smaller than 5% (The TDMA-TDMA case being the only exception). If one of the networks uses the T-MAC protocol however the average prediction error can reach as high as 10% (with the 95-percentiles topping out at 15%), when a large number of gateways are removed. This large difference in prediction error is most likely caused by the fact that, as discussed in chapter 2 T-MAC has severe scalability issues and that increasing the number of T-MAC nodes in the wireless environment causes the reliability of this MAC protocol to drop. This also means that reducing the number of T-MAC nodes (i.e., by disabling virtual gateways) will have a positive effect on the reliability. Given however that this effect is not directly considered by the prediction algorithm, it is not unexpected for the prediction error to be larger when using the T-MAC protocol than when using other MAC protocols. As with the hop count and the duty cycle metrics however, the prediction error can be significantly reduced by limiting the number of gateways removed in a single step.

As shown in figure 4.26, the behaviour of the prediction error is much more erratic for the TDMA-TDMA scenario than it is for the other combinations of MAC protocols. Despite there being a significant variation in the average and especially the 95-percentile of the prediction error, no clear upward or downward trend with either the initial or the removed number of gateways can be discerned. This behaviour may be explained by the fact that, as discussed in section 4.3.1 for the TDMA - TDMA scenario, the link reliability is much more dependent on MAC protocol specific timings that cannot be predicted by the prediction algorithm. Moreover, the significant variance in prediction error between relatively minor differences in the number of gateways may be caused by the fact that, in the case of TDMA, these unpredictable MAC protocol timings heavily depend on the specific slot allocation used and that the slot allocation in turn depends on the number of nodes synchronising to the TDMA-protocol of a specific network, which in turn depends

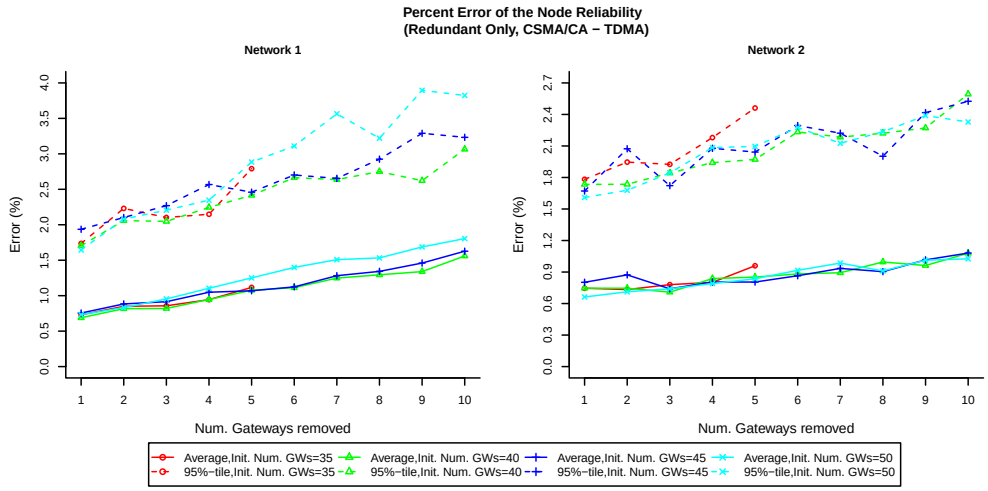


Figure 4.22: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using the CSMA/CA and TDMA protocols.

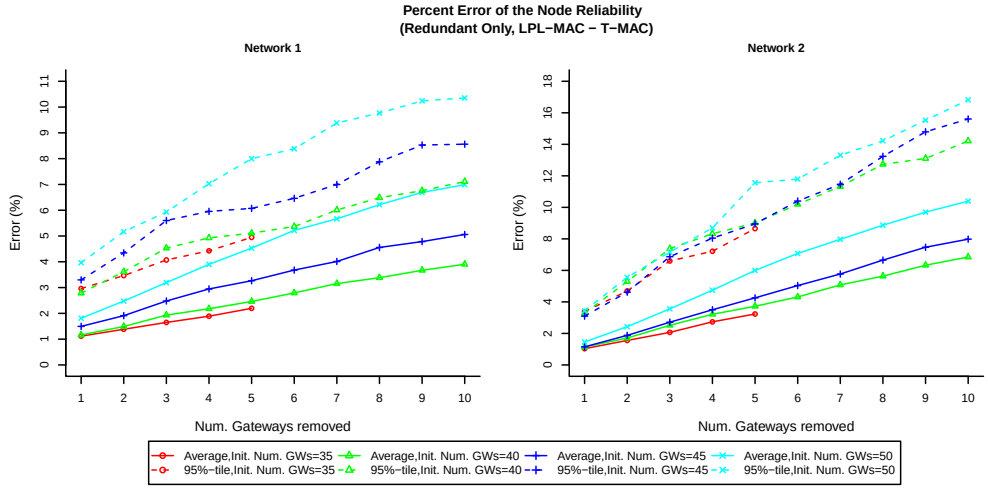


Figure 4.23: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using the LPL-MAC and T-MAC protocols.

on the total number of gateways.

Despite its erratic behaviour, the prediction error is not that different from the one measured in the Non-Border scenario. The average prediction error (max 4.5%) sits squarely between the averages recorded for the Non-Border scenario while 95-percentile error is more or less the same (around 11%).

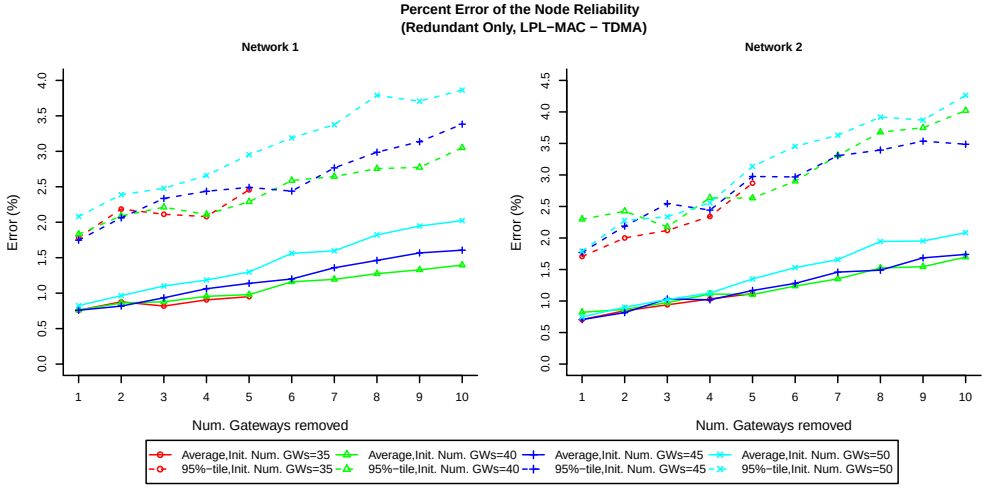


Figure 4.24: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using the LPL-MAC and TDMA protocols.

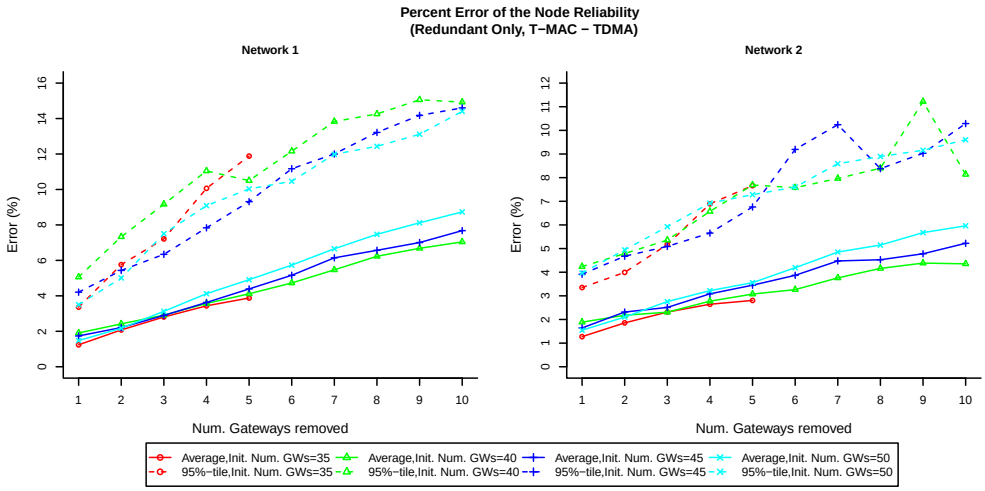


Figure 4.25: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using the T-MAC and TDMA protocols.

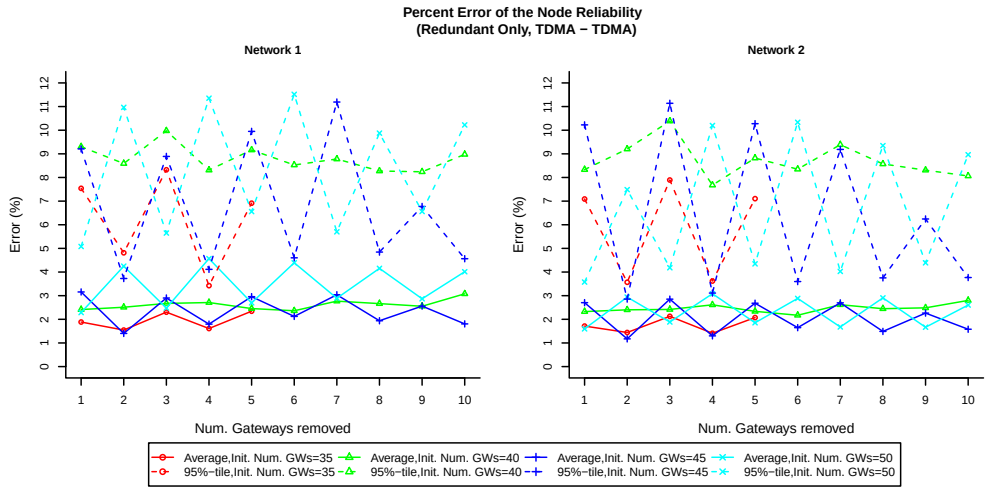


Figure 4.26: Average and 95%-tile of the percent error of the node reliability for the 'Redundant' scenario using two TDMA MAC protocols.

4.3.3 Removal of non-redundant gateways

The accuracy of the prediction error is evaluated in almost exactly the same way for the non-redundant-gateways scenario as it is for the redundant-gateway scenario. The main difference is that, after the *maximum* and *minimum* gateway configuration have been evaluated, all the redundant gateways are removed from the network. The 'initial configurations' are then generated by removing randomly selected gateways from this configuration. As with the redundant-gateways scenario the 'initial number of gateways' and the 'number of gateways removed' are considered as parameters. The 'initial number of gateways' is varied from 21 to 6 gateways in steps of 5 gateways, the 'number of gateways removed' is varied from 1 to 10 gateways. For the case where the 'initial number of gateways' is 6, the 'number of gateways removed' is at most 5 since it is not allowed to remove any of the gateways in the minimal gateway configuration.

4.3.3.1 Duty Cycle

The prediction error for the duty cycle metric is shown in figures 4.27 to 4.33. As for the redundant-gateways scenario, the prediction error rises with the number of removed gateways. In contrast to the redundant-gateways scenario however, the prediction error does not always increase when the initial number of gateways is reduced. In some cases, reducing the initial number of gateways also causes the prediction error to drop, although this is less clear. Moreover, this upward/downward trend is not tied to the use of a specific MAC protocol but is instead dependent on the exact combination of MAC protocols used. When, as shown in figure 4.27, LPL-MAC is combined with CSMA/CA for instance, the prediction error for the LPL-MAC network can clearly be seen to rise when the initial number of gateways is reduced. When LPL-MAC is combined with T-MAC however (see figure 4.30), the inverse happens and the prediction error mostly decreases when there are fewer gateways to start from. This relation between the used MAC protocols, the initial

number of gateways and the prediction error may be explained by the fact that, as far as duty cycle is concerned, removing one or more non-redundant gateways has two main effects on the nodes in the wireless environment. Each of these effects will have a stronger or weaker effect depending on the set of MAC protocols used and while the influence of the first effect is larger for lower numbers of initial gateways, the influence of the second effect on the prediction error rises with the number of initial gateways.

The first effect of removing a non-redundant gateway is that, in contrast to the redundant scenario, a number of routing paths will be broken and that the traffic flowing over these paths needs to be rerouted over alternative paths passing through one of the remaining gateways. Since there is a duty cycle-wise cost associated with the forwarding of traffic, this will also cause the duty cycles of the nodes on these paths to change. Since the prediction algorithm assumes the duty cycle of non-gateway nodes to be unaffected by the removal of a gateway this may cause the prediction error to rise. Moreover, this effect on the prediction error is larger when there are fewer initial gateways because in that case there are fewer 'cross-over' points between the networks and the removal of a single gateway will have a larger impact on the route topology.

The second effect of removing a non-redundant gateway is that, as discussed in section 4.3.2, it indirectly affects the duty cycle of the surrounding nodes. An effect which is also not taken into account by the prediction algorithm. In contrast to the redundant gateway scenario however, a reduction in the number of initial gateways causes the prediction error to decrease rather than increase. This is because the total number of gateways is significantly lower than in the non-redundant scenario (between 1 and 21 instead of between 30 and 50) and that the area in which these gateways are deployed will decline when the number of gateways is reduced. (This is in contrast to the redundant scenario where the gateways are spread evenly to prevent routing paths being broken.) Given that the effects of interference between MAC-heterogeneous sensor networks rise with the area in which these networks overlap (see chapter 2), that this area increases with the initial number of gateways and that none of these effects are taken into account by the prediction algorithm, reducing the number of initial gateways will also have a positive effect on the prediction error.

When the magnitude of the prediction error is examined it quickly becomes clear that, in most cases, the prediction error for the non-redundant scenario is significantly higher overall than for the redundant scenario. More importantly, the prediction error also rises much more rapidly with the number of removed gateways. As expected, the error is always zero for the CSMA/CA MAC protocol. For the LPL-MAC protocol the average error is still relatively small (at most 3.5%), but in all test cases the 95-percentile error rapidly rises to around 9% and can even become as large as 15%. The prediction error is even higher for the T-MAC and TDMA MAC protocols. For the T-MAC protocol the average error can reach as high as 9% while the 95-percentile error can become as high as 21%. For the TDMA MAC protocol, the average error can reach as high as 13% while the 95-percentile error can reach as high as 26%. Although the 'worst case' prediction error can thus become quite high, it should also be noted that in most cases the prediction error is still quite small if only a few virtual gateways are removed at the same time. As for the redundant scenario, the value for the 'number of gateways removed'-parameter will be an important design consideration of the selection algorithm.

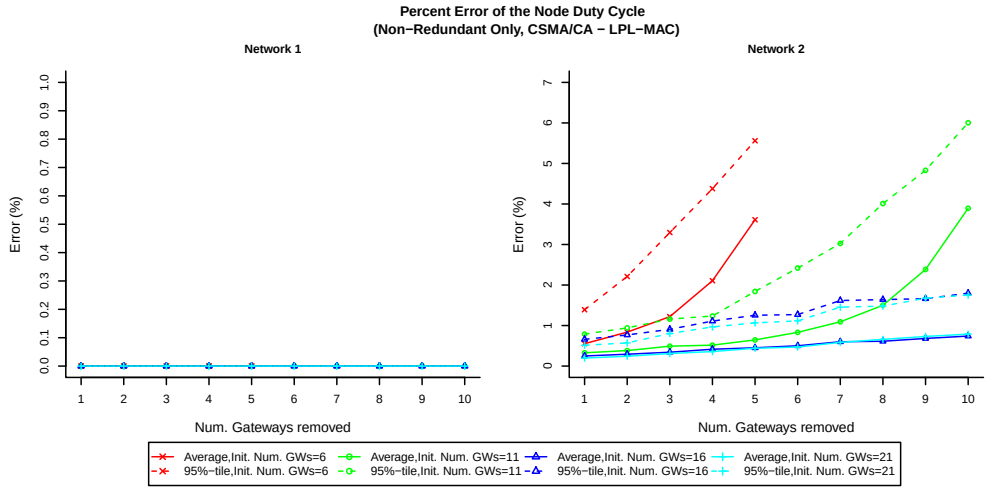


Figure 4.27: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using the CSMA/CA and LPL-MAC protocols.

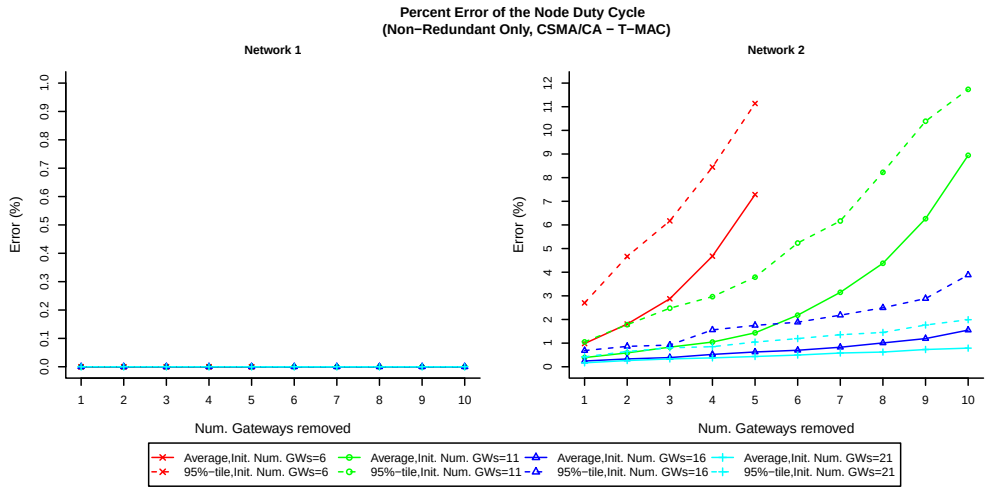


Figure 4.28: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using the CSMA/CA and T-MAC protocols.

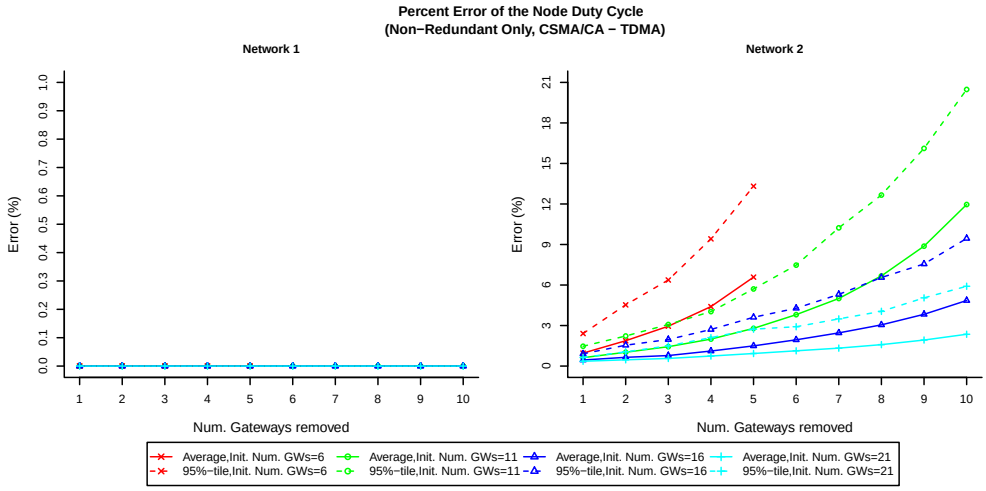


Figure 4.29: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using the CSMA/CA and TDMA MAC protocols.

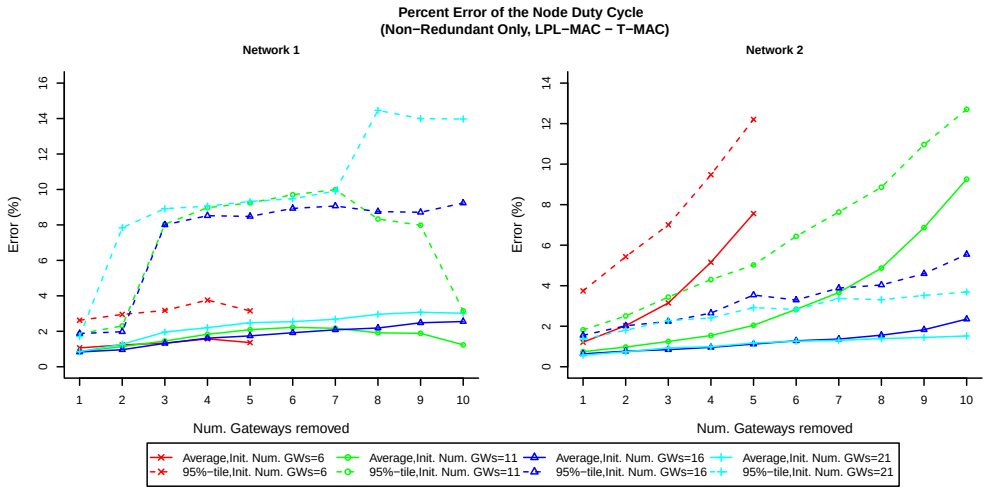


Figure 4.30: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using the LPL-MAC and T-MAC protocols.

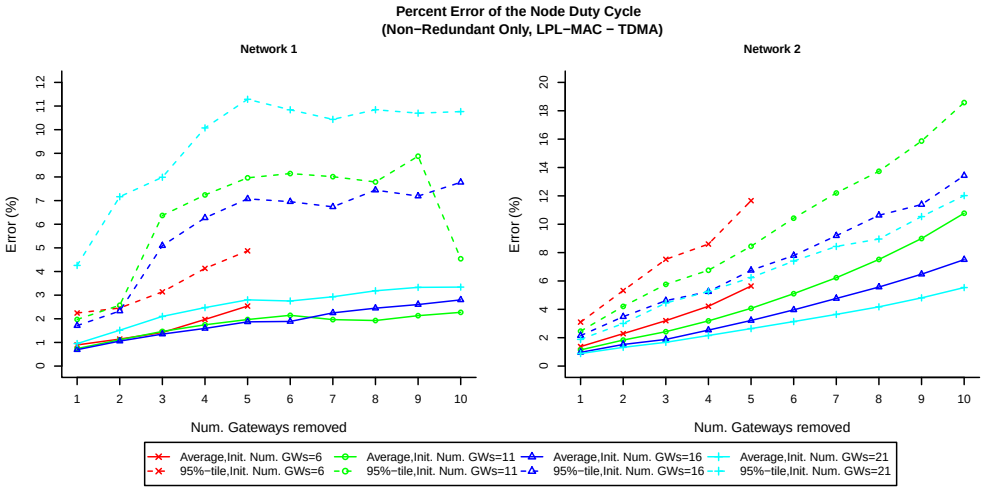


Figure 4.31: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using the LPL-MAC and TDMA MAC protocols.

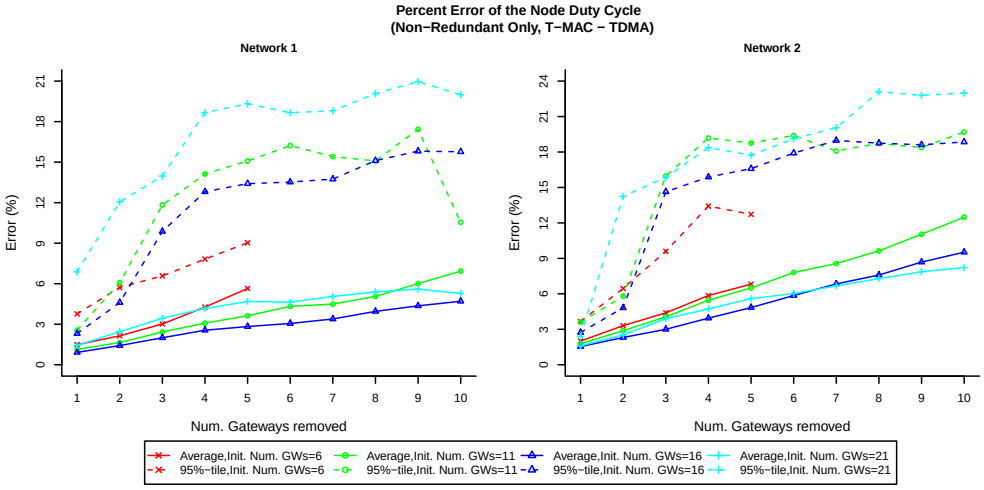


Figure 4.32: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using the T-MAC and TDMA MAC protocols.

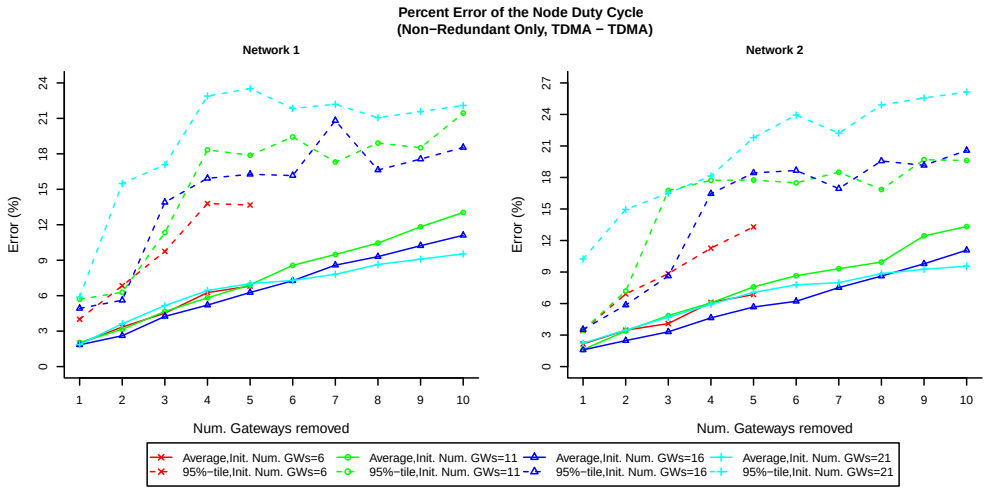


Figure 4.33: Average and 95%-tile of the percent error of the node duty cycle for the 'Non-Redundant' scenario using two TDMA MAC protocols.

4.3.3.2 Hop Count

The prediction error for the hop count metric is shown in Figures 4.34 to 4.40. In contrast to the redundant scenario, the value of the 'initial number of gateways' parameter has a definite impact on the prediction error. When the number of initial gateways is high (16 or 21 gateways), the prediction error can clearly be seen to rise with the number of removed gateways. When the initial number of gateways is reduced further (11 or 6 gateways) this behaviour changes. In that case, the prediction error initially rises with the number of removed gateways, but then drops significantly when the number of removed gateways is increased even further.

This drop in prediction error is caused by the fact that, when a large number of gateways is removed from a 'small' initial configuration, only very few gateways will remain in the *final* configuration (that is: the set of gateways remaining after the gateway have been removed). Since, as discussed in the beginning of section 4.2, every gateway configuration *must* contain the minimal configuration, the final configuration will resemble the minimal configuration more and more as more gateways are removed from the initial configuration. Given that the prediction algorithm uses, amongst others, paths from the minimal configuration to replace paths that were broken by the removal of a virtual gateway (see section 4.2.3), it should come as no surprise that in this case the predicted topology will more closely match the actual topology and that as the prediction error will therefore also be lower. It should be noted though that, although the prediction error drops significantly as more gateways are removed, it does not disappear entirely. Even when 5 gateways are removed from a set of 6 (or 10 gateways are removed from a set of 11) initial gateways and the final configuration thus equals the minimal configuration, a non-zero prediction error can still be observed. This is because the prediction algorithm does not consider the special case where the *final* configuration equals the *minimum* configuration separately. The prediction algorithm therefore predicts the topology for this configuration in exactly the same manner as it is done for any other gateway configuration, despite the fact that

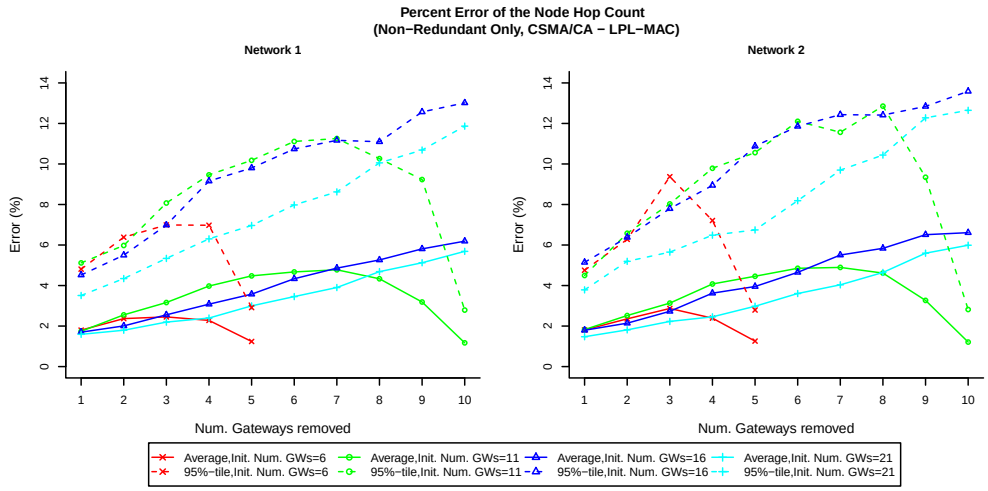


Figure 4.34: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using the CSMA/CA and LPL-MAC protocols.

it could already 'know' what the topology would be. While broken flows are thus, most likely, replaced by paths from the minimal configuration, there is no guarantee that the distribution of the traffic over these paths will be the same as in the minimal configuration. Given that the predicted hop count is calculated as a weighted average over the hop counts of the different predicted paths, this can cause a discrepancy between the predicted and the actual average hop count.

When the magnitude of the prediction error is considered, it can be observed that, as for the duty cycle, the prediction error can become significantly higher than in the redundant gateway scenario. When only a few gateways are removed, the average prediction error is still quite low (between 2% and 4%) but when the number of removed gateways is increased it can, in the worst case become as high as 10%. For the 95-percentile error is already quite high (but still acceptable) when only a few gateways are removed (between 5% and 9%). When the number of removed gateways is increased however, the 95-percentile error rises very rapidly and can reach as high as 23%. This means that, as with the duty cycle metric, the value for the 'number of removed gateways' parameter will need to be carefully chosen to keep the prediction error within acceptable bounds.

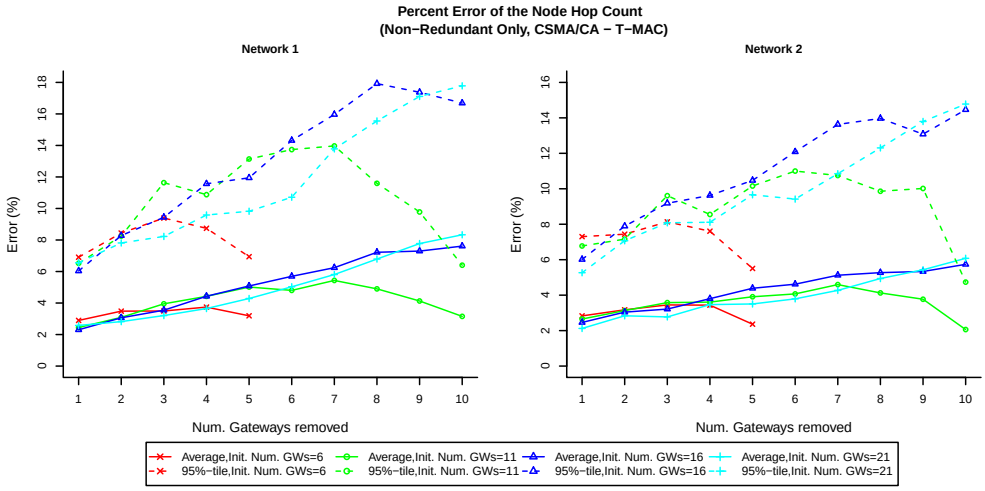


Figure 4.35: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using the CSMA/CA and T-MAC protocols.

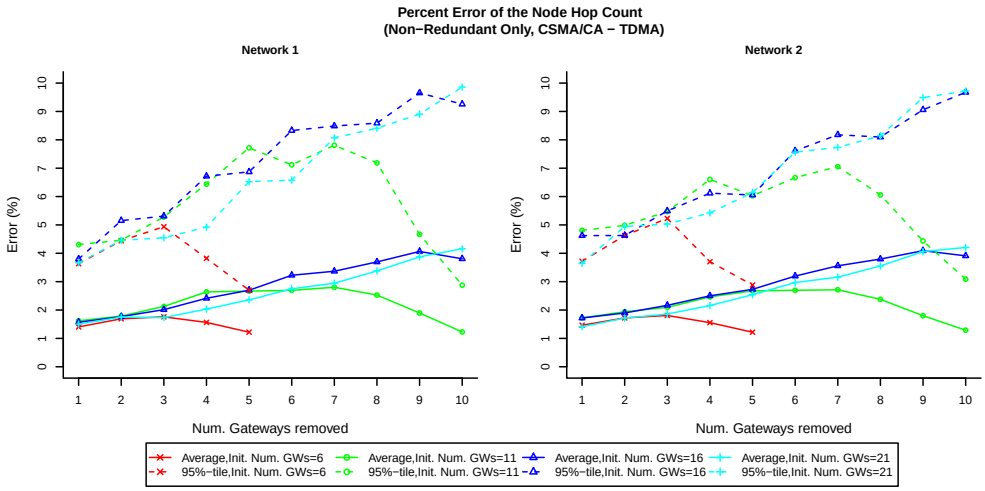


Figure 4.36: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using the CSMA/CA and TDMA protocols.

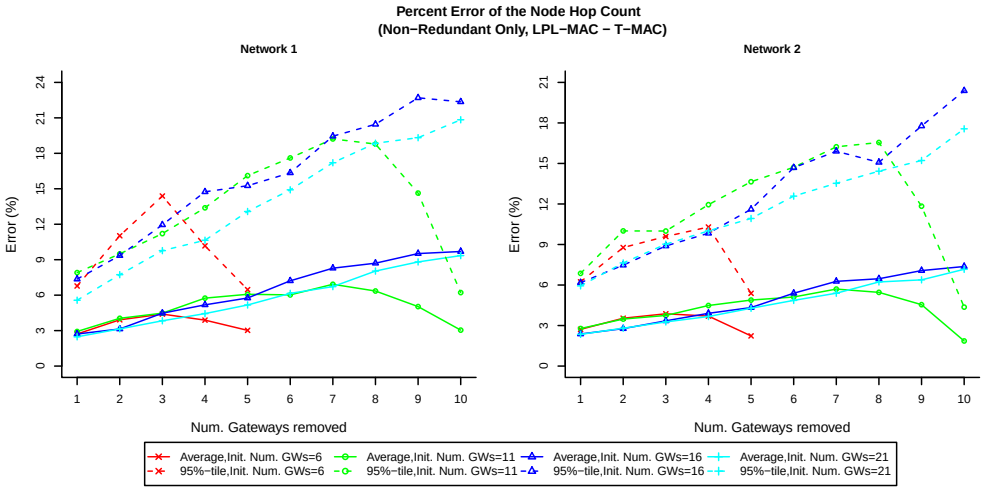


Figure 4.37: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using the LPL-MAC and T-MAC protocols.

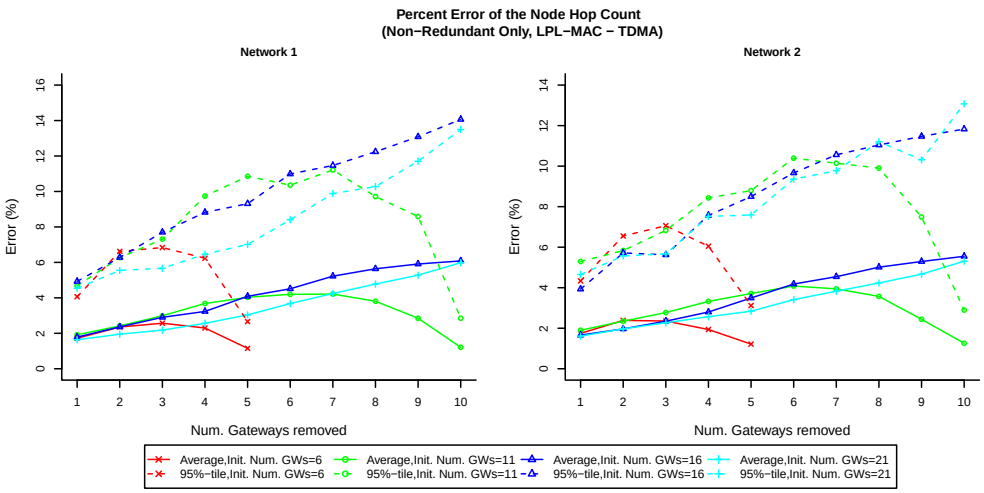


Figure 4.38: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using the LPL-MAC and TDMA protocols.

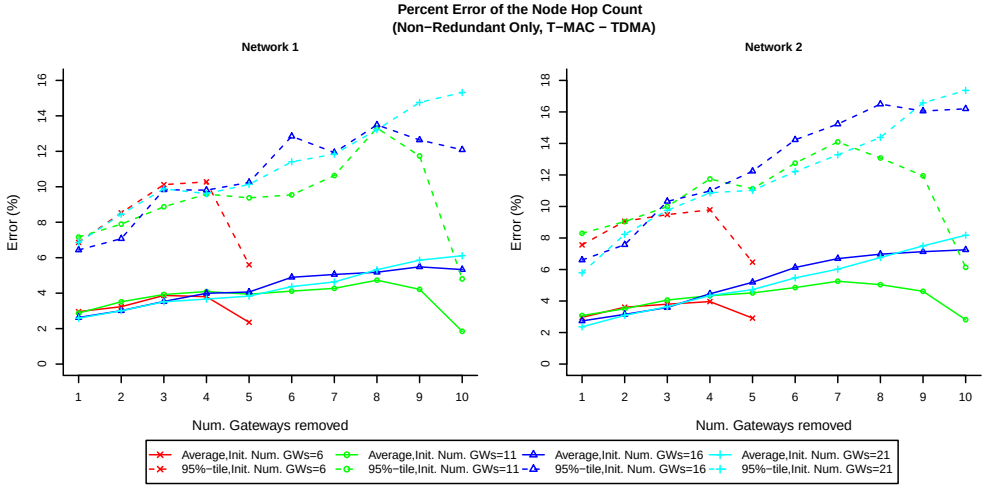


Figure 4.39: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using the T-MAC and TDMA protocols.

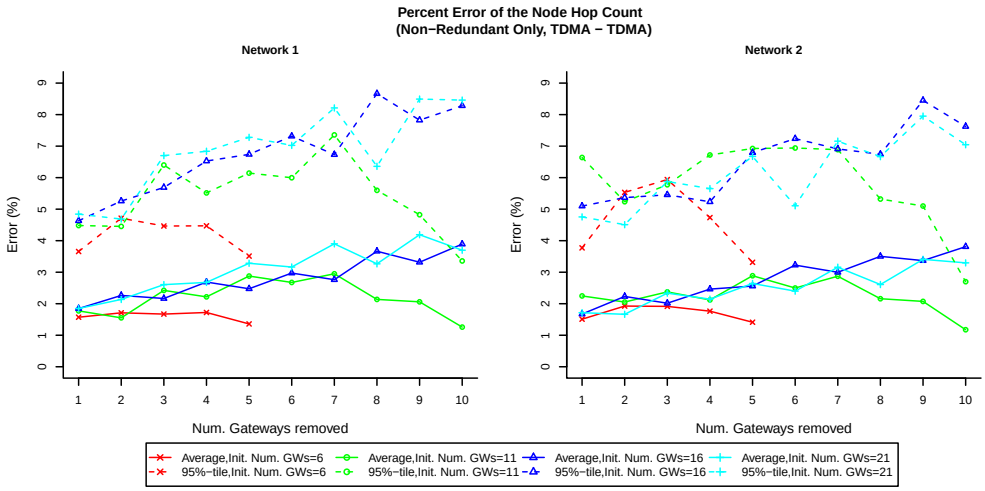


Figure 4.40: Average and 95%-tile of the percent error of the node hop count for the 'Non-Redundant' scenario using two TDMA MAC protocols.

4.3.3.3 Reliability

The prediction error for the reliability metric is shown in Figures 4.41 to 4.47. For most combinations of MAC protocols, the prediction error reacts in more or less the same way to variations in either the number of initial or the number of removed gateways, with the TDMA-TDMA case being a notable exception. As for the redundant scenario, the prediction error rises with the number of removed gateways. When the initial number of gateways is reduced the prediction error also increases. This is in contrast to the redundant scenario where a reduction in the number of initial gateways also causes the prediction error to drop. This different behaviour is most likely caused by the fact that the paths broken by the removal of a non-redundant gateway cannot be repaired by the local repair mechanism (see section 4.2.2). Instead, the path is either removed or, if it is the last remaining path in the flow, replaced by a set of 'plausible' replacement paths. Since, as discussed in section 4.2.3, in this work only a set of relatively 'general' path replacement policies are used and that the routing protocol is not bound by the predictions of the prediction algorithm, this incurs a certain error in the predicted reliability. Moreover, every time the number of initial gateways is reduced, the set of paths broken by the removal of a gateway, as well as the traffic flowing over these paths, becomes proportionally larger and as a result the prediction error also increases. For the TDMA-TDMA case, the behaviour of the prediction error is more erratic. As with the other MAC protocols, a reduction in the number of initial gateways does cause the prediction error to rise, but when the influence of the number of removed gateways is examined it is clear that the average and especially the 95-percentile of the error fluctuate heavily with minor variations in the number of removed gateways. This indicates that, as for the redundant scenario, for the TDMA-TDMA case the slot allocation used is most likely a prominent factor in determining the link reliability.

As with the redundant scenario, there is a significant difference in prediction error depending on whether or not the T-MAC protocol was used (the only exception is the TDMA-TDMA case). When the T-MAC protocol is *not* used, the prediction error is somewhat higher than in the redundant scenario but even then the average prediction error is still quite low (max 4% overall) and although the 95-percentile can reach as high as 10%, it is also significantly lower when only a few gateways are removed at the same time. When one of the MAC protocols used is the T-MAC protocol however, the prediction error is much more substantial. Even when only a few gateways are removed the 95-percentile can reach as high as 9%. Moreover, the error also rises substantially faster and while the average error can reach as high as 16%, the 95-percentile error can reach as high as 30% when more gateways are removed. For the TDMA - TDMA case the average prediction error is a little bit lower (around 10%), the 95-percentile can also reach as high as 30%. Moreover, as is clear from figure 4.47, these enormous prediction errors not only occur when a large number of gateways is removed but also when fewer gateways are removed at the same time.

Although in general the prediction error is thus fairly low or can otherwise be sufficiently reduced by choosing a conservative value for the 'number of gateways removed'-parameter, this is unfortunately not the case for all combinations of MAC protocols (the TDMA-TDMA case being a prime example). While this in itself does not make the predictions unusable for the selection of the virtual gateways, it does mean that, apart from choosing a small enough value for the 'number of gateways removed'-parameter, the selection algo-

rithm will have to incorporate some sort of mechanism to compensate for the substantial prediction errors that will sometimes occur.

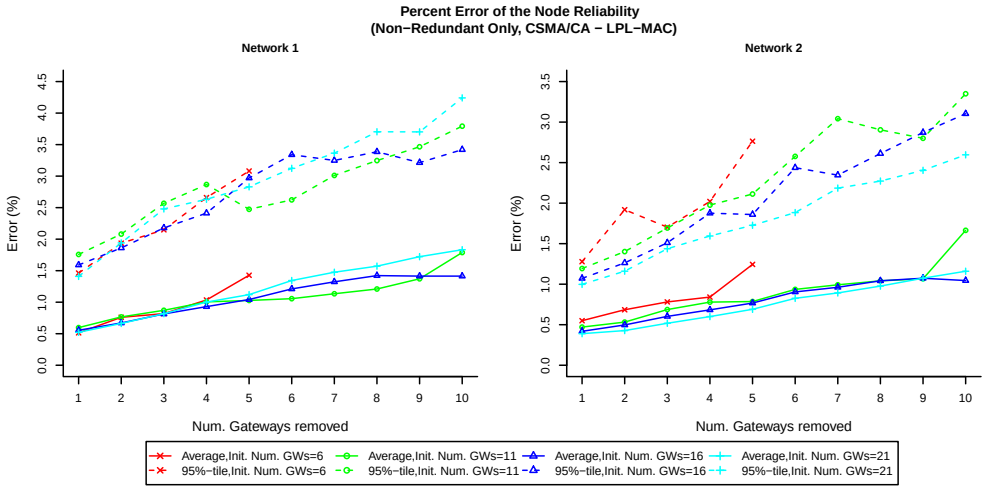


Figure 4.41: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using the CSMA/CA and LPL-MAC protocols.

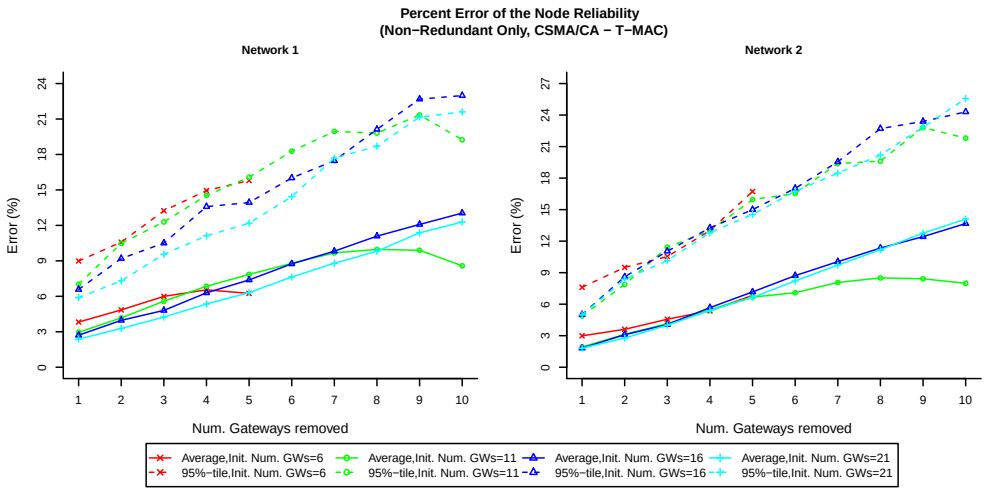


Figure 4.42: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using the CSMA/CA and T-MAC protocols.

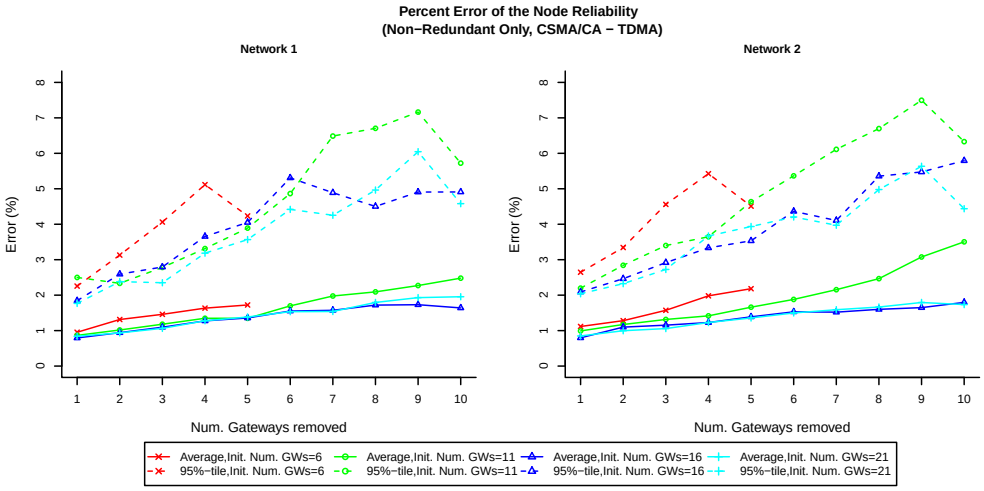


Figure 4.43: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using the CSMA/CA and TDMA protocols.

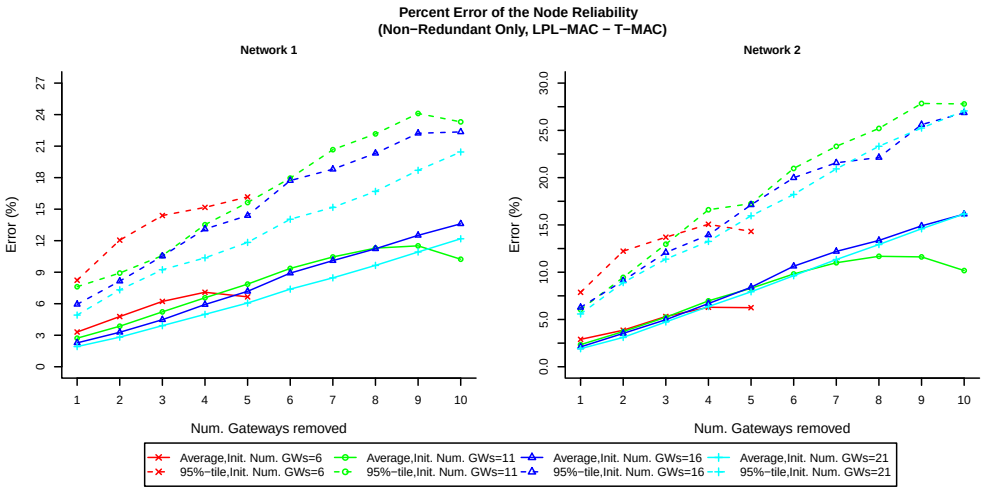


Figure 4.44: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using the LPL-MAC and T-MAC protocols.

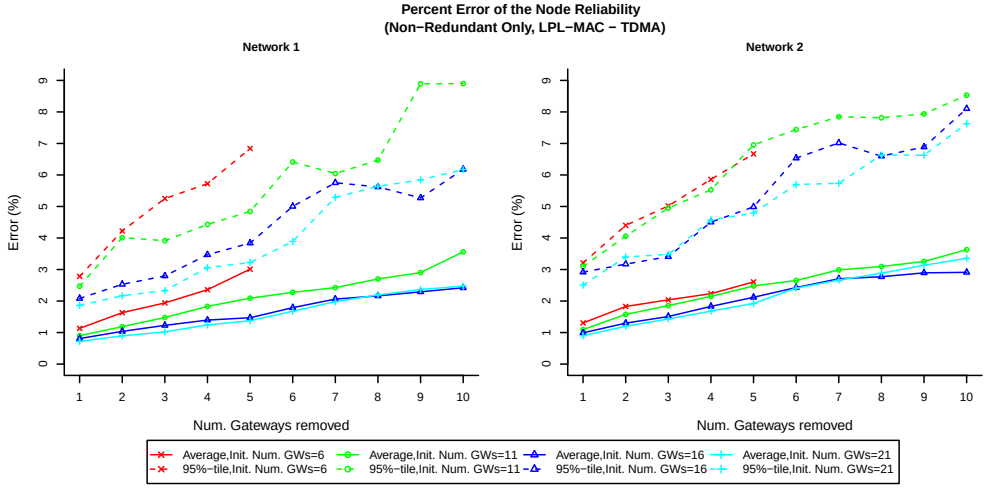


Figure 4.45: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using the LPL-MAC and TDMA protocols.

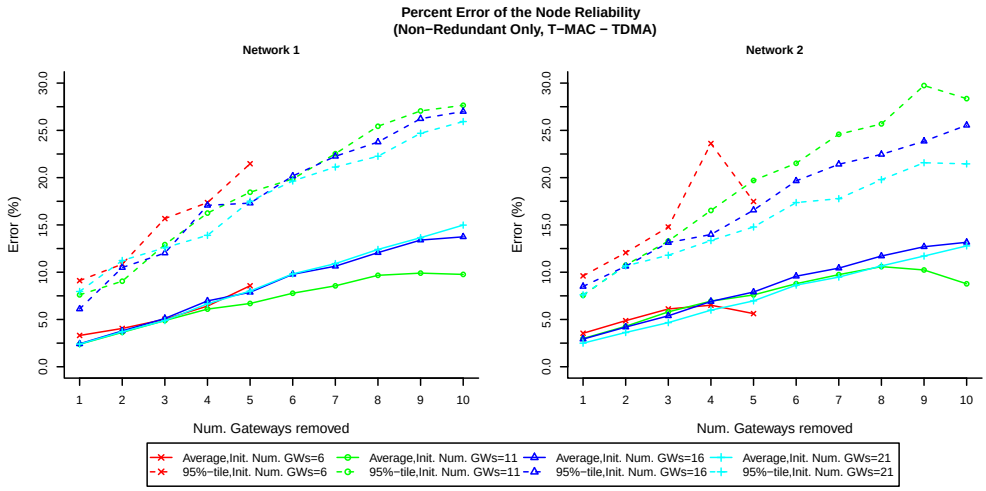


Figure 4.46: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using the T-MAC and TDMA protocols.

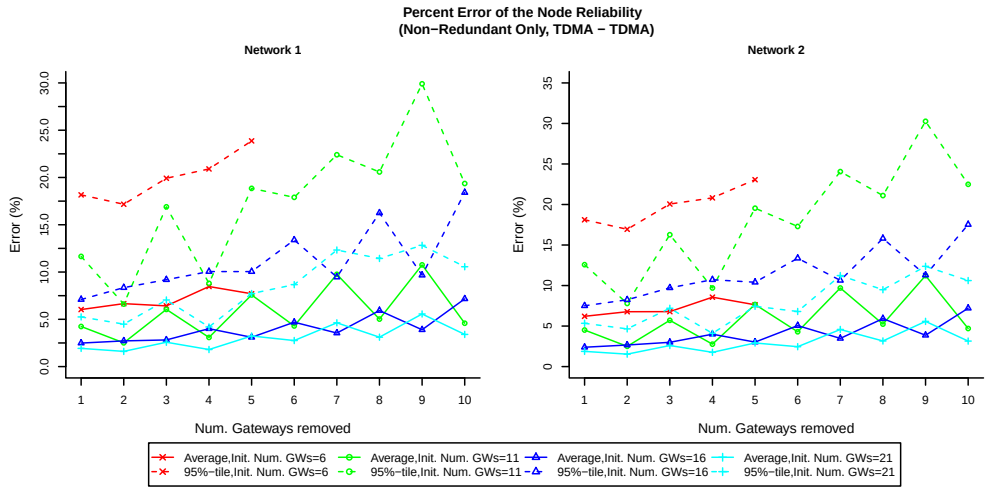


Figure 4.47: Average and 95%-tile of the percent error of the node reliability for the 'Non-Redundant' scenario using two TDMA MAC protocols.

4.4 Conclusion

This chapter introduced IRVG: a heuristic mechanism for selecting the set of virtual gateways to use to enable interoperability between MAC-heterogeneous sensor networks. After discussing the design considerations and providing a general outline of IRVG, this chapter mainly focussed on the prediction algorithm of IRVG. A detailed explanation of this algorithm was provided in section 4.2 after which the prediction algorithm was extensively evaluated (see section 4.3).

From the performance evaluation it is clear that, although for some cases the worst case prediction error can become quite high, for most combinations of MAC protocols the prediction algorithm is able to accurately predict the impact of removing one or more gateways on the duty cycle, node hop count and node reliability of the networks. Overall the prediction error varies with the exact combination of MAC protocols used, with the CSMA/CA - LPL-MAC case being perhaps the most predictable combination and the TDMA-TDMA case being the least predictable. The prediction error also varies depending on the type of virtual gateway that is removed (i.e., non-border, redundant or non-redundant). When only non-border gateways are removed the prediction error varies with the set of MAC protocols used but is generally very low. The only notable exception is the reliability in the TDMA-TDMA case and even then the 95-percentile error is only slightly above the 10% mark.

When only redundant gateways are removed, the accuracy of the prediction algorithm, again, varies with the set of MAC protocols used but the prediction error is also heavily influenced by the number of gateways that are removed in a single step. More specifically, in the redundant scenario, the prediction error is quite low when only a single gateway is removed and then either stays more-or-less the same or rises steadily when the number of removed gateways is increased. The only notable exception is, again, the TDMA-TDMA case where, as discussed in 4.3.2, the prediction error of the node reliability

is more erratic. Despite this, the number of gateways to remove in a single step will almost certainly become an important design/tuning parameter of the gateway selection algorithm as it allows to make a trade-off between the accuracy of the predictions on the one hand and the number of steps required to complete the selection process on the other hand.

For the non-redundant scenario, the prediction error, again, varies with the set of MAC protocols. Moreover, the same relation between the prediction error and the number of removed gateways can be observed, with two important differences: firstly, the prediction error grows much more rapidly and can become significantly higher than in the redundant scenario. Secondly, for a few combinations of MAC protocols the prediction error is already relatively high even when only a single virtual gateway is removed. The prediction error of the node reliability for the TDMA - TDMA case is a prime example of this. This means that, although the number of gateways to remove remains an important design parameter, it will not always be possible to arbitrarily lower the prediction error by limiting the number of removed gateways. Although this does not prevent these predictions from being used by the selection algorithm it does mean that the selection algorithm will have to allow for the fact that, in the non-redundant scenario, the prediction error will occasionally be higher than expected.

CHAPTER 5

Virtual Gateway Selection Part 2: Iterative Gateway Selection

The previous chapter briefly introduced the IRVG (Iterative Removal of Virtual Gateways) mechanism and then discussed the prediction algorithm of IRVG in detail. This chapter discusses the selection algorithm of IRVG in detail and explains how the selection algorithm of IRVG interacts with both the prediction algorithm and the deployed sensor networks to optimise the selection of virtual gateway nodes.

As previously discussed, IRVG uses a relatively straightforward *heuristic* method to determine the set of virtual gateways to use. In the past more advanced machine learning techniques have been successfully applied [151, 152, 153] to a wide range of sensor network research problems, but unfortunately none of these techniques can be applied to the virtual gateway selection problem being investigated here.

Supervised learning techniques, such as decision trees, have for instance been applied to link quality estimation [154] and MAC protocol selection [107], but they are not a good fit for the virtual gateway selection problem since they require a large set of *labelled* training samples which, in this case, are simply not available. Moreover, given that the effects of interference depend largely on the exact node deployment and on the specific MAC protocols used, these training samples would need to be collected from the specific deployment for which a solution is needed. Since doing so would basically come down to 'brute forcing' an acceptable solution, this is not feasible. Genetic algorithms have been used for clustering [155, 156], optimisation of energy efficiency and coverage [157] and routing [158, 21]. Due to the sheer number of candidates that need to be evaluated however, these algorithms can only operate on a *model* of the sensor network. This makes them unusable for the virtual gateway selection problem since, as discussed in section 4.1.2, gateway selection needs to be performed without relying on a model of the sensor network. Neural Network-based solutions have also been proposed for an equally

wide variety of sensor network related problems [159, 160, 161, 162]. These however either keep the number of neurons extremely small ([159, 160]), collect lots of training data to train the neural network beforehand ([161]), or also rely on a model of the sensor network ([162]).

Reinforcement Learning [163] and especially Q-learning would seem to be a better fit since it does not require a model of the network in order to operate and the optimal action for each 'state' of the network can be learned entirely from network measurements. Because of this, it has been applied not only to optimise the different layers of the network stack [79, 108, 164, 165, 166] but also to do high-level network optimisation of already deployed networks [167, 168]. As discussed in [167] however, one of the key concerns when using Q-learning is that the *"state space needs to be kept as small as possible"*. When Q-learning is applied to the problem of virtual gateway selection however, the number of states rises exponentially with the number of candidate virtual gateway nodes since each combination of virtual gateways would need to be considered as a separate state.

Because of this, Q-learning is also not suited for the problem of virtual gateway selection and IRVG instead relies on a heuristic gateway selection strategy. The selection algorithm used is explained further in section 5.1 and is evaluated in section 5.2

5.1 Iterative Removal of Virtual Gateways

The selection algorithm of IRVG determines the set of virtual gateways to use for communication between the two MAC-heterogeneous sensor networks. As previously discussed, this is done by iteratively applying multiple virtual gateway configurations to the network and subsequently collecting both the topology and the network performance resulting from these configurations (see section 4.1.3 for more details about what information is exactly collected). During each *iteration*, the collected performance measurements are used to evaluate the performance of the *current* configuration and the gathered topology information is used (by the prediction algorithm) to make predictions about the performance of possible subsequent gateway configurations.

As discussed in section 4.1.3, this thesis considers three metrics to evaluate the performance of the networks: *node duty cycle*, *node hop count* and *node reliability*. Since different network administrators may value each of these metrics differently, a *reward* function is used to evaluate and compare the performance of different gateway configurations. As will be further discussed in section 5.1.3, this reward function allows the administrators of each network to specify specific *goals* and *weights* to each metric depending on the application scenario and the capabilities of the network.

To minimise the limitations imposed by IRVG on the routing protocols of the networks (see section 4.1.3), the selection algorithm employs a removal-only strategy to select the final gateway configuration. The main idea is to first enable all virtual gateways in both networks and then to iteratively disable virtual gateways until no further performance improvement is possible.

5.1.1 Mathematical Definitions

The mathematical notations defined in section 4.2.1 are also used in this chapter. Given that, in contrast to the prediction algorithm, the selection algorithm does distinguish between nodes of different networks, the following notations are defined to separate the nodes into multiple networks:

NW is the set of all networks. As discussed in section 4.1.2, $|NW| = 2$ within the scope of this work.

$V(i)$ is the set of all nodes in network $i \in NW$ and $NW(v)$ denotes the network of node v. $V(i)$ and $NW(v)$ have the following properties:

$$\bigcup_{i \in NW} V(i) = V$$
$$\forall i, j \in NW : i \neq j \Leftrightarrow V(i) \cap V(j) = \emptyset$$
$$\forall v, w \in V : NW(v) = NW(w) \Leftrightarrow M_d(v) = M_d(w)$$

As discussed in section 4.2.1, $M_d(v)$ denotes the default MAC protocol of node v. These properties, in essence, state that each node is a part of exactly one network, that all nodes in the same network have the same default MAC protocol and that each network has a different default MAC protocol.

5.1.2 Selection Algorithm

As shown in figure 5.1, the selection algorithm consists of three distinct phases: a *preparation* phase, a *redundant removal* phase and a *non-redundant removal* phase.

During the *preparation phase*, the *maximum* and *minimum* gateway configurations are first applied to the networks. As discussed in section 4.2, the routing and network performance information gathered for these two configurations is later used by the prediction algorithm to predict the effect of removing a particular set of virtual gateways from the network. The *maximum* gateway configuration consists of all nodes capable of acting as a virtual gateway, the *minimum* configuration consists of the minimal set of virtual gateways needed to make communication between the two networks feasible. As discussed in section 4.2, the virtual gateways in the *minimum* configuration are never disabled but, depending on the use case, the minimum gateway configuration is allowed to be empty. Finally, the *initial* gateway configuration is applied. This configuration consists of all the gateways in the *maximum* configuration with the *non-border* gateways, that is the gateways without any foreign neighbours, removed. (See section 5.1.4 for a more formal definition).

During the *redundant removal* phase, all *redundant* gateways are removed over the course of several iterations. A set of virtual gateways is said to be redundant if their removal does not break any of the existing routing paths apart from, perhaps, requiring a different MAC protocol being used on some of the inter-network links. (See section 5.1.4 for a more formal definition). To remove these redundant gateways, during each iteration the topology and performance information is first collected for the *current* gateway configuration. Next, the gathered information is examined to determine whether there are any

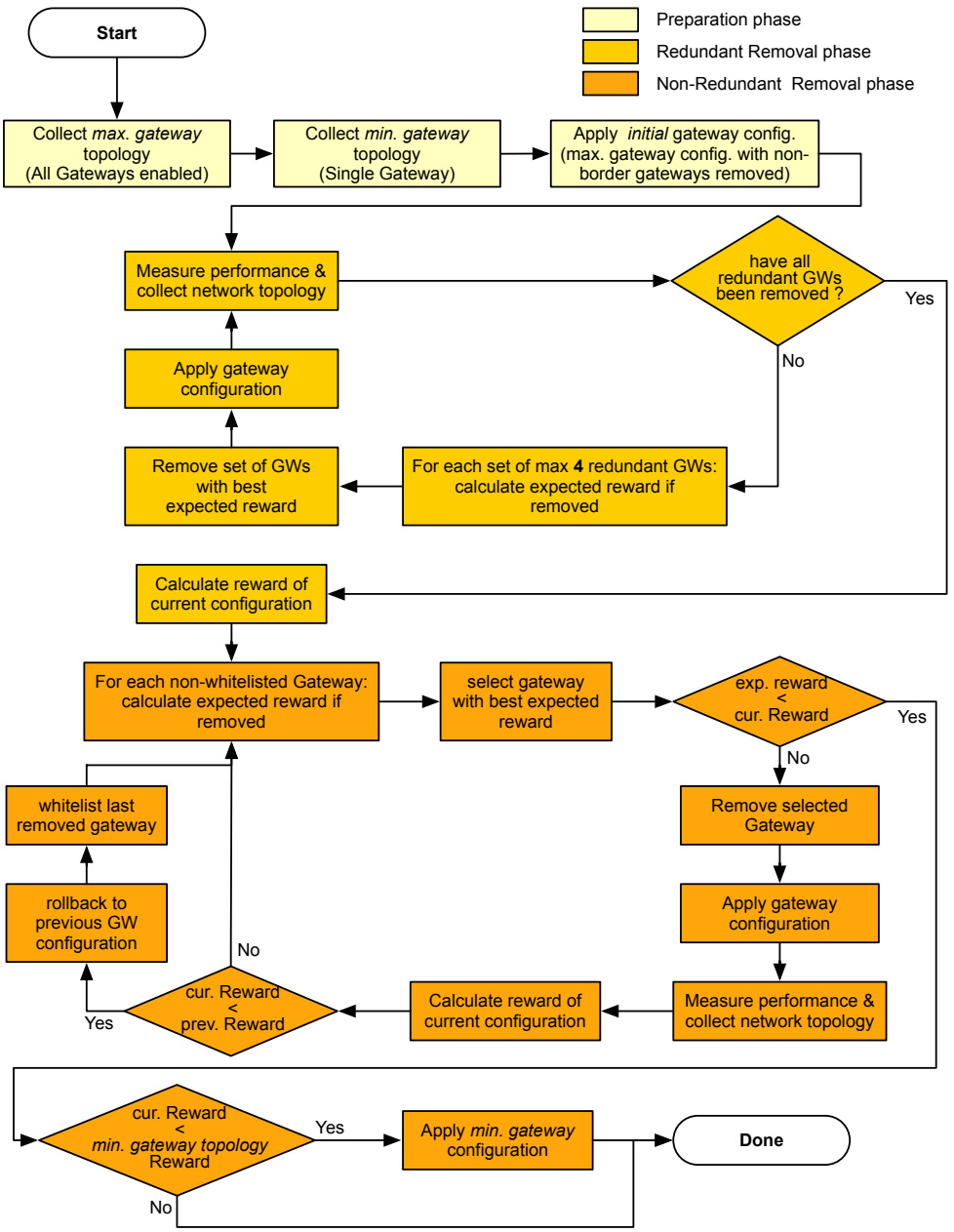


Figure 5.1: Summary of the selection algorithm of IRVG.

more redundant gateways remaining in the current topology. If so, the selection algorithm selects a redundant set of up to four gateways to remove from the current configuration. (The reason for removing redundant gateways four at a time will be explained in section 5.1.5.) In order to do so, first all redundant sets of four gateways are enumerated. If no redundant set of four gateways can be identified, progressively smaller redundant sets are enumerated. For each enumerated redundant set, the expected *reward* resulting from the removal of these gateways is calculated. This is done by first using the prediction algorithm to predict the performance resulting from the removal of these gateways and subsequently calculating the reward based on the goals and weights specified by the network administrators. How this reward is calculated will be explained in section 5.1.3. The final step is to remove the set of redundant gateways with the highest expected reward and to apply the new configuration to the networks. Once all redundant gateways have thus been removed, the reward of the latest gateway configuration is calculated after which the selection algorithm enters the non-redundant removal phase.

In the *non-redundant removal* phase the number of virtual gateways is further reduced by removing or whitelisting the remaining (non-redundant) virtual gateways. As with the previous phases, this is done in multiple iterations. Each iteration starts by first calculating, for every non-whitelisted gateway in the current configuration, the expected reward resulting from the removal of that gateway. As with the redundant removal phase, this is done by first using the prediction algorithm to predict the network performance and subsequently calculating the reward. The virtual gateway with the highest expected reward (if removed) is then selected for removal. If the expected reward of removing the selected virtual gateway is higher than the *current* reward (that is: the reward of the current gateway configuration), the gateway is removed from the current configuration and the new gateway configuration is applied in the networks. Next, the performance measurements and network topology of the networks is collected after which the *actual* reward of the virtual gateway configuration is calculated. Before selecting a new gateway to remove, this actual reward is compared with the previous reward. If it is smaller than the previous reward, the last removed gateway is whitelisted since removing it had a negative effect on the overall network performance. This process of selecting and removing gateways is repeated until either there are no more (non-whitelisted) gateways left to remove or until the expected reward of the new configuration is smaller than the reward of the current configuration.

The final step in the selection algorithm is to compare the reward of the *selected* gateway configuration, that is: the set of gateways remaining after the above steps have been performed, with the reward of the *minimum* configuration. In the event that the reward of the minimum configuration is larger than that of the *selected* configuration, the *minimum* configuration is adopted as the *final* configuration after which this configuration is applied to the networks. Otherwise, the *selected* configuration is adopted as the final configuration. This step is needed because the selection algorithm uses a greedy strategy to determine the set of gateways to use which may cause a local rather than a global optimum to be selected. This in turn may cause the *selected* gateway configuration to have a smaller reward than the *minimum* configuration in cases where the reward is mostly determined by the *number* of virtual gateways being used and less by the location of these gateways. This will for instance be the case when the network administrators specify an extremely high weight to the node duty cycle metric in comparison to the weights for the

other metrics.

5.1.3 Reward Calculation

As part of the virtual gateway selection process, the selection algorithm needs to be able to compare the network performance resulting from different gateway configurations. This is done using a reward function to allow the network administrators to tune the gateway selection mechanism to the application requirements of the individual networks.

To calculate the reward, the obtained performance measurements are first combined into network averages. Section 4.2.4 discusses how this is done when the performance of a gateway configuration is predicted by the prediction algorithm. When the performance measurements are directly collected from the deployed network, the following formulas are used instead:

$$avgDC_x(i) = \frac{\sum_{v \in V(i)} DC_x(v)}{|V(i)|}$$

$$avgR_x(i) = \frac{\sum\limits_{p \in PATHS_x} R_{path_x}(p) TX_x(p,i)}{\sum\limits_{p \in PATHS_x} TX_x(p,i)}$$

$$avgHC_x(i) = \frac{\sum\limits_{p \in PATHS_x} HC_{path_x}(p) TX_x(p,i)}{\sum\limits_{p \in PATHS_x} TX_x(p,i)}$$

In the above formulas, $avgDC_x(i)$, $avgR_x(i)$, $avgHC_x(i)$ are the average node duty cycle, node reliability and node hop count for network i when gateway configuration x is used. For simplicity's sake the x subscript is omitted in the rest of this section since the reward is always calculated using the network averages of one and the same gateway configuration. The above formulas are identical to the ones specified in section 4.2.4 with one exception: when calculating the average node reliability $avgR_x(i)$, there is no $TX_{broken}(i)$ term in the denominator. As discussed in section 4.2.4, that term is only used to account for any traffic for which the prediction algorithm was not able to predict a plausible path. Since the performance and topology information is collected directly from the networks instead of being supplied by the prediction algorithm, this term is removed.

Once the network averages have been calculated, the average node duty cycle and average node hop count need to be 'inverted'. This is because the reward function defined here causes the selection algorithm to optimise for higher values of the considered metrics. Given that for both node hop count and node duty cycle lower values are preferred, the actual values that are inserted into the reward function are calculated as follows:

$$M_{\mathrm{DC},i} = 1\text{-}avgDC(i)$$
$$M_{\mathrm{HC},i} = networkDiameter\text{-}avgHC(i)$$
$$M_{\mathrm{R},i} = avgR(i)$$

In the above equations $M_{m,i}$ is the performance value for metric m in network i that will be used to calculate the reward. As discussed above, only the duty cycle and hop count are 'inverted', the reliability is used as is.

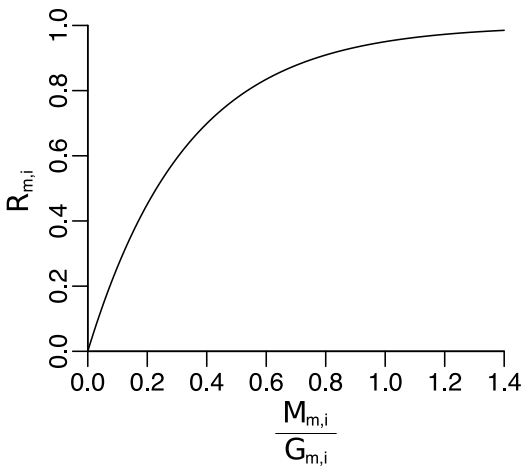


Figure 5.2: $R_{m,i}$ as a function of $\frac{M_{m,i}}{G_{m,i}}$.

To calculate the reward, these values are then normalised based on explicit performance *goals* set by the network administrator. As with the performance metrics, the goals for the duty cycle and hop count metric need to be 'inverted' before being used:

$$G_{\mathrm{DC},i} = 1\text{-}goalDC(i)$$
$$G_{\mathrm{HC},i} = networkDiameter\text{-}goalHC(i)$$
$$G_{\mathrm{R},i} = goalR(i)$$

In the above equations, $goalDC(i)$, $goalHC(i)$ and $goalR(i)$ are the performance goals of network i as set by the network administrator for each of the respective metrics. $G_{m,i}$ is the 'performance goal' value for metric m in network i that will be used to calculate the reward.

The *normalised* performance values are then calculated as follows using the formula proposed by [167].

$$R_{m,i} = 1\text{-}e^{-\alpha \frac{M_{m,i}}{G_{m,i}}} \tag{5.1}$$

In the above formula $R_{m,i}$ is the normalised value for that metric. Performance measurements are normalised exponentially rather than linearly. As illustrated by figure 5.2, this causes $R_{m,i}$ to rise more slowly as $M_{m,i}$ nears and subsequently exceeds $G_{m,i}$, thus ensuring that the influence of a metric tapers off quickly once it has reached its goal. This in turn encourages other metrics to be optimised once the goal has been reached. At the same time the normalised value does continue to slowly rise even after the specified goal has been reached, which allows the metric to be optimised further in the event that no more improvement in any of the other metrics is possible. The α parameter can be used to tune the slope of the normalisation curve but in this work the value proposed by [167] ($\alpha = 3$) is used.

Once the measurements have been normalised, the 'per network' reward (R_i) is calculated

as follows:

$$R_i = \sum_m w_{m,i} R_{m,i} \tag{5.2}$$

$w_{m,i}$ is the *weight* assigned to metric m for network i. These weights allow the network administrator to tune the importance of the different metrics in the reward calculation to the requirements of the network. It should be noted that $\forall m, i : w_{m,i} \in [0, 1]$ and $\forall i : \sum_m w_{m,i} = 1$. Finally, the total reward for the entire configuration (R_{tot}) is calculated from the rewards for the individual networks:

$$R_{tot} = \beta \frac{R_1 + R_2}{2} + (1\text{-}\beta)\, min\,(R_1, R_2)$$

In the above formula, R_1 and R_2 are the individual rewards for networks 1 and 2, calculated according to (5.2). The first term of the above formula maximises the overall reward in both networks while the second term discourages gateway configurations that favour one network over the other. The tradeoff between these two goals can be set by tuning the β parameter. In this work the value $\beta = 0.5$ is used.

5.1.4 Types of Virtual Gateways

When removing virtual gateways, the selection algorithm differentiates between three types of gateways: non-border, redundant and non-redundant gateways. These three types of gateways differ in how important they are to the route topology established by the routing protocols of the networks and, subsequently, how difficult it is for the prediction algorithm to predict the effect of their removal on the route topology of the network. The difference between these types of gateways has already been (briefly) discussed in sections 4.3 and 5.1.2. This section provides formal definitions for each of these three types of gateways.

Non-Border Gateways

Non-border gateways are gateways that are, as the name implies, not located on the 'border' of a network in the sense that they do not have any *foreign* neighbours (neighbours that are not in the same network as the node itself).

Non-border gateways are formally defined as follows:

Let $N_x^L(v)$ and $N_x^R(v)$ denote the *neighbours* of node $v \in V$ in respectively the link graph $LINK_x$ and route graph $ROUTE_x$ for a specific gateway configuration x:

$$N_x^L(v) = \left\{ w \in V \,|\, \exists\, m \in M : v \xrightarrow{m} w \in E_x^L \vee w \xrightarrow{m} v \in E_x^L \right\}$$
$$N_x^R(v) = \left\{ w \in V \,|\, \exists\, m \in M : v \xrightarrow{m} w \in E_x^R \vee w \xrightarrow{m} v \in E_x^R \right\}$$

In that case $NonBorder_x$, the set of non-border gateways for gateway configuration x, is defined as:

$$NonBorder_x = \left\{ v \in GW_x \,|\, \nexists\, w \in N_x^R(v) : NW(v) \neq NW(w) \right\}$$

That is: $NonBorder_x$ is the set of all virtual gateways that don't have any 'foreign' neighbours in the route graph and that are therefore not located on the 'border' of a network.

Using the above definitions, the *initial* gateway configuration discussed in section 5.1.2 can now be formally defined as follows:

$$GW_{init} = GW_{max} \setminus NonBorder_{max}$$

Redundant Gateways

A set of virtual gateways is said to be *redundant* if disabling this set of gateways does not break any of the existing routing paths apart from, perhaps, requiring a different MAC protocol to be used on some of the inter-network links. This is explained by the examples shown in figure 5.3. In figure 5.3.A, both {2, 3} and {4, 5} are redundant sets of virtual gateways. If gateways {2, 3} are disabled these nodes will only be able to use the default MAC protocol of their network and as a result the 'red' links between nodes {2, 3} and nodes {4,5} will become unavailable. Since node {4, 5} are also virtual gateways, the original routing paths can be preserved by switching to the 'green' MAC protocol for communication between the networks. Similarly, if {4, 5} is disabled instead of {2, 3}, the 'green' links will become unavailable and the 'red' links can be used instead. It should be noted that in this example, only one of the sets {2,3} or {4,5} can be removed since, once one set of gateways is removed, the remaining virtual gateways become essential to maintain the existing routing paths between the network. In figure 5.3.B, node 5 is no longer a virtual gateway and as a result of this {2, 3} is no longer a redundant set of virtual gateways. The only remaining set of virtual gateways is the singleton {4}.

Redundant gateways are formally defined as follows:

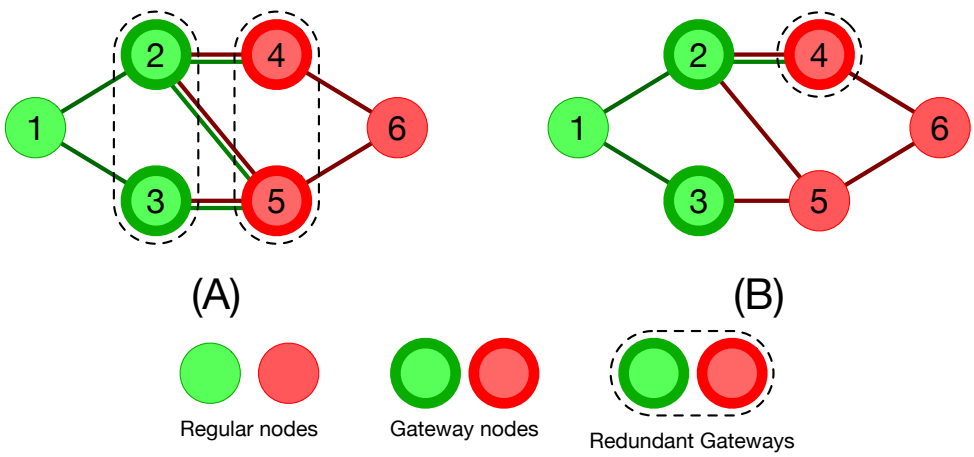


Figure 5.3: Examples of redundant and non-redundant sets of virtual gateways. In (A) sets {2,3} and {4,5} are redundant sets of virtual gateways. In (B) {4} is the only redundant set of virtual gateways as gateways 2 and 3 are needed to maintain connectivity with node 5.

Let $BORDER_x = \left(V_x^B, E_x^B\right)$ be the border-graph for gateway configuration x. It consists of all nodes that are located on the border of a network and the *inter-network* links between them. It is constructed from the route-graph $ROUTE_x$ as follows:

$$V_x^B = V \setminus NonBorder_x$$
$$E_x^B = \left\{ v \xrightarrow{m} w \in E_x^R \,|\, v, w \in V_x^B, NW(v) \neq NW(w) \right\}$$

In accordance with the notations above, $N_x^B(v)$ denotes the neighbour-set of node $v \in V_x^B$:

$$N_x^B(v) = \left\{ w \in V_x^B \,|\, \exists\, m \in M : v \xrightarrow{m} w \in E_x^B \vee w \xrightarrow{m} v \in E_x^B \right\}$$

A set of virtual gateway nodes $S \subset GW_x$ is then defined to be redundant if the following three conditions are met:

$$\forall v \in S \setminus NonBorder_x : N_x^B(v) \subset GW_x \tag{5.3}$$

$$\forall v \in S \setminus NonBorder_x : N_x^B(v) \cap S = \emptyset \tag{5.4}$$

$$\forall v \in S \setminus NonBorder_x : \begin{cases} \forall v \xrightarrow{m_1} w \in E_x^B : m_1 = m_d(v) \vee \exists\, m_2 \in M : \exists v \xrightarrow{m_2} w \in E_x^L \\ \forall w \xrightarrow{m_1} v \in E_x^B : m_1 = m_d(v) \vee \exists\, m_2 \in M : \exists w \xrightarrow{m_2} v \in E_x^L \end{cases} \tag{5.5}$$

Condition (5.3) ensures that all the foreign neighbours of the nodes in S are also virtual gateways. This requirement, for instance, prevents $\{2, 3\}$ in figure 5.3.B from being marked as a redundant set since both nodes have (non-gateway) node 5 as a foreign neighbour. Condition (5.4) ensures that a redundant set of gateways does not contain nodes that are foreign neighbours of one-another. This requirement prevents for instance $\{2,4\}$ in figure 5.3.B from being marked as a redundant set. Finally, condition (5.5) ensures that if a link in the *border-graph* is broken by the removal of a redundant set of gateways, an alternative link (using a different MAC protocol) is available in the *link-graph* to replace it.

It should be noted that (5.3), (5.4) and (5.5) specifically exclude nodes that are not located on the border of the network. This, in essence, means that any redundant set of virtual gateways can contain an arbitrary number of non-border gateways since these gateways do not have any foreign neighbours.

Non-redundant Gateways

Non-redundant gateways are gateway nodes that don't meet the criteria of either non-border of redundant gateways. In essence, these are (mostly) virtual gateways that have

one or more foreign neighbours at least one of which is *not* a virtual gateway. More formally, $NonRedundant_x$, the set of non-redundant gateways for gateway configuration x, is defined as:

$$NonRedundant_x = \left\{ v \in V_x^B \mid \{v\} \text{ is not a redundant set of virtual gateways} \right\}$$

5.1.5 Managing prediction inaccuracies

One of the main conclusions of the previous chapter is that, although the prediction algorithm is generally capable of making fairly accurate predictions, care must none-the-less be taken in how the prediction algorithm is used in order to keep the prediction error within bounds. As discussed in section 4.4, the prediction error is mostly determined by (1) the specific *type* of gateways being removed (i.e., *non-border*, *redundant* or *non-redundant*) and (2) by the *number* of gateways that are removed in a single iteration. Moreover, there are some cases in which a high prediction error cannot be avoided, for instance when non-redundant gateways are removed from an environment where both networks use a pure TDMA MAC protocol (see section 4.3.3).

In order to cope with these limitations, the selection algorithm operates in three different phases. By removing a different type of gateway in each phase, the number of gateways removed per iteration (or 'step size') can be varied depending on the type of gateway removed. This allows the number of iterations needed to complete the selection process to be minimised as much as possible while still keeping the prediction error under control. Given that the prediction error also varies between MAC protocols, one might consider varying the step size with the MAC protocols used, as well as with the type of gateway being removed. Doing so however, would require MAC-specific information to be fed into the selection algorithm which in turn would violate the design constraint that the IRVG-mechanism should be independent from the MAC protocols used.

The removal strategy used by the selection algorithm has been chosen based on the results discussed in the previous chapter. When removing non-border gateways, the prediction error is generally very low, even when all the non-border gateways are removed in a single iteration. Even in the worst case scenario (node reliability for the TDMA-TDMA case), the prediction error can still be considered to be more-or-less within acceptable bounds. Because of this, all non-border gateways are removed in a single step at the end of the preparation phase. Moreover, the error-bounds to be used as a *guideline* in choosing the removal strategy for the other types of virtual gateways, are also chosen based on this 'worst case' non-border prediction error. More specifically, these removal strategies try to keep the average and 95-percentile error below 5% and 10% respectively. This equates to the mean of the average and 95-percentile errors measured for the individual networks in the worst case scenario non-border scenario.

In the redundant-removal phase, virtual gateways are removed four at a time, while in the non-redundant removal phase, at most one gateway is removed in a single iteration. The prediction errors observed for these chosen 'step sizes' are summarised in figures 5.4 to 5.6. In all these figures, the values shown are the *maximum* average, 5%-tile & 95%-tile errors encountered over the different values for the 'initial number of gateways' parameter (see section 4.3). For the redundant removal phase, the chosen step size never results in an average prediction error of more than 5%. Moreover the 95-percentile error is lower

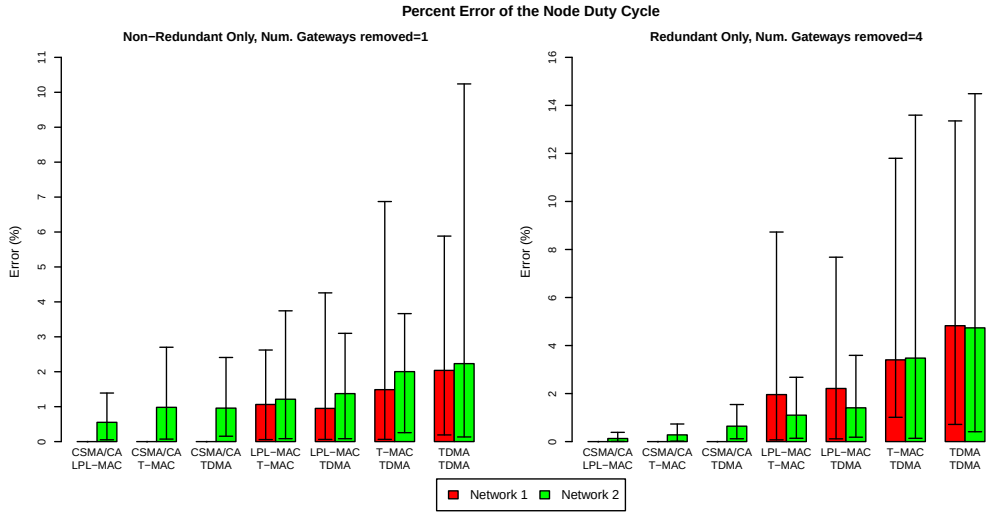


Figure 5.4: Maximum average, 5%-tile & 95%-tile error encountered for the node duty cycle when either removing 1 non-redundant gateway or 4 redundant gateways.

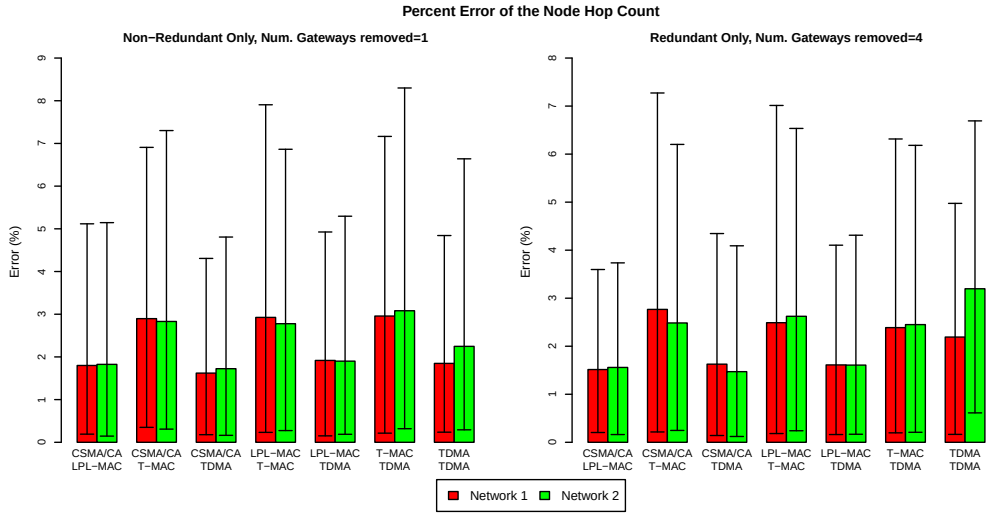


Figure 5.5: Maximum average, 5%-tile & 95%-tile error encountered for the node hop count when either removing 1 non-redundant gateway or 4 redundant gateways.

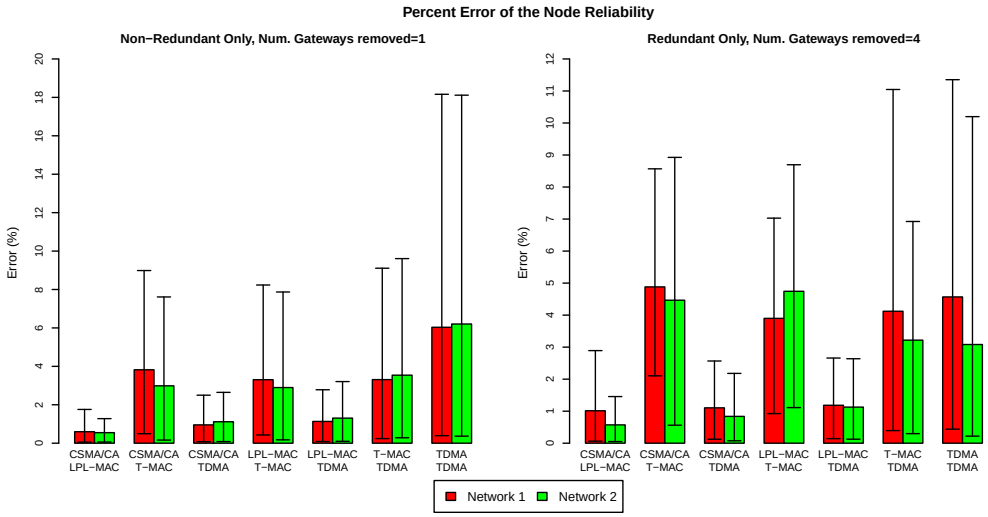


Figure 5.6: Maximum average, 5%-tile & 95%-tile error encountered for the node reliability when either removing 1 non-redundant gateway or 4 redundant gateways.

than 10%, except when the TDMA MAC protocol is used. As discussed in section 4.3.2, this MAC protocol behaves somewhat more 'erratic' than the other MAC protocols and while the 95%-error can indeed reach almost 14% for the node duty cycle metric, it is only around 11% for the node reliability metric and is well below the 10%-mark for the node hop count metric. Moreover, to reduce the 95-percentile error of the node duty cycle to below 10% would require a step size of 1 to be used, which would dramatically increase the number of iterations required for the selection algorithm to complete. Given that for all other combinations of MAC protocols the 95-percentile error is below the 10%-boundary, a step size of 4 is used in the redundant removal phase, despite the spike in prediction error for the TDMA MAC protocol.

In the non-redundant removal phase, gateways are removed one at a time to reduce the prediction error as much as possible. While doing so ensures that, except in the TDMA - TDMA case, the average prediction error remains below the 5%-boundary, the 95-percentile of the error still rises above 10% and can sometimes reach as high as 18%. As already suggested in section 4.4, it is thus not always possible to keep the prediction error within acceptable bounds when removing non-redundant gateways. Given that occasional spikes in the prediction error may thus cause the wrong virtual gateway to be removed, the selection algorithm includes the *whitelisting* mechanism discussed in section 5.1.2. This mechanism gives the selection algorithm limited *backtracking* capabilities that allow it to 'undo' any gateway removals that were the caused by a too high prediction error.

5.2 Evaluation

The test setup used to evaluate the selection algorithm of IRVG is similar to the one used to evaluate the prediction algorithm. The simulator, simulation parameters and application scenario are exactly the same as in section 4.3. The main difference between

that test setup and the one described here, is that the evaluated gateway configurations are determined by the selection algorithm rather than being randomly generated. 100 independent test runs are performed for each parameter-set. During each test run the selection algorithm is performed from start to finish. As with the evaluation of the prediction algorithm, all simulations within the same test run use the same set of 'random flows' and the same seed for the random number generator.

As discussed in section 5.1.3, the selection algorithm allows the network administrators to specify both *goals* and *weights* for each of the considered metrics. Although network administrators are of course free to set their own performance goals, in this work the 'No Interference' performance of the individual networks is used as a goal for the selection algorithm. The 'No Interference' performance of a network is the performance that network would achieve if (1) it were the only network in the wireless environment and (2) all the information that would normally require communication with another network to obtain, is available in the network itself. The reason for using this 'No Interference' network performance as a target for the selection algorithm, is that it can be considered to be a 'best case' scenario and that it thus would be unreasonable for a network administrator to expect the selection algorithm to yield a better performance than this. To obtain the 'No Interference' performance values, two additional simulations are performed at the beginning of each test run. For each network, the random-flows scenario is simulated with only one network present instead of two. All simulation parameters are the same as for the normal scenario, except that a different set of random flows is used.

IRVG is evaluated using a number of different test cases. Each of these assigns, for each of the networks, a different set of *weights* to the metrics. The test cases discussed in section 5.2.1 evaluate the performance of IRVG when optimising for a specific metric while section 5.2.2 discusses a more 'general' scenario in which IRVG has to optimise for multiple, often conflicting, metrics.

Since IRVG is a heuristic mechanism ideally its performance should, for each of these test cases, be evaluated against the performance of the 'optimal' gateway configuration since that would reveal how much performance is 'lost' by using a heuristic instead of a brute-force algorithm. Likewise, IRVG should ideally also be compared with a number of existing gateway placement algorithms. Unfortunately, it is not possible to do either of these comparisons. The test setup used to evaluate IRVG contains 50 nodes, each of which can be configured as a virtual gateway. Finding the 'optimal' gateway configuration would thus require 2^{50} (roughly 10^{15}) gateway configurations to evaluated, which is clearly not feasible. Moreover, using a smaller test setup containing less nodes is also not possible. Given the parameter ranges to consider (combinations of MAC protocols, seed values for the simulator, ...), the largest test setup small enough to allow the 'optimal' gateway configuration to be brute-forced, would only contain 8 nodes and even then it would require well over 10^6 configurations to be evaluated to actually find the 'optimal' gateway configuration for each set of parameters. Moreover, a test setup containing only 8 nodes would be too small to be representative of an actual network since, as discussed in chapter 2, the network size plays an important role in the effects of the interference between the two networks. In addition, the gateway placement algorithms that are described in the literature (see section 4.1.1), have been developed for an entirely different purpose than IRVG and the properties of the gateways these algorithms consider, are completely different from those of the virtual gateways considered here. As a result, no

meaningful comparison can be made between any of these algorithms and IRVG.

Instead of comparing the performance of IRVG against the performance of the 'optimal' gateway configuration, the performance of IRVG is therefore compared to the performance of a hypothetical 'Same MAC'-scenario in which, instead of using virtual gateways, communication between the two networks is enabled by having all nodes in one network switch to the MAC protocol of the other network. Doing so would of course be extremely detrimental to the performance of the network that has to change its MAC protocol. On the other hand, it can also be considered to be a 'best case' scenario for the network whose MAC protocol is used since this effectively eliminates interference between nodes using heterogeneous MAC protocols. Because of this, the performance of the 'Same MAC'-scenario is measured separately for each of the MAC protocols after which the 'best case' performance measurements of each network are combined into a single set of performance figures for further comparison. The performance of IRVG is thus compared against an 'ideal', but in reality completely impossible, scenario where each network enjoys the benefits of having all the nodes in the wireless environment using its preferred MAC protocol, but none of the networks suffers the performance impact of having to switch to a different MAC protocol itself. While comparing the performance of IRVG against the performance of the 'Same MAC'-scenario therefore does not reveal how much performance is 'lost' by possibly using a sub-optimal gateway configuration, it does reveal how close IRVG is able to match a network's 'best case' performance when using the only option available for enabling direct communication between MAC-heterogeneous networks that does not involve virtual gateways.

It should be noted that although the 'No Interference' scenario discussed above can, like the 'Same MAC'-scenario, be considered to be a 'best case' scenario, the 'No Interference' scenario is not a useful comparison point for the evaluation of IRVG. Unlike the 'Same MAC'-scenario, the 'No Interference'-scenario assumes all application-layer information to be available within the network itself and as a result the paths used in this scenario will be significantly shorter than when this information needs to be collected from nodes in another network. Because of this, not even the 'optimal' gateway configuration (that would need to be discovered by a brute-force algorithm) would be able to match the performance of the 'No Interference'-scenario and as a result the 'No Interference'-scenario would not be a fair point of comparison. Inversely and in contrast to the 'No Interference'-scenario, the measurements obtained for the 'Same-MAC'-scenario are not suitable for use as performance *goals* in IRVG. The reason for this is that, while one might reasonably expect a network administrator to have at least a general idea of what the 'No Interference'-performance of the network is (perhaps based on pre-deployment tests or historical data) one cannot expect the network administrator to have foreknowledge about what the network performance would be if an additional network, controlled by a different administrator, is deployed in the same wireless environment at a later date.

Since, as discussed above, IRVG cannot be compared to any of the existing gateway placement algorithms, it is instead compared to a number of 'random-selection' algorithms. The first random selection-algorithm generates virtual gateway configurations containing a *fixed* number of nodes. To do so it first performs a 'random shuffle' on the list of nodes and then selects the first n elements of the resulting list (where n, the number of gateways to select, is specified as a parameter). Given that the performance of these 'random' gateway configurations will most likely vary with their size, a number of differently-

sized gateway configurations are generated and subsequently evaluated during each test run. IRVG is compared to random gateway configurations with 5, 10, ... 30 nodes. To keep the graphs from 'ballooning' with superfluous information, not all sizes of random configurations are discussed. Only the performance of random configurations with 5, 10 and 15 nodes are shown in the graphs below since these are the most relevant. For those interested, the full data set can be found online [139].

To get an idea of how well IRVG is able to choose which, rather than how many, gateways to use two additional random gateway configurations are also generated and evaluated during each test run. The first one of these is generated using the above 'random selection' algorithm, with n set to the size of the configuration generated by IRVG. The generated configuration thus has the same number of gateways as the configuration generated by IRVG. The second random configuration is generated by a 'probability-based' selection algorithm which decides for each node individually whether or not to include it in the gateway configuration based on a fixed probability-parameter. This probability parameter is set to the average fraction of gateways selected by IRVG for a particular parameter-set. This allows the size of the randomly generated configurations to deviate from the size of those generated by IRVG while still ensuring that on average they contain approximately the same number of nodes. Although it was originally expected that the random configurations generated by these two algorithms would yield very different performance figures, this did not turn out to be the case. Instead, for all test cases, the performance of the 'probability-based' configurations is extremely similar to that of the 'fixed-size' configurations. Because of this, the performance of the 'probability based' configurations, though present in the full data set [139], is not shown in the graphs below.

5.2.1 Single-metric optimisation

In this section, the performance of IRVG is investigated separately for each individual metric. To do so, for each network a weight of 1.0 is assigned to the metric being considered while the other metrics are assigned a weight of 0.0. This effectively causes IRVG to optimise to that specific metric only, while ignoring the other ones. That in turn allows the performance of the individual metrics to be compared directly across multiple scenario's (e.g., compare the node duty cycle obtained when using IRVG to the node duty cycle obtained with one of the 'random' placement algorithms). Such comparisons are not possible when performing multi-metric optimisation since, in that case, it is impossible to distinguish between a lower-than-expected performance for a single metric being the result of IRVG giving priority to another metric or this being caused by a limitation of the algorithm itself.

In the graphs below, the performance measurements obtained for the different scenarios are expressed as a '*relative difference*' compared to the performance of the baseline ('Same MAC') scenario rather than as an absolute performance figure. The difference between these two ways of displaying the collected measurements is exemplified by figures 5.7, 5.11 and 5.15. The left graph of each of these figures shows the average, 5- and 95-percentiles of the raw performance measurements obtained from the networks while the graph on the right shows the same data using the *relative* representation that is also used in the other graphs in this section. The reason for displaying the data in this manner is that the performance of the networks tends to vary quite a bit from one set of MAC protocols to

another and that, regardless of the specific set of MAC protocols used, this representation makes it immediately clear how much performance is 'gained' or 'lost' compared to the baseline scenario. Each of the graphs below shows the (relative) performance of IRVG as well as that of the different random virtual gateway configurations discussed above. In addition, the performance of IRVG in the case that the 'early stop' check is disabled is also shown. When the 'early stop' check is disabled, IRVG continues to remove (and whitelist) virtual gateways even if the expected reward of the best gateway configuration is lower than the current actual reward. This performance figure is included to reveal how much performance is 'lost' because of IRVG stopping early as a result of inaccuracies in the predictions. Finally, the performance of the *minimum gateway* configuration is also shown as it serves as a 'fallback' configuration for IRVG (see section 5.1.2).

5.2.1.1 Node Duty Cycle

The performance of IRVG when optimising only for node duty cycle is shown in figures 5.7 to 5.10.

When combining CSMA/CA with the T-MAC protocol, the duty cycle of the CSMA/CA network does not vary across the different scenarios. This is hardly surprising given that when using this MAC protocol, the node duty cycle is always 100%. For the T-MAC network, IRVG is able to *improve* on the performance of the 'Same MAC' scenario. Although it might seem counterintuitive that IRVG would perform better than what might be considered an 'ideal scenario' for the T-MAC network, this behaviour can be explained by the fact that the T-MAC protocol will extend its 'active period' every time it overhears a T-MAC packet from another node and there is thus a duty cycle-wise cost associated with increasing the number of T-MAC nodes in the wireless environment.

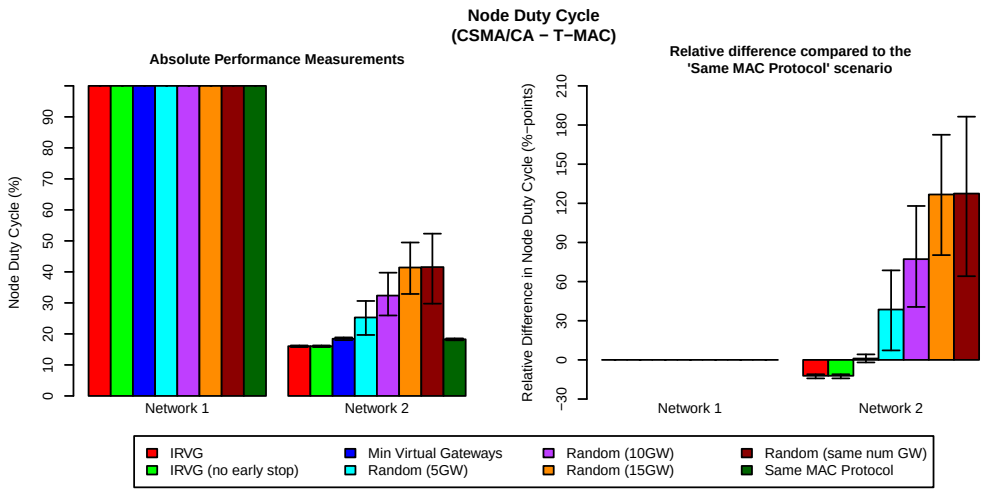


Figure 5.7: Average, 5- and 95%-tile of the node duty cycle in the 'single metric' optimisation case when using CSMA/CA and T-MAC. The left graph shows the absolute node duty cycle measurements while the graph on the right shows the relative difference of the node duty cycle when compared to the 'Same MAC' scenario.

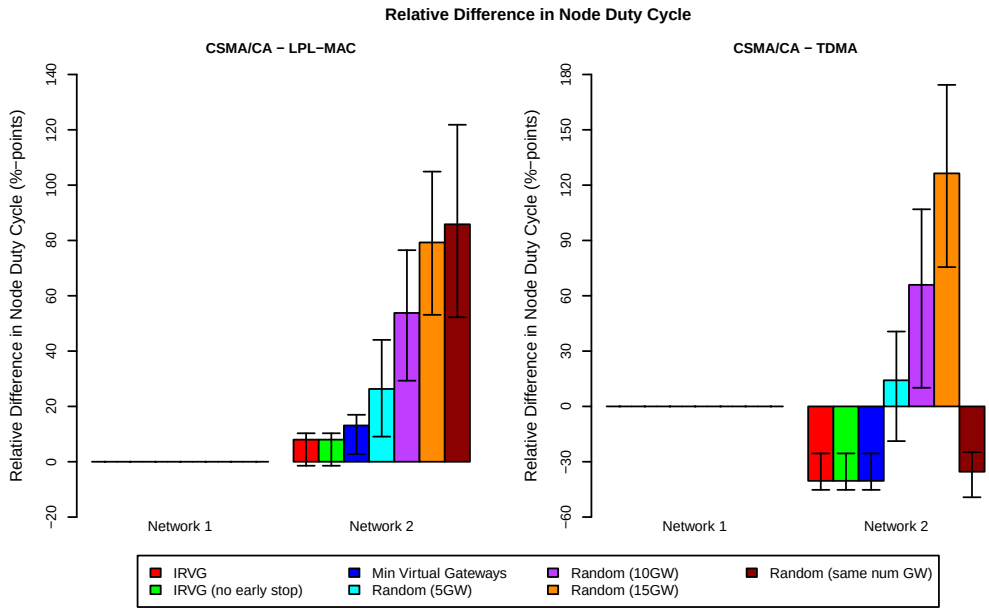


Figure 5.8: Average, 5- and 95%-tile of the relative difference in node duty cycle in the 'single metric' optimisation case when using either CSMA/CA and LPL-MAC or CSMA/CA and TDMA.

This does not happen when a CSMA/CA packet is overheard because in that case, the packet is dropped before it can be processed by the T-MAC protocol. In addition, IRVG performs significantly better than any of the 'random' scenarios which, depending on how many gateways are used, incur an average performance overhead between 42% and 125%. Finally, it should also be noted that there is a huge difference in performance between IRVG and the random configurations using the same number of virtual gateways as IRVG. This is because IRVG is able to take advantage of the fact that, for nodes using CSMA/CA as their default MAC protocol, the duty cycle is already 100% and that running an additional MAC protocol therefore does not incur a duty cycle-wise cost for these nodes. This causes IRVG to place all virtual gateways in the CSMA/CA network which results in the significant node duty cycle improvement shown in figure 5.7.

When combining CSMA/CA with the LPL-MAC protocol, a similar behaviour as in the CSMA/CA - T-MAC-case can be observed. The only notable difference is that, for the LPL-MAC network, IRVG is not quite able to match the performance of the 'Same MAC'-scenario. This indicates that there is indeed a performance-wise cost to be paid when enabling communication between the networks in this manner rather than 'forcing' the other network to switch to a different MAC protocol. Even so, this performance cost is fairly small (only around 8%). Moreover, as in the CSMA/CA - T-MAC case, IRVG performs significantly better than any of the 'random selection' scenarios which, in this case, incur an average performance overhead between 28% and 90%.

When the CSMA/CA MAC protocol is used in combination with TDMA, IRVG again outperforms the 'Same MAC'-scenario, this time with an average margin of 40%. This

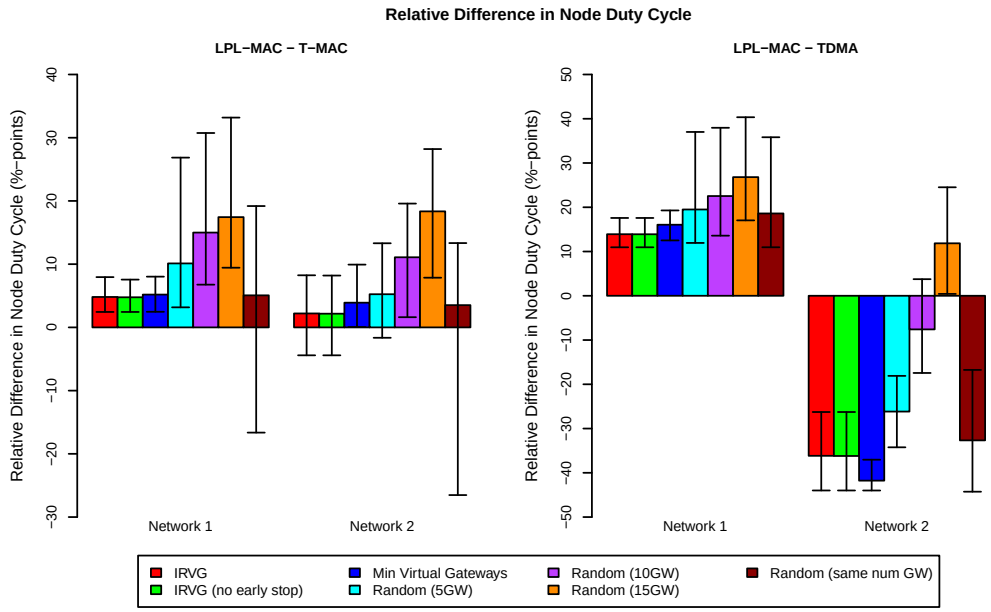


Figure 5.9: Average, 5- and 95%-tile of the relative difference in node duty cycle in the 'single metric' optimisation case when using either LPL-MAC and T-MAC or LPL-MAC and TDMA.

is mainly caused by the fact that, as discussed in chapter 2, the TDMA implementation used here reserves a 'SYNC'-slot and multiple 'broadcast'-slots for every node using the TDMA MAC protocol. Given that all nodes need to stay awake during these slots there is a significant energy cost associated with increasing the number of TDMA nodes. This means that, when node duty cycle is the only metric of importance, it is better to enable communication using virtual gateways (and IRVG) than to basically double the number of TDMA nodes. The need to minimise the number of TDMA nodes, is also visible in the number of gateways enabled by IRVG. In this case, IRVG tries to reduce the number of gateways as much as possible and therefore always either 'falls back' to the minimum gateway configuration or keeps disabling gateways until the minimum gateway configuration is reached. Because of this, the performance of IRVG is exactly the same as that of the *minimum gateway* configuration. Moreover, the difference in performance between IRVG and the random configurations using the same number of gateways as IRVG is significantly smaller than it is in either the CSMA/CA - LPL-MAC or CSMA/CA - T-MAC case. This indicates that the node duty cycle of the TDMA network is mostly decided by the *number* of gateways used, and not the *position* of these gateways.

When LPL-MAC is used in combination with the T-MAC protocol, IRVG is able to nearly match the performance of the 'Same MAC' scenario: the overhead is, on average, less than 5% for the LPL-MAC network and around 2% for the T-MAC network. In addition, IRVG also performs better, on average, than any of the randomly generated virtual gateway configurations. When the performance of the random configurations with the same number of gateways as IRVG is considered however, it is clear that there is significantly more variation in the performance of these random configurations than in

the performance of IRVG. More importantly, the 5-percentile node duty cycle measured for these configurations is also significantly lower than the one measured for IRVG. This indicates that although IRVG yields *on average* a better performance than the same-sized random configurations, there are also a number of test runs for which the opposite is true.

To explain why this is the case, it should first be noted that a similar observation can be made for the *minimum gateway* configurations. As for the configurations generated by IRVG, the 5-percentile node duty cycle recorded for these minimum configurations is also significantly higher than the one recorded for the "Random (same num GW)" configurations. In contrast to the configurations generated by IRVG however, these minimum gateways also have a lower *average* performance than the randomly generated configurations. Secondly it should also be noted that, when only optimising for duty cycle, IRVG will try to limit the number of virtual gateways as much as possible. (This is because, as discussed in section 3.2.2, there is a noticeable duty cycle-wise cost associated with turning a node into a virtual gateway.) Given that, as discussed in section 5.1.2, the *minimum configuration* must always be included in the *final configuration*, this causes IRVG to generate virtual gateway configurations that are very close to the *minimum* configuration. Despite this a closer examination of the test results also revealed that for the LPL-MAC - T-MAC case the final configuration is never *equal* to the minimum configuration and always includes one or more additional gateways. This shows that, despite the duty cycle wise cost of doing so, the performance of the *minimum* configuration is low enough to warrant the addition of one or more additional virtual gateways to the final configuration. Moreover, the test results also revealed that in over 90% of the cases where IRVG was outperformed by a same-sized random-configuration, the final configuration contained only two virtual gateways (one of which is the one present in the minimum configuration). In these cases, the specific node selected for the minimum configuration will have had a significant impact on the performance of the final configuration. The fact that IRVG is sometimes outperformed by a same-sized random configuration is therefore most likely caused by IRVG selecting a suboptimal minimum gateway configuration in these cases. To understand why a suboptimal minimum configuration is being selected, it is important to realise that, as discussed in section 5.1.2, this configuration is selected during the *preparation phase* of the selection algorithm. During this phase, the reward-based selection capabilities of IRVG cannot be used yet since these rely on the prediction algorithm which, in turn, needs the topology information collected for the *minimum gateway* configuration to operate. Because of this, the *minimum gateway* configuration can only be selected using somewhat basic selection strategies. For the random-flows application scenario investigated here, the minimum configuration therefore always consists of a single node which is the 'most used' virtual gateway from the maximum configuration (see section 4.2). While this strategy ensures that the selected gateway is at least used for a significant portion of the inter-network traffic, it also creates a strong preference for the few gateways that are 'the most central' in the wireless environment. For the LPL-MAC - T-MAC test case considered here, four specific nodes covered the *minimum gateway* configurations of nearly 75% of all test runs. Moreover, this selection strategy does not ensure that the *minimum gateway* configuration is in any way suitable for the requirements specified by the network administrators. Normally this needn't be a problem since the other gateways in the *final* configuration of IRVG are chosen more intelligently. In the case where the final configuration is very close to the minimum configuration however,

this can have a significant impact on the performance of IRVG.

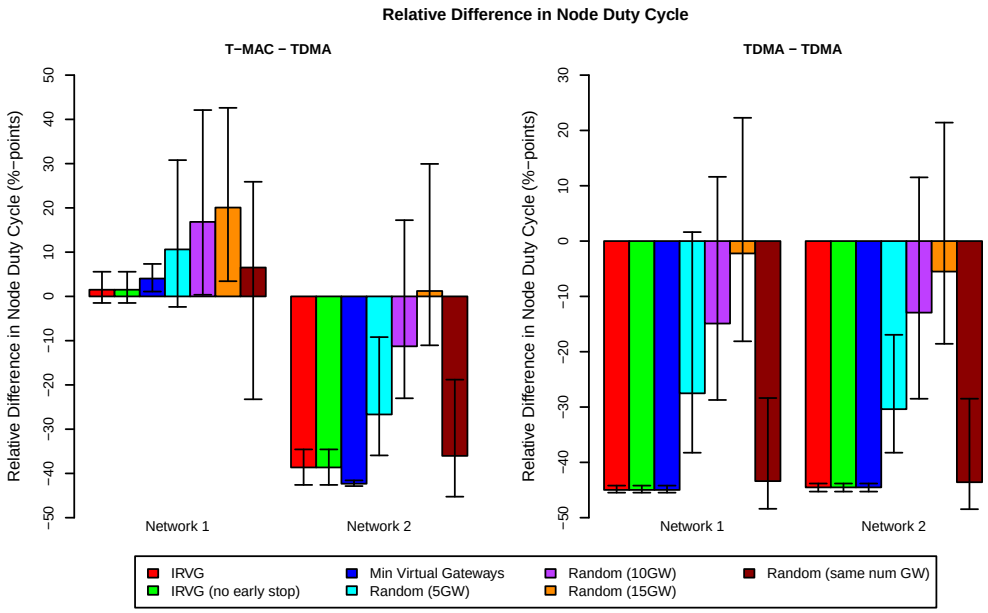


Figure 5.10: Average, 5- and 95%-tile of the relative difference in node duty cycle in the 'single metric' optimisation case when using either T-MAC and TDMA or two TDMA MAC protocols.

It should be noted that this limitation of IRVG is only an issue when node duty cycle is the only metric being optimised. Once a trade-off is made between multiple metrics, the generated configurations will grow to accommodate these and as a result a possibly suboptimal *minimum gateway* configuration won't have as large an impact on the performance. Moreover, even when solely optimising the node duty cycle IRVG is for the LPL-MAC - T-MAC case only outperformed by the random configuration in a quarter of all test cases. Finally, it should also be noted that, as evidenced by the error bars in the left graph of figure 5.9, in this case the performance of IRVG is significantly more predictable than that of the 'random selection'-algorithm.

In the LPL-MAC - TDMA case, IRVG outperforms the 'random' gateway placement algorithm for all gateway configuration-sizes. When the random configuration and the one generated by IRVG have the same number of nodes, the average performance difference however is quite small (around 3%). This is, again, due the manner in which the *minimum gateway* configuration is selected. More interesting, is that there is a significant difference in the performance gained (or lost) between the two networks. Using IRVG improves the performance of the TDMA network by more than 35%. At the same time, it also incurs a performance overhead of, on average, 14% for the LPL-MAC network. Given that the same behaviour can be observed for the various 'random' configurations this shows that it is not always possible to find a balance between the performance of the two networks. It should also be noted that the *final* configuration of IRVG yields a lower node duty cycle than the *minimum gateway* configuration for the LPL-MAC network while the opposite is true for the TDMA network. This shows that, in this case, IRVG prioritises

the performance of the LPL-MAC network over the TDMA network in order to keep the 'performance gap' between the two networks as small as possible.

For the T-MAC - TDMA case a similar behaviour as in the LPL-MAC - TDMA case can be observed in the sense that the duty cycle of the TDMA network is improved significantly (by nearly 40%) at the cost of an increase in duty cycle for the T-MAC network. In contrast to the LPL-MAC - TDMA case however, the performance overhead incurred by the T-MAC network is noticeably smaller than it was for the LPL-MAC network (around 2% instead of 14%). In addition it should also be noted that, as in the LPL-MAC - T-MAC case the 5-percentile node duty cycle of the same-sized random configurations is noticeably lower than that of IRVG which indicates that for this combination of MAC protocols there are also a number of test runs in which IRVG is outperformed by these random configurations.

When two TDMA MAC protocols are used (and node duty cycle is the only important metric), it is nearly always better to enable communication by using virtual gateways rather than switching to a common MAC protocol. As shown in the right graph of figure 5.10 this is not only true when using IRVG (in which case the duty cycle is reduced by nearly 45%) but, as long as not too many virtual gateways are used, also when using a purely 'random' virtual gateway configuration. This is caused by the fact that, as discussed above, there is a significant energy cost associated with increasing the number of TDMA nodes in a single network. As with the CSMA/CA - TDMA case, IRVG therefore tries to reduce the node duty cycle by limiting the number of virtual gateways as much as possible and, in this case, therefore always adopts the *minimum gateway* configuration as the *final* configuration.

5.2.1.2 Node Hop Count

The performance of IRVG when optimising only for node hop count is shown in figures 5.11 to 5.14. For all combinations of MAC protocols, IRVG performs significantly better than any of the gateway configurations generated by the 'random selection' algorithm. Moreover, the performance of the different sized 'random configurations' also show that for all cases there is a definite performance benefit to using somewhat 'larger' virtual gateway configurations when optimising for node hop count. This is in contrast to the 'duty cycle only' case discussed above where virtual gateway configurations are usually as small as possible.

The performance of IRVG for the CSMA/CA - T-MAC case is shown in figure 5.11. The graph on the left shows the *absolute* node hop count measurements obtained from the network while the graph on the right shows the performance relative to the performance of the (baseline) 'Same MAC'-scenario. From the data shown in the graph on the right, it would seem that there is an imbalance between the performance of the two networks since using IRVG improves the performance of the T-MAC network by over 33% compared to the 'Same MAC'-scenario at the cost of a 13% performance overhead for the CSMA/CA network. This discrepancy however is not caused by any limitation of IRVG and can be explained by considering the *absolute* node hop count measurements which show that, despite the large difference in *relative* performance between the networks, there is little to no difference between the average *absolute* node hop counts of these networks. This is not only true for the virtual gateway configurations generated by IRVG, but also for

the minimum gateway configurations and for the various configurations generated by the 'random selection' algorithm. For the 'Same MAC'-scenario however, the average node hop count of the T-MAC network is almost twice as large as it is for the CSMA/CA network. Given that the performance of the 'Same MAC'-scenario is used as a *baseline* for the *relative* performance (shown in the graph on the right), the large difference in *relative* performance between these networks can therefore be largely attributed to the difference in the *baseline* ('Same MAC') performance of these networks.

The performance of IRVG for the CSMA/CA - LPL-MAC and CSMA/CA - TDMA cases is shown in figure 5.12. For the CSMA/CA - TDMA case, IRVG is able to match the 'Same MAC'-performance of both networks while for the CSMA/CA - LPL-MAC case, IRVG nearly matches the performance of the LPL-MAC network while also improving the performance of the CSMA/CA network by 5%. Comparing the performance of the CSMA/CA network across all three cases where CSMA/CA is used by one of the networks, also reveals that the performance of the CSMA/CA network varies with the MAC protocol used by the other network. This is most likely due to the fact that the node hop count of a network is calculated as a weighted average over the lengths of the routing paths used to transmit data for that network (see sections 4.1.3 and 5.1.3). Moreover, most of these paths connect nodes of different networks which means that the MAC protocols of both networks are used to forward the traffic along these paths. As discussed in chapter 2, each MAC protocol reacts differently to the interference it receives from the other MAC protocol, and even when there is no interference, the average node reliability and node hop count vary with the MAC protocol used. Because of this, it should come as no surprise that, as is the case for the CSMA/CA network, the node hop count recorded for one network will, at least partially, depend on the MAC protocol used by the other network. When CSMA/CA is for instance combined with LPL-MAC, this causes the CSMA/CA network to benefit from the fact that using LPL-MAC tends to result in a lower node hop count. Conversely, when CSMA/CA is combined with T-MAC this results in a 14% performance overhead since, as evidenced by the node hop count measurements for the 'Same MAC'-scenario shown in figure 5.11, using T-MAC tends to result in a higher node hop count.

The performance of IRVG for the other combinations of MAC protocols are shown in figures 5.12, 5.13 and 5.14. In both the LPL-MAC - T-MAC case and the T-MAC - TDMA case, a behaviour similar to the CSMA/CA - T-MAC case can be observed. In both cases IRVG is able to improve the performance of the T-MAC network by more than 35% compared to the 'Same MAC'-scenario, while also inducing a 14% performance overhead for the other network involved. As with the CSMA/CA - T-MAC case discussed earlier, this is behaviour is caused by the fact that, for the 'Same MAC'-scenario the node hop count of the T-MAC network is significantly higher than that of the other network involved. When either CSMA/CA or LPL-MAC are combined with TDMA, IRVG is able to slightly improve the performance of the TDMA network (around 3%) at the cost of a slight performance overhead (on average at most 2.5%) for the other network involved. Finally, when two TDMA MAC protocols are used IRVG is able to match the performance of the 'Same MAC'-scenario for both networks while, as with the other cases, performing significantly better than any of the random configurations.

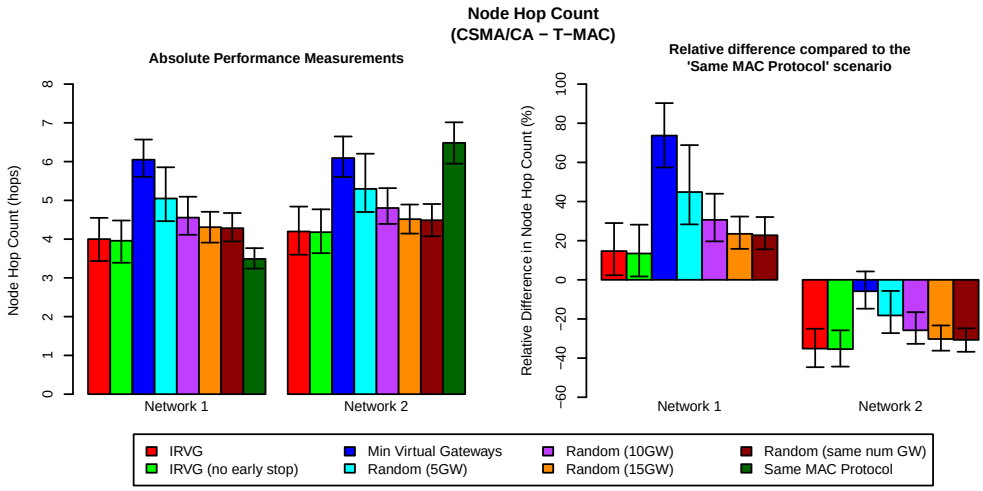


Figure 5.11: Average, 5- and 95%-tile of the node hop count in the 'single metric' optimisation case when using either CSMA/CA and T-MAC.

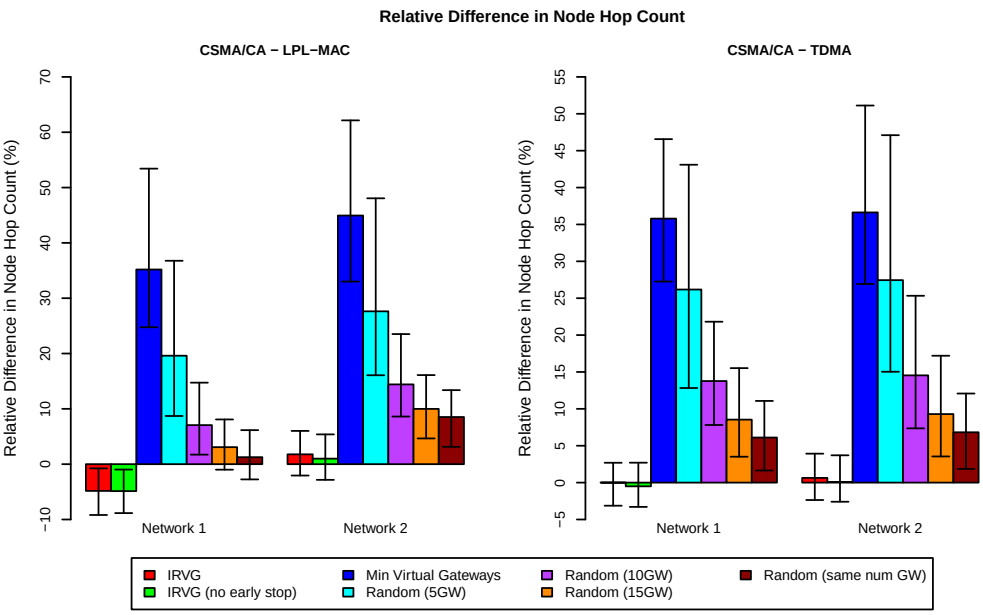


Figure 5.12: Average, 5- and 95%-tile of the node hop count in the 'single metric' optimisation case when using CSMA/CA in combination with either LPL-MAC or with TDMA.

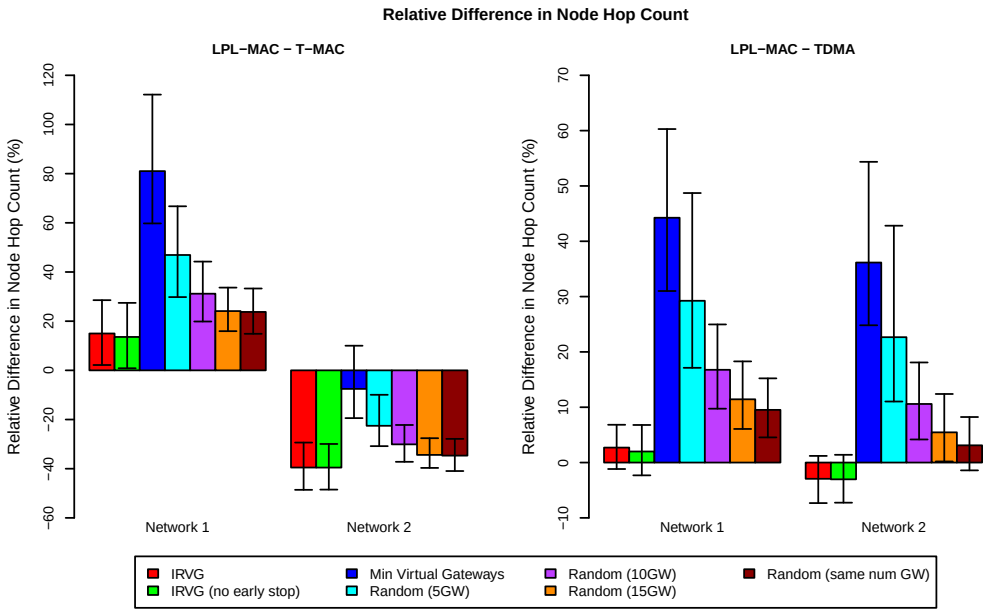


Figure 5.13: Average, 5- and 95%-tile of the node hop count in the 'single metric' optimisation case when using LPL-MAC in combination with either T-MAC or with TDMA.

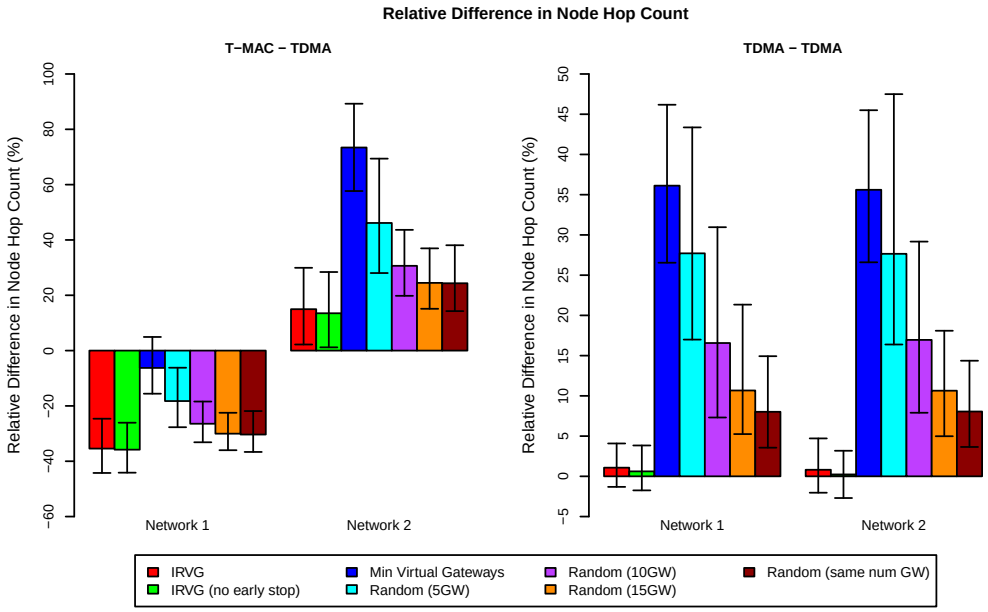


Figure 5.14: Average, 5- and 95%-tile of the node hop count in the 'single metric' optimisation case when using either T-MAC and TDMA or two TDMA MAC protocols.

5.2.1.3 Node Reliability

The performance of IRVG when optimising only for node reliability is shown in figures 5.15 to 5.18. For all combinations of MAC protocols, IRVG performs significantly better than any of the gateway configurations generated by the 'random selection' algorithm. Moreover, in all cases the performance of the randomly generated configurations tends to rise with the size of the random configuration. This shows that, as when optimising for node hop count, there is a definite performance benefit to using a sufficiently large gateway configuration when optimising for node reliability.

The performance of IRVG for the CSMA/CA - T-MAC case is shown in figure 5.15. The graph on the left shows the *absolute* node reliability measurements obtained from the network while the graph on the right shows the performance relative to the performance of the (baseline) 'Same MAC'-scenario. Similarly to the node hop count metric discussed above, it would seem that for this combination of MAC protocols there is, again, an imbalance between the performance of the CSMA/CA and the performance of the T-MAC network. The left graph of figure 5.15 however shows that for both IRVG and any of the random virtual gateway configurations, there is once again little to no difference in the *absolute* node reliability measured for the networks. At the same time there is a significant difference between the node reliability of the CSMA/CA network and that of the T-MAC network. The large difference in *relative* performance between these two networks can therefore also be attributed to the difference in the *baseline* ('Same MAC') performance of these networks. In contrast to the node hop count metric however, IRVG only has a negligible impact on the performance of the CSMA/CA network while still improving the performance of the T-MAC network by more than 70%.

For the CSMA/CA - LPL-MAC case, IRVG is able to nearly match the 'Same MAC'-performance of the LPL-MAC network while also improving the performance of the CSMA/CA network by around 5%. When CSMA/CA is combined with TDMA however,

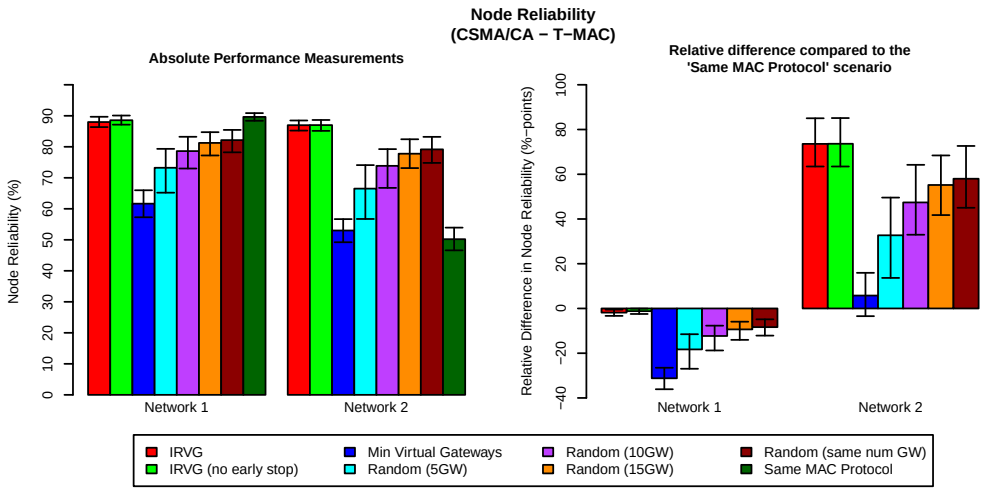


Figure 5.15: Average, 5- and 95%-tile of the node reliability in the 'single metric' optimisation case when using either CSMA/CA and T-MAC.

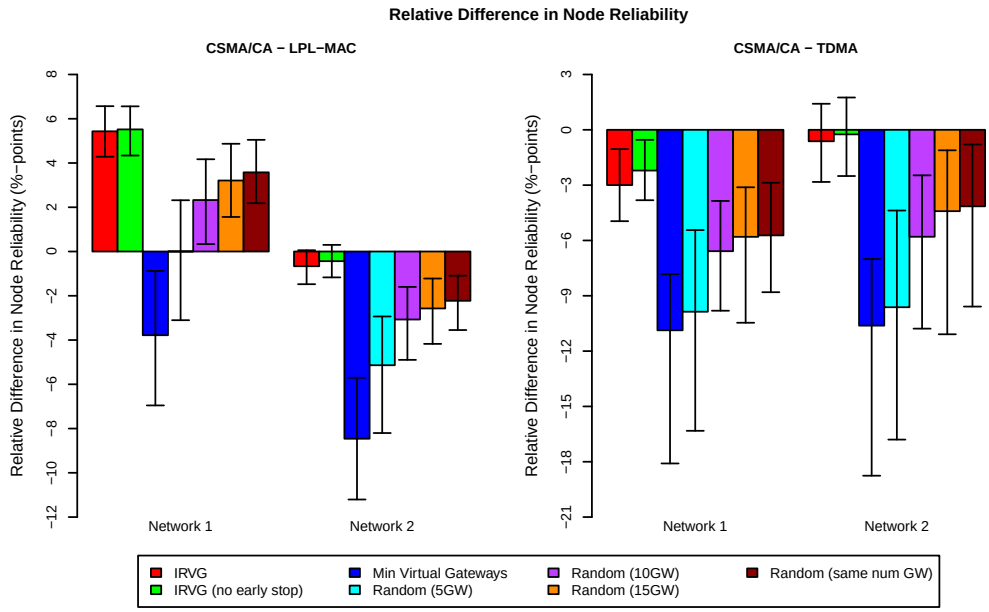


Figure 5.16: Average, 5- and 95%-tile of the node reliability in the 'single metric' optimisation case when using CSMA/CA in combination with either LPL-MAC or with TDMA.

IRVG is not quite able to match the 'Same MAC'-performance of the respective networks. Despite this, the performance overhead is very low (around 3% for the CSMA/CA network and around 0.75% for the TDMA network). This slight shortfall in performance is most likely the result of interference between the two MAC protocols (see chapter 2). In the case of CSMA/CA and TDMA it is not surprising that the node reliability of the TDMA network is somewhat impacted since this MAC protocol does not use any form of acknowledgements or retransmissions. The somewhat lower node reliability of the CSMA/CA network can also be attributed to this since traffic for the CSMA/CA network is also being forwarded by nodes in the TDMA network.

When LPL-MAC is combined with the T-MAC protocol, behaviour similar to the CSMA/CA - T-MAC case can be observed in the sense that IRVG is able to significantly improve the node reliability of the T-MAC network (nearly 85%) at the cost of a slight (3%) performance overhead for the LPL-MAC network. As with the CSMA/CA - T-MAC case, the seemingly large difference in *relative* performance between these two networks is caused by the fact that there is a significant difference in the *baseline* ('Same MAC') performance of these networks.

In the LPL-MAC - TDMA case, using IRVG improves the performance of the TDMA network by 4% while also incurring an equally large performance overhead for the LPL-MAC network. This slight imbalance in performance between the two networks can partially be attributed to the fact that there is a noticeable difference in the *baseline* ('Same MAC') performance of these networks and partially to the fact that traffic attributed to one network is also forwarded by nodes in the other network. Given that TDMA has a poorer reaction to interference from another MAC protocol than LPL-MAC (see chap-

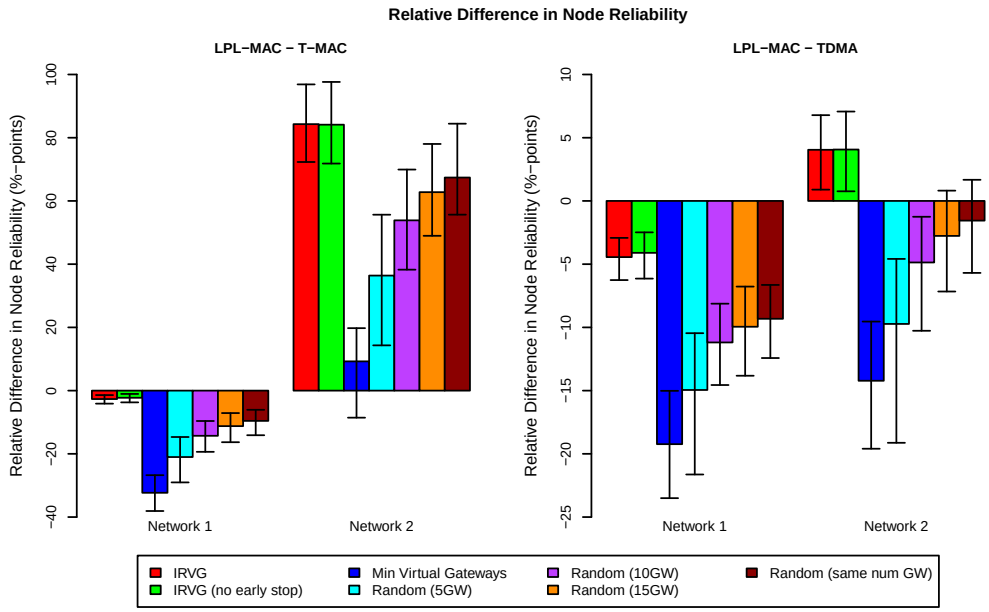


Figure 5.17: Average, 5- and 95%-tile of the node reliability in the 'single metric' optimisation case when using LPL-MAC in combination with either T-MAC or with TDMA.

ter 2), this causes the TDMA network to benefit from its traffic being forwarded by the LPL-MAC network while the opposite is true for the LPL-MAC network.

When T-MAC is combined with TDMA, both networks behave similarly to the CSMA/CA - TDMA and LPL-MAC - T-MAC cases discussed earlier. IRVG is once again able to significantly increase the node reliability of the T-MAC network (by around 65%) at the cost of a small (3%) performance overhead for the TDMA network. When two TDMA MAC protocols are combined however, the incurred performance overhead is larger (around 10.5%) than it is for any other combination of MAC protocols. Given that, as discussed in section 4.4, the TDMA - TDMA case is the most difficult one to predict, this higher performance overhead might be caused by IRVG being negatively affected by inaccuracies in the predictions made by the prediction algorithm. That being said it should be noted however that, in contrast to all other combinations of MAC protocols, in this case neither network employs a MAC protocol that is capable of (partially) mitigating the effects of inter-MAC interference on the reliability (the TDMA MAC protocol does not use CCA, ACKs or retransmissions). As a result, the larger performance overhead encountered for this case can also be partially attributed to inter-MAC interference. Regardless of what is the primary cause of the encountered performance overhead however, most sensor network applications should be able to cope with a 10% drop in reliability and as a result the performance overhead incurred in this case can still be considered to be within acceptable bounds.

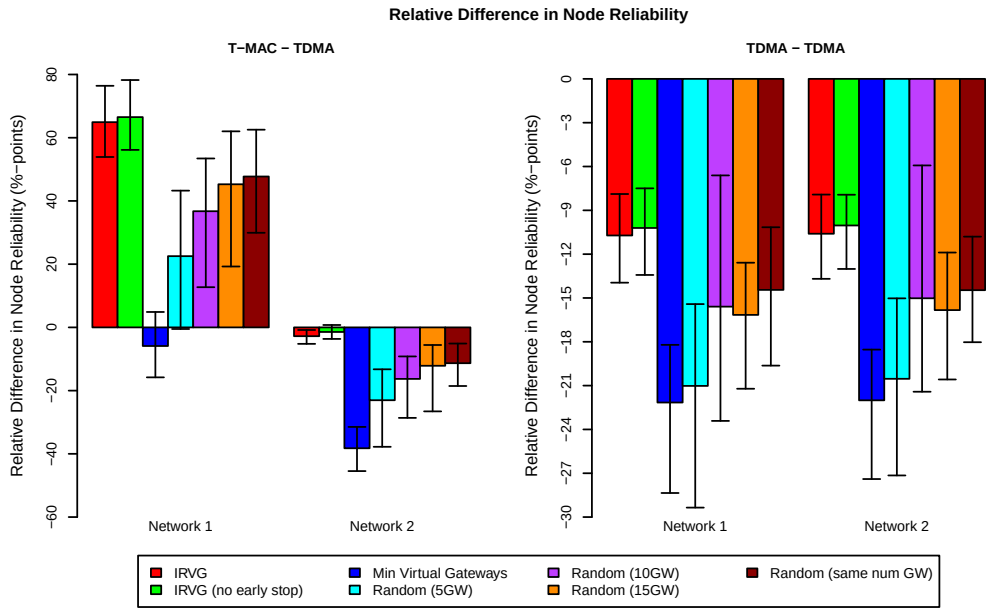


Figure 5.18: Average, 5- and 95%-tile of the node reliability in the 'single metric' optimisation case when using either T-MAC and TDMA or two TDMA MAC protocols.

5.2.2 Multi-metric optimisation

This section investigates the behaviour of IRVG when applied to a more 'real world' scenario in which multiple metrics need to be considered and a trade-off needs to be made between the possibly conflicting requirements of the different networks. As with the single-metric optimisation, the *goals* specified for each of the metrics are the performance values measured for these metrics in the 'No Interference' scenario. In contrast to the single-metric optimisation however, the *weights* assigned to these metrics depend entirely on the default MAC protocol used by the network. Except in the TDMA - TDMA case, each of the two networks thus specifies a different set of requirements for IRVG to optimise to.

	Duty Cycle	**Hop Count**	**Reliability**
CSMA/CA	0	0.7	0.3
LPL-MAC	0.5	0.3	0.2
T-MAC	0.4	0	0.6
TDMA	0.8	0	0.2

Table 5.1: Metric weights specified to IRVG for each of the MAC protocols used.

The different metric weights used for each MAC protocol are listed in table 5.1. These weights have been chosen based on the properties of the different MAC protocols. The node reliability of the network is taken into consideration regardless of the MAC protocol used. This is because, as explained in section 5.2.1.1, there is no point in setting up a sensor network if the node reliability is too low to allow information to be exchanged.

When using CSMA/CA, the radio is enabled all of the time and as a result there is no point in trying to optimise the node duty cycle of the network in the case that CSMA/CA is used as a default MAC protocol. Since using CSMA/CA should, under low traffic conditions, result in a fairly low delay (no need to wait for the receiver to wake up or for the correct transmission slot to begin), focus is instead shifted to reducing the node hop count since this also has a definite impact on the end-to-end delay. In contrast to CSMA/CA, both LPL-MAC and T-MAC are 'duty cycled' MAC protocols and as a result the node duty cycle of the network does need to be taken into consideration. When using LPL-MAC, node hop count is also taken into consideration. This is because LPL-MAC tends to keep the channel occupied for quite a long time when sending packets and there is thus a definite benefit to minimising the average node hop count of the network. In contrast, T-MAC emphasises more on node reliability since even in the 'No Interference' scenario the end-to-end node reliability for the T-MAC protocol tends to be somewhat lower than for the other MAC protocols and that node reliability is therefore an important metric to optimise. Finally, for the TDMA MAC protocol, most emphasis is placed on optimising the node duty cycle since minimising the node duty cycle is one of the most important reasons for using TDMA in a sensor network.

5.2.2.1 Reward & Performance tradeoffs

The total reward achieved by IRVG for the various combinations of MAC protocols is shown in figure 5.19. To illustrate the tradeoffs made by IRVG during the optimisation process, figures 5.20 and 5.21 also show the performance obtained for the individual metrics for the CSMA/CA - T-MAC and the T-MAC - TDMA case. In all three figures, the performance of IRVG (both with and without 'early stop') is shown alongside the performance achieved by the various 'random configurations'. The performance of the 'No Interference' scenario is also shown since the metric *goals* are set to the performance measurements obtained in this scenario. The performance of the 'Same MAC'-scenario is not shown as it is less relevant to the tradeoffs made by IRVG.

It should be noted that figure 5.19 uses a non-linear Y-axis to display the Total Rewards for the various combinations of MAC protocols. This is done to compensate for the fact that the exponential function used to calculate the reward (see section 5.1.3) tends to 'squeeze' the rewards together as the performance metrics near their goals (which makes it difficult to visually compare these rewards if they are plotted along a linear Y-axis). When considering the 'total rewards' shown in this figure, it is immediately clear that the reward of the 'No Interference' scenario is the same across all combinations of MAC protocols. This is because, when calculating the reward, the performance measurements are scaled relative to the specified goals which, in this case, are equal to the performance measurements of the 'No Interference' scenario. As a result, the total reward shown for the 'No Interference' scenario is the reward that would be achieved if *all* performance goals were to be met for both networks. As is clear from figure 5.19 however, neither IRVG nor any of the 'random configurations' are quite able to match the 'No Interference' reward. This is not because of any limitations of IRVG but rather because, as discussed at the beginning of section 5.2, the 'No Interference' scenario sets a standard that is simply to high to achieve, regardless of the virtual gateway configuration used.

For every combination of MAC protocols considered here, IRVG yields a better reward

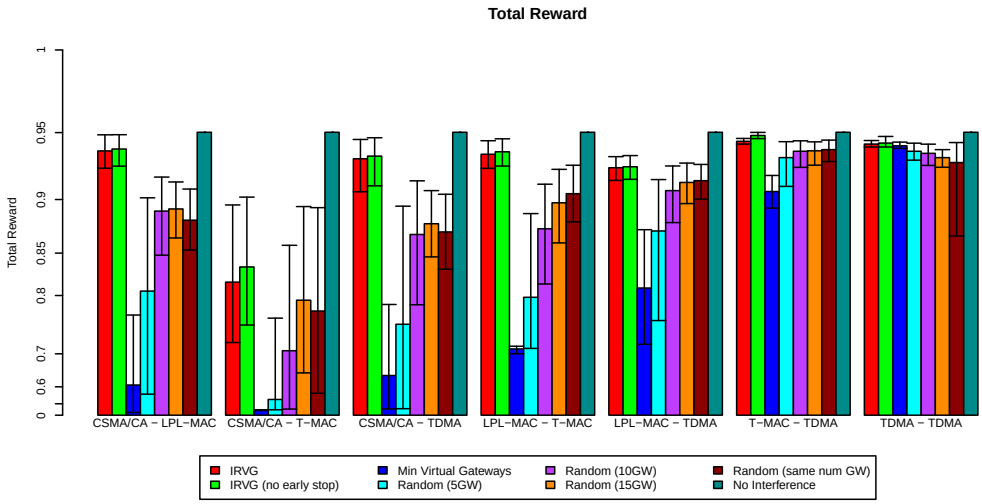


Figure 5.19: Average, 5- and 95%-tile of the Total Reward achieved by IRVG in the in the 'multi metric' optimisation case.

than any of the 'random configurations'. This is true both in the case that the 'early stop' check is enabled and in the case that this check is disabled. In addition, the 'reward gap' between IRVG and the various random configurations varies from one combination of MAC protocols to another. Although from these differences it would seem that in some cases (such as the TDMA-TDMA case) IRVG is only able to achieve a very small performance advantage over the random configurations, it should be noted that these differences in reward are in no way representative of the difference in performance for the individual metrics. As discussed in section 5.1.3, the reward function is designed in such a way that, for each metric, the same 'jump' in performance has an increasingly smaller effect on the reward as the performance of that metric nears its specified performance *goal*. The same difference in metric performance between two configurations can therefore result in either a large or a small difference in reward depending solely on the performance goals chosen for the individual metrics. While the total rewards shown in figure 5.19 thus reveal that IRVG yields a better performance than any of the random configurations, no statements can be made about the size of the performance difference between the different configurations based on the reward alone.

To gain a better insight into the actual performance figures underlying the total reward and the tradeoffs made by IRVG to achieve this reward, two specific cases are discussed in more detail: the CSMA/CA - T-MAC and the T-MAC - TDMA case. The performance of the individual metrics for the CSMA/CA - T-MAC case is shown in figure 5.20. In this particular case, IRVG heavily optimises both node hop count and node reliability at the cost of a slight overhead in node duty cycle for the T-MAC network. For both node hop count and node reliability there is only a relatively small difference in performance between the two networks despite the fact that these networks assign different *weights* to these metrics. This is because around 80% of the traffic benefitting each network is exchanged between nodes of different networks and each MAC protocol therefore has almost the same effect on the average node hop count and node reliability of both networks. The

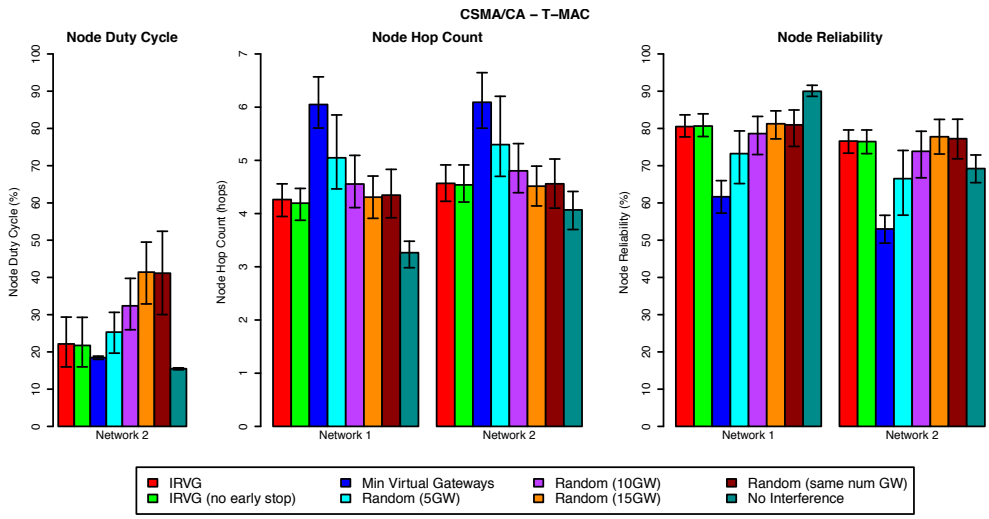


Figure 5.20: Average, 5- and 95%-tile of the node duty cycle, node hop count and node reliability for the CSMA/CA - T-MAC case. The node duty cycle of the CSMA/CA network is not shown as it is always 100%.

node reliability metric is important to both networks but the T-MAC network assigns a greater weight to it than the CSMA/CA network. The node reliability achieved by IRVG exceeds the *goal* set by the T-MAC network but does not reach the one set by the CSMA/CA network. One reason why IRVG stopped optimising the node reliability can be found by considering the node reliability of the various 'random' configurations. These reveal that the node reliability of the "Random (15GW)" and "Random (same num GW)" configurations are nearly equal to the node reliability of the configurations generated by IRVG. This indicates that, for this specific case, the number of gateways used is more important than the exact location of these gateways. Moreover, the 'jump' in node reliability resulting from the addition of more gateways quickly tapers off as the size of the (random) gateway configuration is increased. IRVG therefore most likely stopped optimising the node reliability because adding the gateways required to do so would have had too high an impact on the node duty cycle of the T-MAC network. The node hop count metric is only important to the CSMA/CA network but apart from that the situation is very similar to the node reliability metric. IRVG is not quite able to meet the specified node hop count goal for the CSMA/CA network and as with the node reliability metric this is most likely because adding additional gateways would only have had a very small effect on the average node hop count of the CSMA/CA network while also incurring a heavy node duty cycle wise cost for the T-MAC network. Finally, when considering the node duty cycle metric, it is immediately clear that IRVG could have optimised the node duty cycle of the T-MAC network further, since the node duty cycle of the *minimum* gateway configuration is around 3% lower than the node duty cycle resulting from the configuration generated by IRVG. Doing so however would have had a very significant impact on both the node hop count and the node reliability of both networks and as a result IRVG has 'chosen' to incur a slight overhead in node duty cycle in favour of the other metrics. Despite this, the node duty cycle of IRVG is still significantly lower than

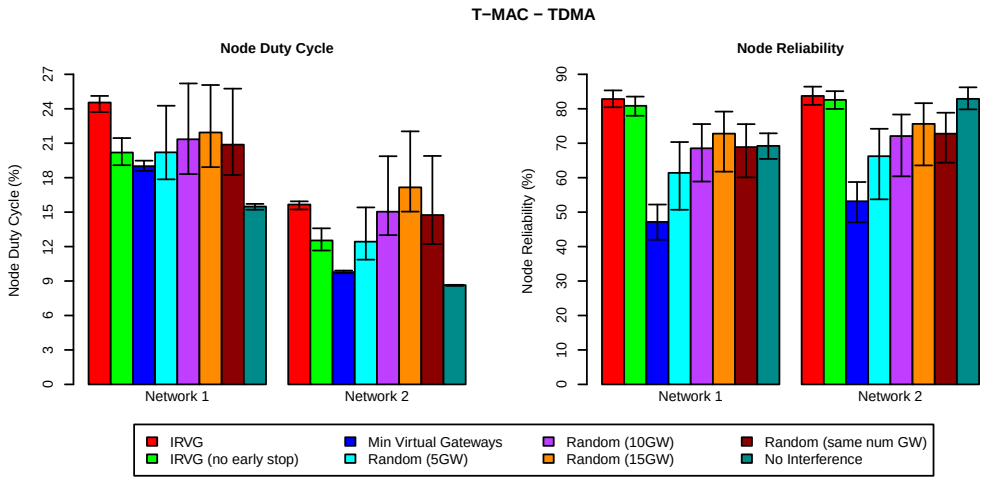


Figure 5.21: Average, 5- and 95%-tile of the node duty cycle and node reliability for the T-MAC - TDMA case.

the node duty cycle of the 'random' configurations containing the same number of nodes. This is because, as discussed in section 5.2.1.1, IRVG places all virtual gateways in the CSMA/CA network and that for nodes using CSMA/CA running an additional MAC protocol does not incur a node duty cycle-wise cost.

The individual metric performance for the T-MAC - TDMA case is shown in figure 5.21. The node hop count is not shown as it is not important to either network. For this combination of MAC protocols, IRVG limits the node duty cycle as much as possible without allowing the node reliability to drop below the specified node reliability *goals*. In addition it is also clear from the node reliability and node duty cycle measurements shown in this figure that, in contrast to all the previous cases where IRVG was applied, there is a definite difference in performance between IRVG with and without the 'early stop check' enabled. Disabling the 'early stop check' results in a significant drop in node duty cycle for both networks while the node reliability is only marginally affected. This behaviour is most likely caused by the prediction algorithm being too pessimistic about the impact of disabling (additional) virtual gateways on the node reliability of the networks. Although in this case the performance gap in thus not insignificant, it is also not so large as to render the 'early stop check' unusable. Moreover, users can decide whether or not to enable the 'early stop check' depending on their specific use case. The node reliability achieved by IRVG (with the 'early stop check' disabled) is almost exactly the same for both networks. In contrast to the CSMA/CA - T-MAC case however IRVG is able to both meet the node reliability goal of the TDMA network while also significantly exceeding the node reliability goal of the T-MAC network. Moreover, the node reliability achieved by IRVG is significantly better than the node reliability of the various 'random' configurations, including the random configurations that have the same size as the configurations generated by IRVG. This indicates that, in this specific case, not only the number but also the location of the selected gateways has a significant effect on the node reliability of both networks. When the node duty cycle of the networks

is considered it is clear that the node reliability achieved by IRVG does come at the cost of an increased node duty cycle. As with the CSMA/CA - T-MAC case, IRVG could have optimised the node duty cycle further by reverting to the *minimum* gateway configuration, but 'chose' not to do so because this would have had a detrimental effect on the node reliability of both networks.

5.2.2.2 Number of Iterations

Up until now, the evaluation of IRVG has mainly focussed on the performance of the gateway configurations selected by IRVG. Another important aspect of IRVG is the number of *iterations* it requires to generate the *final* gateway configuration. Since, during each iteration, a new virtual gateway configuration is applied to the networks and that there is a definite cost associated with doing so it makes sense to try and limit the number of iterations as much as possible as long as this does not have too great an impact on the overall performance of the *final configuration*. For this purpose, IRVG includes the 'early stop check' discussed in section 5.1.2 which reduces the number of required iterations by halting the optimisation process as soon as the removal of another gateway is not predicted to yield an additional increase in performance. This section investigates the effect that this 'early stop check' has on the number of iterations required by IRVG as well as the overall performance and the number of gateways in the final configuration.

The number of iterations required for IRVG to finish both with and without the 'early stop check' enabled is shown in figure 5.22. As shown in this figure, the 'early stop check' has a dramatic effect on the number of iterations required. When the 'early stop check' is disabled, the average number of iterations varies between 26 and 29, depending on the specific combination of MAC protocols used. Given that IRVG starts from a *maximum configuration* of 50 nodes, the number of iterations required in this case is thus quite considerable. The reason that such high numbers of iterations are required is that, when

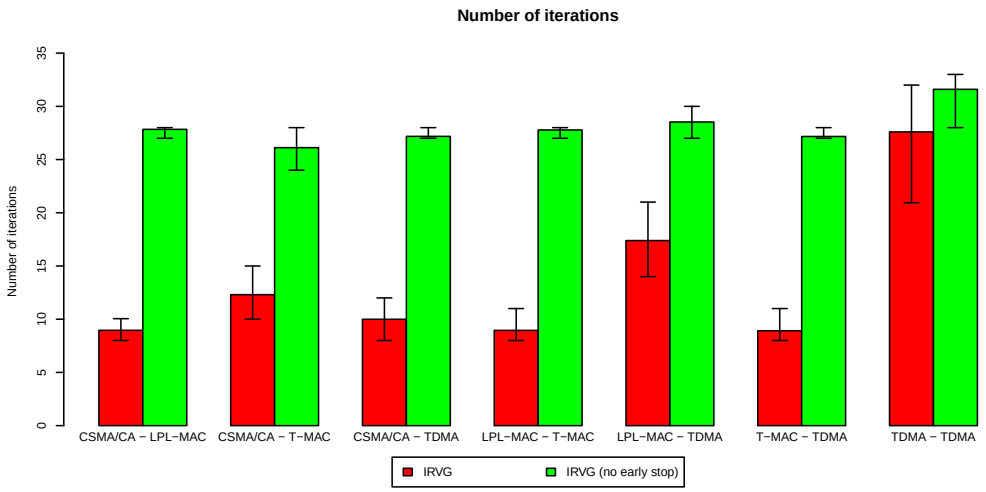


Figure 5.22: Average, 5- and 95%-tile of the number of iterations required by IRVG to complete the optimisation process for each of the combinations of MAC protocols.

the 'early stop check' is disabled, IRVG will try to remove every single *non-redundant* gateway that remains after the *redundant removal* phase. Since only half to, at the very most two thirds, of all gateways are removed during the first two phases of the selection algorithm, IRVG needs between 16 and 25 iterations *just* to complete the *non-redundant* phase of the optimisation process. When the 'early stop check' is enabled however, IRVG is able to stop much sooner and as a result IRVG is able to complete the optimisation process in half or even one third of the number of iterations that would otherwise be required. The only exception is the TDMA - TDMA case in which the 'early stop check' only reduces the number of iterations by around 10%. The reduced effectiveness of the 'early stop check' for that case is most likely due to the node duty cycle being extremely important to both networks. This causes IRVG to try and remove as many gateways as possible and given that only one non-redundant gateway is removed per iteration this also causes the number of iterations to increase.

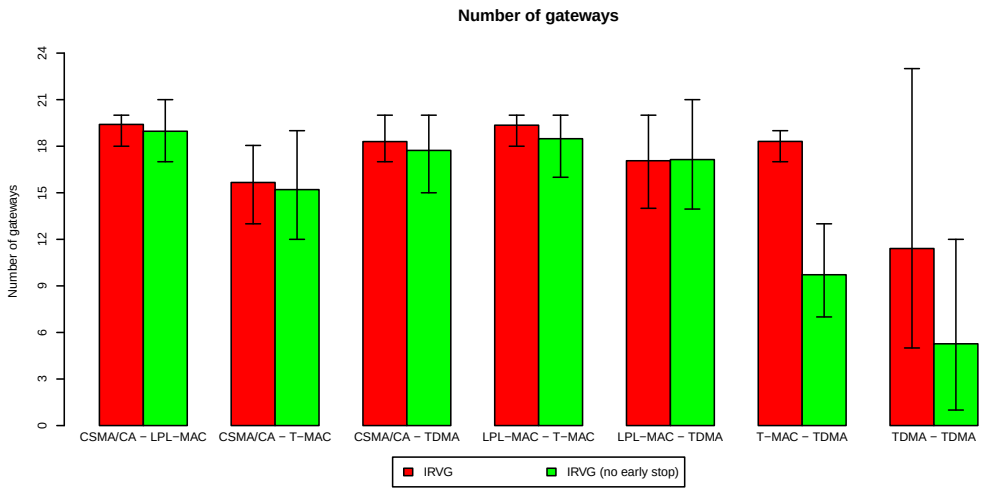


Figure 5.23: Average, 5- and 95%-tile of the number of gateways present in the *final* configurations generated by IRVG.

To allow the effect of 'early stop check' on the overall performance to be evaluated, all the graphs in sections 5.2.1 and 5.2.2 show the performance of IRVG both with and without the 'early check enabled'. Except for two notable exceptions, these figures show that for all test cases considered in this thesis, the performance difference between IRVG with and without the 'early stop check' enabled is either very minimal or even so small as to be negligible. The first exception is the CSMA/CA - T-MAC case when optimising for multiple metrics at the same time. For that case, figure 5.19 reports a relatively large difference in total reward between IRVG with and without the 'early stop check' enabled. This however is solely caused by the manner in which the reward is calculated given that, as shown in figure 5.20, there is very little difference in the performance of the individual metrics from which this reward is calculated. The other exception is the T-MAC - TDMA case (also) when optimising for multiple metrics. In that specific case, the inaccuracies of the predictions made by the prediction algorithm cause IRVG to abort the optimisation process prematurely which results in the node duty cycle being noticeably higher than when the 'early stop check' is enabled. Given that for this case the 'early stop check'

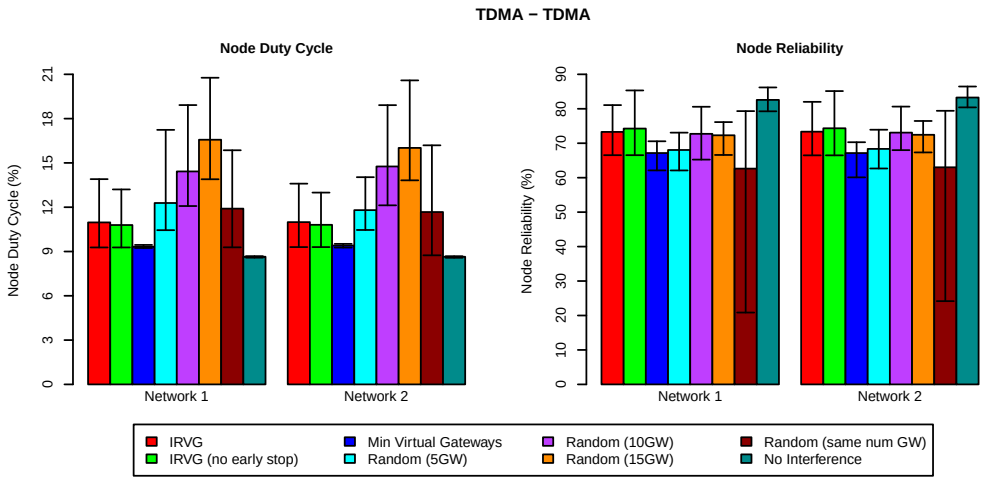


Figure 5.24: Average, 5- and 95%-tile of the node duty cycle and node reliability for the TDMA - TDMA case.

also severely reduces the number of iterations required, it might still be preferable to also enable this check for this specific case despite the somewhat lower performance of the final configuration. In addition, it should also be noted that, as shown in figures 5.10, 5.14 and 5.18, this issue does not occur for any of the single-metric optimisation test cases where T-MAC is combined with TDMA. While the 'early stop check' thus causes a performance overhead with the specific set of metric weights used here, this would not necessarily be the case if other metric weights were used.

Finally, the effect of the 'early stop check' on the number of gateways is shown in figure 5.23. This figure displays the number of gateways in the *final* configurations generated by IRVG both when the 'early stop check' is enabled and when it is disabled. For most combinations of MAC protocols, the 'early stop check' only has a minimal effect on the number of gateways present in the *final* configuration. For the T-MAC - TDMA and TDMA - TDMA cases however, enabling the 'early stop check' results in a *final* configuration that is significantly larger than when this check is disabled. As discussed above, for the T-MAC - TDMA case this is the result of the IRVG algorithm halting the optimisation process prematurely when the 'early stop check' is enabled. In the TDMA - TDMA case however, the additional gateways that are removed when the 'early stop check' is disabled only have a negligible effect on the overall performance. This is evidenced by the performance measurements displayed in figure 5.24, which show that the node duty cycle and node reliability achieved with the 'early stop check' enabled is nearly identical to the one achieved with the 'early stop check' disabled. The same is true for the total reward shown in figure 5.19. This shows that, for this case, the difference in number of gateways is not caused by the 'early stop check' halting the optimisation process prematurely. Instead it does exactly what it is designed to do and causes IRVG to stop optimising as soon as there is no further benefit to (trying to) remove additional gateways.

5.3 Conclusion

This chapter continued the discussion of the IRVG-mechanism first introduced in chapter 4. While chapter 4 mainly investigated the prediction algorithm of IRVG, this chapter focussed on the selection algorithm that decides which nodes are to be used as a virtual gateway. After first providing a rationale for using the heuristic selection strategy employed by IRVG, a detailed explanation of every aspect of the selection algorithm was provided in section 5.1. Afterwards, the performance of the entire IRVG mechanism was extensively evaluated in section 5.2.

The *single metric* optimisation tests discussed in section 5.2.1 show that when each metric is considered individually IRVG is, in most cases, able to match the performance of the 'ideal' baseline (Same-MAC) scenario. In addition, there are also a number of cases where IRVG is able to significantly improve upon the baseline performance, for instance when optimising for node duty cycle in the CSMA/CA - TDMA or TDMA - TDMA cases. Moreover in those cases where IRVG is not quite able to meet the baseline performance, the performance 'overhead' introduced by IRVG is fairly limited, especially when considering that the 'ideal' baseline scenario used for these tests is actually impossible in real life (see section 5.2). Finally, these tests also show that, with one exception (see section 5.2.1.1), IRVG consistently outperforms any of the randomly generated virtual gateway configurations. Although this is hardly surprising, given the amount of information available to IRVG, it does show that IRVG is able to use this information to its advantage and that there is a definite benefit to using IRVG in lieu of more primitive selection techniques.

The *multi metric* optimisation tests discussed in section 5.2.2 reveal that this is also the case when optimising for multiple and possibly conflicting metrics since, in all considered test cases, IRVG once again outperforms any of the randomly generated gateway configurations. Moreover, the results and examples discussed in section 5.2.2.1 also show that IRVG is, within the limits of what is physically possible, able to balance between the performance of different metrics based on the *goals* and *weights* provided by the network administrators.

When considering the number of iterations required by IRVG, the measurements discussed in section 5.2.2.2 show that, without the 'early stop check' enabled it can take quite a bit of time for IRVG to finish the optimisation process. Despite this, the 'early stop check' is, in most cases able to drastically reduce the number of iterations required while only having a minimal impact on the performance of the *final* configuration. The only case where the 'early stop check' incurs a noticeable performance overhead is the T-MAC - TDMA case (when optimising for multiple metrics) and even then, the performance difference does not necessarily prevent the 'early stop check' from being used.

The results of chapter 3 already showed that, although virtual gateways offer a viable method of enabling inter-network communication, great care should be taken in selecting the specific nodes to be used as a virtual gateway. The results of this chapter show that the IRVG mechanism is able to choose the gateways very carefully since it is able to near or even exceed the performance of the idealised 'Same-MAC' scenario and at the same time is capable of adjusting the selected gateways based on the requirements of the network administrators. This shows that, for the random-flows scenario discussed here, using

virtual gateways in combination with IRVG is thus a viable method of enabling internetwork communication and cooperation. The performance of the IRVG-mechanism for the node-to-sink scenario is discussed in chapter 6.

CHAPTER 6

Virtual Gateway Selection for the node-to-sink scenario

The previous two chapters introduced IRVG and investigated its performance for the random-flows scenario. This chapter investigates the use of IRVG in the node-to-sink scenario. As discussed in section 2.2.2, the node-to-sink scenario covers the more traditional sensor network use case in which sensor readings from different nodes need to be relayed to a so-called 'sink node' in the most efficient manner possible. In contrast to the random-flows scenario, nodes of different networks do not need to be able to communicate directly with one another in order for the application running in the networks to operate. For this scenario, the use of *virtual gateways* and *IRVG* therefore only serves to optimise the operation of the networks involved.

Given that virtual gateways make alternative routing paths available to both networks, it might be possible to achieve significant optimisations by *solely* using virtual gateways (in combination with IRVG). This however will only be the case for certain atypical node deployments for which the alternative routing paths are significantly shorter than the original ones. In the more general 'random grid' deployment considered in this thesis (see section 2.2.3) however, the nodes of both networks are mostly deployed in the same area and as a result the alternative routing paths made available through the use of virtual gateways are expected to have roughly the same length as the original ones. The optimisation achieved by IRVG alone is therefore also expected to be relatively minor in this case.

To achieve the largest possible optimisation, the involved networks not only need to use virtual gateways but also need to "share their sink nodes". This, in essence, means that every node relays its sensor readings to the nearest sink rather than to the sink of its own network and that these sink nodes then forward the sensor readings over the backbone network to the backend infrastructure of the network from which the sensor readings

originated. Depending on the location of the sink nodes within the wireless environment, doing so can considerably shorten the routing paths used and thus also reduce the interference between the heterogeneous MAC protocols. That being said however, "sink sharing" not only requires the networks to cooperate on the MAC- and routing layers of the network stack but also requires cooperation from the application(s) running in the network and the backbone infrastructure to which these networks are connected.

When investigating the performance of IRVG for the node-to-sink scenario it is assumed that "sink sharing" as described above is supported and enabled by both networks. Given that this thus requires these networks to cooperate "*across all layers and network boundaries*", the node-to-sink optimisation scenario discussed here is thus a prime example of the *Symbiotic Networking* paradigm [32] that was briefly discussed in the introduction of this thesis.

As with the random-flows scenario, the prediction and selection algorithm of IRVG are discussed and evaluated separately. Section 6.1 discusses the modifications that were made to the prediction algorithm to support the node-to-sink scenario after which the modified prediction algorithm is evaluated in section 6.2. Likewise, the changes made to the selection algorithm are discussed in section 6.3 after which this algorithm is evaluated in section 6.4.

6.1 Modifications made to the Prediction Algorithm

In order to support the "sink sharing" mechanism described above, a few modifications need to be made to the prediction algorithm of IRVG. The main issue is that this algorithm separates the traffic in the wireless environment into different 'data flows' based on the source and destination node between which the traffic is being forwarded and that these flows are moreover assumed to be independent from the specific gateway configuration used. This in essence means that when one or more virtual gateways are removed, only the routing paths used to forward the traffic are affected. The source and destination node of the flow are not expected to change from one virtual gateway configuration to another. When using "sink sharing" in the node-to-sink scenario however, this assumption is no longer valid. In that case, nodes will choose their sink node depending on the available routing paths and as a result the removal of a (non-redundant) gateway may cause a node to switch back from the sink node of the 'foreign' network back to the sink of its own network.

This causes problems when all the paths of a flow are 'broken' by the removal of one or more virtual gateways. When this happens, the various path replacement policies defined in section 4.2.3 all try to find a plausible set of replacement paths based on historical data. These policies however all operate based on the assumption that the source and destination node of a flow do not change as a result of gateways being removed. Because of this they only generate replacement paths that run between the original source and destination node of the flow. This means that when the removal of a gateway causes a node to switch from one sink node to another, the current path replacement policies will not be able to generate a suitable set of replacement paths. This in turn causes the prediction algorithm to assume that all the traffic of that flow will be lost while in fact it will merely be redirected to a different sink node.

To resolve this issue a new 'Equivalent Flow' path replacement policy, $EQFlow_x$, is added to the prediction algorithm. As with the other replacement policies defined in section 4.2.3, the 'x' denotes the topology from which the replacement paths are sourced and can refer to either the predicted, current, minimum or maximum topology. In contrast to the existing replacement policies, the 'Equivalent Flow' path replacement policy recognises that when "sink sharing" is enabled, all sink nodes can be considered to be 'equivalent destinations' and that if all the paths from a source node to a specific sink node are broken by the removal of a virtual gateway, they can be replaced by paths to another sink node.

ALGORITHM 5: Equivalent Flow Replacement Policy

```
1  Function EQFlow_x(i , j , TX_broken)
       Result: (PATHS_r, TX_r, E_r^L, E_r^R)
       // Initialise variables
2      PATHS_r ← ∅ ; E_r^L ← ∅ ; E_r^R ← ∅ ;
3      if j ∈ SINKS then
4          ReplacementSinks ← {s ∈ SINKS | s ≠ j ∧ FLOW_x(i, s) ≠ ∅};
5          for s ∈ ReplacementSinks do
6              for path ∈ FLOW_x(i, s) do
7                  (path_r, E_l, E_r) = SourceLinks (path) ;
8                  if path_r ≠ ∅ then
9                      PATHS_r ← PATHS_r ∪ {path_r} ;
10                     E_r^L ← E_r^L ∪ E_l ; E_r^R ← E_r^R ∪ E_r ;
11                 end
12             end
13         end
14         if PATHS_r ≠ ∅ then
15             TX_tot = Σ_{path∈PATHS_r} TX_x (path) ;
16             for path ∈ PATHS_r do
17                 TX_r(path) ← (TX_x(path) / TX_tot) TX_broken ;
18             end
19         end
20     end
21 end
```

The operation of the 'Equivalent Flow' path replacement policy is shown in algorithm 5. In this algorithm $SINKS$ is the set of all *shared* sink nodes (with $SINKS \subset V$). All other mathematical notations have previously been defined in section 4.2.1. On line 2 first a number of variables are initialised. Next, the replacement policy checks whether the destination of the flow is a sink node (line 3). If so, the set of sinks that can replace the original destination of the flow ($ReplacementSinks$) is constructed on line 4 after which these sinks are processed one at a time (lines 5 to 13). For each replacement sink s, each of the paths in '$FLOWS_x(i, s)$' (the set of paths from the source node of the broken flow to the replacement sink) is passed to the $SourceLinks$ algorithm to determine whether or not it is a feasible replacement path (line 7). The $SourceLinks$ algorithm itself is shown in algorithm 3 and further discussed in section 4.2.3. The $SourceLinks$ algorithm returns a tuple $(path_r, E_l, E_r)$ where $path_r$ is either the replacement path to be added to $PATHS_r$

or $\emptyset$, depending on whether the path passed to *SourceLinks* is a feasible replacement path or not. If the path is a feasible replacement path, E_l and E_r are the links to be added to E_r^L and E_r^R respectively (which is done on line 10). Once all paths and sinks have been processed, the traffic flowing over each of the paths in $PATHS_r$ is calculated (lines 15 to 18). As with the path replacement policies discussed in section 4.2.3, this is done by distributing the traffic of the flow over the different paths proportionally to the amount of traffic that was sent over these paths in topology 'x'.

To give an idea of the computational complexity of this path replacement policy, it should be noted that the overal computational cost of this replacement policy is due to the fact that it needs to scan all the different paths from the source to the different sinks (lines 5 to 13). For each of these paths, the *SourceLinks* algorithm is invoked. As discussed in section 4.2.3, this algorithm has a complexity of $\mathcal{O}\left(|candidate|(log(|E_p^L|) + log(|E_{min}^L|)\right)$. Given that the length of each candidate path is at most $Pmax_x$ and that, at the very worst at most $|PATH_x|$ paths need to be scanned, the complexity of the 'Equivalent Flow' path replacement policy can be expressed as:

$$\mathcal{O}\left(|PATHS_x|Pmax_x(log(|E_p^L|) + log(|E_{min}^L|)\right)$$

To ensure that the 'Equivalent Flow' replacement policy is actually used by the prediction algorithm, the *ReplacementPolicies* variable of algorithm 1 also needs to be updated. Within the scope of this chapter, this variable is initialised as follows:

$$\begin{aligned} ReplacementPolicies = \{&EQFlow_p, RFlow_p, MinGraph, \\ &EQFlow_{min}, RFlow_{min}, Concat_p, Concat_{min}\} \end{aligned}$$

It should also be noted that in scenarios where "sink sharing" is not used (such as the random-flows scenario), the set of "shared sinks" ($SINKS$) will be empty which causes the 'Equivalent Flow' policy to ignore the broken flow. In those cases, the 'Equivalent Flow' replacement policy thus has no effect on the predictions made by the prediction algorithm and the list of replacement policies specified here will yield exactly the same result as the one specified in section 4.2.3. Because of this, the above list of replacement policies can thus be used regardless of whether "shared sinks" are supported or not.

6.2 Evaluation of the Prediction Algorithm

The prediction algorithm is evaluated for the node-to-sink scenario in almost exactly the same manner as it was done for the random-flows scenario. The only differences are the changes to the prediction algorithm discussed above and the fact that a different application is running on the nodes in the sensor networks. As discussed in section 4.3, virtual gateways can be separated into three classes depending on how broad their removal is expected to affect the current route topology. As with the random-flows scenario, the accuracy of the prediction algorithm for the node-to-sink scenario is expected to vary with the class of the removed virtual gateways. Like the evaluation discussed in section 4.3, the prediction algorithm is therefore also evaluated separately for each class of virtual gateways.

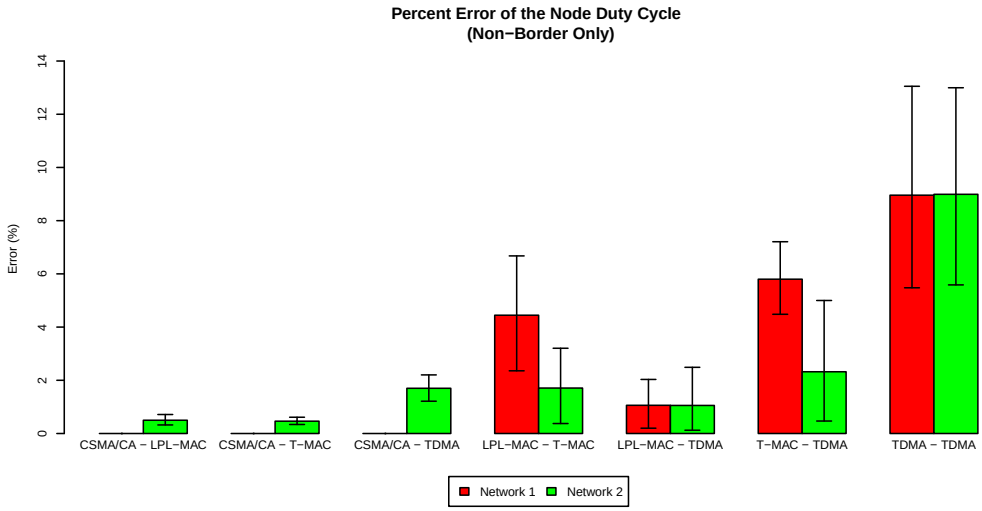


Figure 6.1: Average, 5- and 95%-tile of the percent error of the node duty cycle when removing 'Non-Border' gateways in the node-to-sink scenario.

6.2.1 Removal of non-border gateways

Figure 6.1 displays the average, 5- and 95-percentile error of the node duty cycle for all combinations of MAC protocols. For the three cases where CSMA/CA is used as well as for the LPL-MAC - TDMA case, the prediction error is very small (less than 2.5%). When LPL-MAC is combined with T-MAC, the error is considerably larger but is still within the "5% average, 10% 95-percentile" bounds set in section 5.1.5. When T-MAC is combined with TDMA the average error is around 5.75% which is higher than the 5% boundary but not dramatically so. In the TDMA - TDMA case however, the average and 95-percentile error are respectively 9% and 13% and are thus significantly higher than the bounds specified in section 5.1.5. Moreover, the error observed for these three cases is also considerably higher for the node-to-sink scenario considered here than for the random-flows scenario discussed in section 4.3.1. To explain why this is the case it should first be noted that the removal of a virtual gateway not only affects the duty cycle of the node itself but also that of the surrounding nodes. This is because the removal of a virtual gateway lowers the inter-MAC interference experienced by the surrounding nodes. To remain independent from the specific MAC protocols used however, the effects of this interference are not taken into consideration by the prediction algorithm and as a result the prediction error tends to rise with the number of removed gateways. While this is true for both the random-flows and the node-to-sink scenario, the influence of this interference on the duty cycle also tends to rise with the amount of traffic being sent in the vicinity of the nodes (see section 2.3.1) and thus also depends on the traffic patterns in and between the networks. In the node-to-sink scenario, most of the traffic is concentrated into two 'hot spots' around the sink nodes rather than being uniformly distributed throughout the wireless environment. Moreover, given the higher rate of traffic in these hot spots, there will also be more interference between the MAC protocols which means that the removal of a virtual gateway will also have a larger effect on the node duty cycle. This, combined

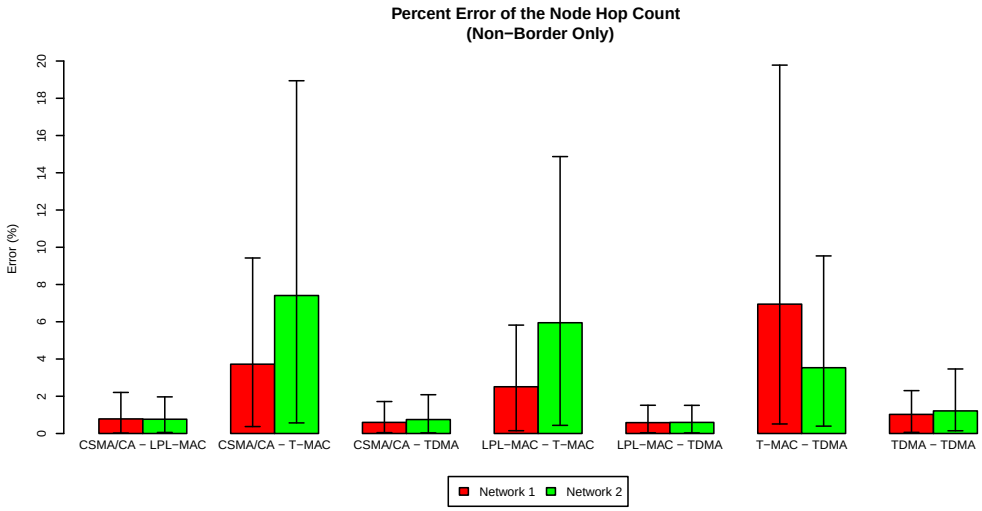


Figure 6.2: Average, 5- and 95%-tile of the percent error of the node hop count when removing 'Non-Border' gateways in the node-to-sink scenario.

with the fact that in the node-to-sink scenario there are on average 40% more non-border gateways to be removed than in the random-flows scenario (14 rather than 10), results in the larger prediction errors encountered here.

The percentile error of the node hop count is shown in figure 6.2. As this figure shows, the use of the T-MAC protocol has a very significant impact on the prediction error. When the T-MAC protocol is not used, the average prediction error is around 1% and the 95-percentile error is at most 3.5%. This is quite low considering that in the random-flows scenario the average and 95%-tile error are slightly higher than this.

When the T-MAC protocol is used however, the prediction error for the T-MAC network is extremely large with averages as high as 7.5% and 95-percentiles reaching nearly 20%. In addition, the prediction error of the other network is also noticeably higher. This is probably due to the fact that traffic from the non T-MAC network is (also) being sent to the sink of the T-MAC network and that traffic crossing the network-boundary is affected by the properties of the MAC protocols of both networks. This however does not explain why the prediction error for the T-MAC protocol is so much larger than for the other MAC protocols. To uncover why this might be the case, two additional set of test runs (each with 100 repetitions) were performed. The first set of test runs considered the 'No Interference' case introduced in section 5.2 for the T-MAC protocol, except that the node-to-sink application was used instead of the random-flows application. In this 'No Interference' case only the T-MAC network is active in the network (without outside interference). Likewise, the second set of test runs considered the 'Same MAC' case (also introduced in section 5.2) for the T-MAC protocol except that, again, the node-to-sink application was used instead of the random-flows application. In this case there are two networks active, both of which use the T-MAC protocol. Since in the 'Same MAC' case "sink sharing" is used in combination with a second sink, the average hop count for the 'Same MAC' case is expected to be lower than that of the 'No Interference' case. In reality

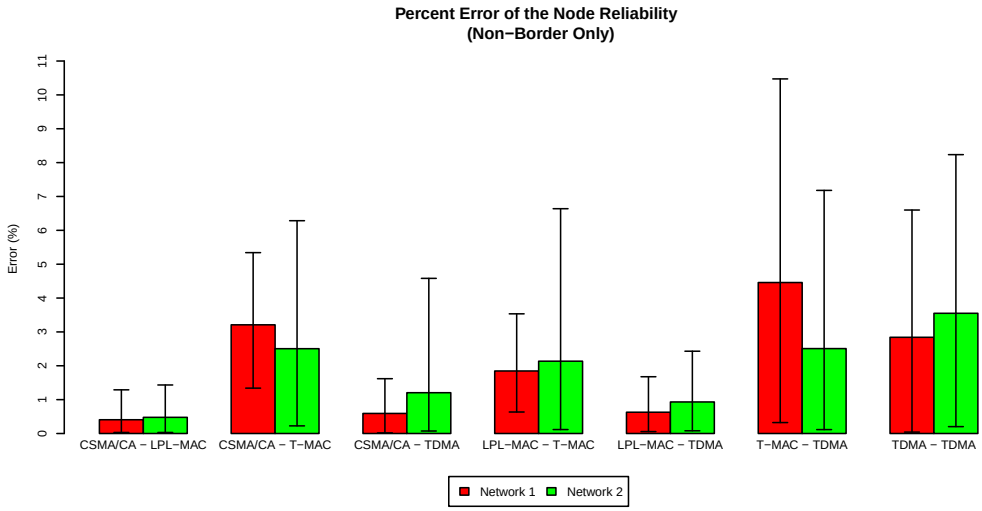


Figure 6.3: Average, 5- and 95%-tile of the percent error of the node reliability when removing 'Non-Border' gateways in the node-to-sink scenario.

the hop counts observed for the 'Same MAC' case were significantly higher. On average there was a 10% difference with a 95-percentile of 23%. In addition, the variation in node hop count between different test runs was also much higher for the 'Same MAC' case than for the 'No Interference' case. This thus shows that, even with no inter-MAC interference or virtual gateways present, the number of T-MAC nodes in the wireless environment has a very significant impact on the hop count of the network. Given that this behaviour is not taken into account by the prediction algorithm, these scalability issues of the T-MAC protocol are also reflected in the prediction errors shown in figure 6.2.

Finally, figure 6.3 shows the percentile error of the node reliability for all combinations of MAC protocols. This figure shows that although for the node-to-sink scenario the prediction error is somewhat higher than it was for the random-flows scenario, it is still within the "5% average, 10% 95-percentile" bounds set in section 5.1.5. The only exception is the T-MAC - TDMA case and even then, the 95-percentile error is only 10.5%.

Although in general the prediction error is thus somewhat higher for the node-to-sink scenario than it is for the random-flows scenario, in most cases the observed error is still quite low and within the bounds used in chapter 5 to tune the selection algorithm. Despite this there are a few cases where the prediction error is significantly higher than these bounds and this will have to be taken into account by the selection algorithm.

6.2.2 Removal of redundant gateways

The accuracy of the prediction algorithm when removing redundant gateways is evaluated in exactly the same manner for the node-to-sink scenario discussed here as it was for the random-flows scenario. As discussed in section 4.3.2, two additional parameters are considered in addition to the set of MAC protocols used: the 'initial number of gateways' and the 'number of gateways removed'. As with the tests discussed in that section, the

'initial number of gateways' is varied from 50 to 35 gateways in steps of 5 gateways and the 'number of gateways removed' is varied from 1 to 10 gateways. The accuracy of the prediction algorithm is discussed separately for each metric below.

6.2.2.1 Duty Cycle

The prediction error for the node duty cycle metric when removing redundant gateways in the node-to-sink scenario is shown in figures 6.4 to 6.10. These figures show that the prediction error 'behaves' in mostly the same manner for the node-to-sink scenario as it did for the random-flows scenario discussed in section 4.3.2.1. For the CSMA/CA MAC protocol the prediction error is, unsurprisingly, always zero while for the other MAC protocol the prediction error varies slightly with both the initial number and the number of removed gateways but is still very small overall (less than 3%). When LPL-MAC is combined with either T-MAC or with TDMA, the T-MAC and TDMA MAC protocols behave in exactly the same way as when they are combined with CSMA/CA. For the T-MAC network the error is still very small (less than 3% overall) but for the TDMA network, the prediction error is slightly higher when combined with LPL-MAC than when combined with CSMA/CA: the average error can rise to around 2% while the 95-percentile error reaches nearly 6%. In both the LPL-MAC - T-MAC and LPL-MAC - TDMA case the prediction error for the LPL-MAC network is noticeably higher than that of either the T-MAC or the TDMA network. The prediction error is still quite low when only a few virtual gateways are removed but when this number is increased, the average error can reach 5.2% and 3% for respectively the LPL-MAC - T-MAC and LPL-MAC - TDMA case while the 95-percentile error can reach respectively 13% and 9%.

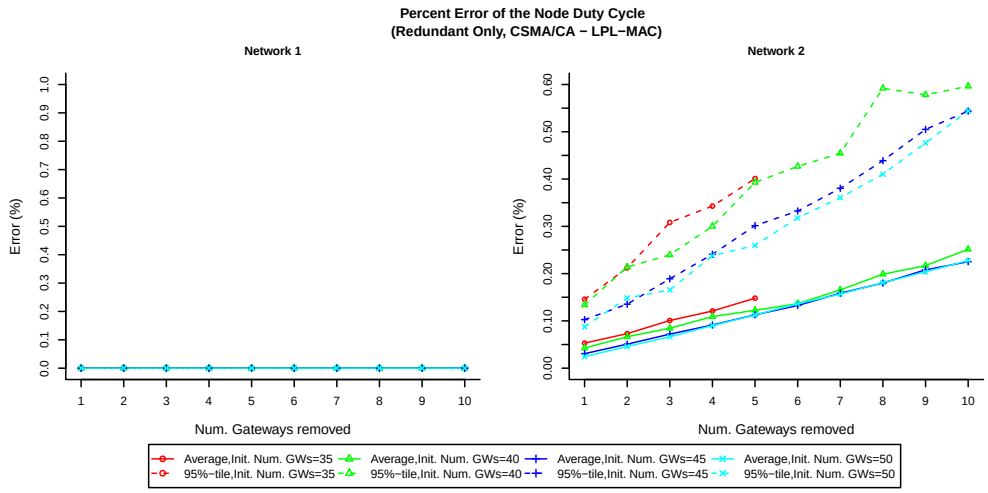


Figure 6.4: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and LPL-MAC protocols.

When the T-MAC protocol is combined with TDMA the prediction error, again, 'behaves' very similarly to the random-flows scenario. It increases both with the number of removed gateways and when the number of initial gateways is reduced. As before, the prediction error is still quite small when only a few virtual gateways are removed but rises rapidly

when more virtual gateways are removed at the same time. For the T-MAC network the average prediction error can reach as high as 4% while the 95-percentile error can top 16%. For the TDMA network the error is a little bit higher and also rises more rapidly. In the worst case, the average prediction error can become as high as 6.2% while the 95-percentile error can top 20%.

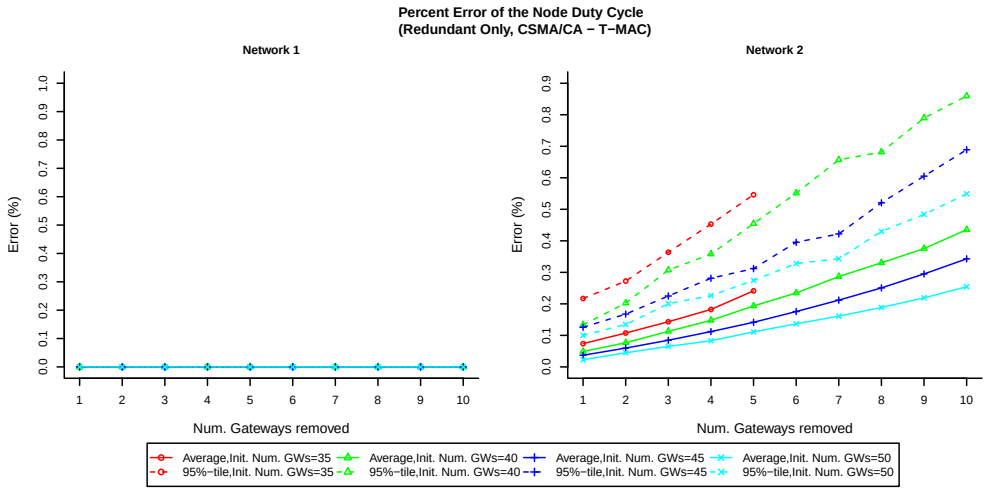


Figure 6.5: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and T-MAC protocols.

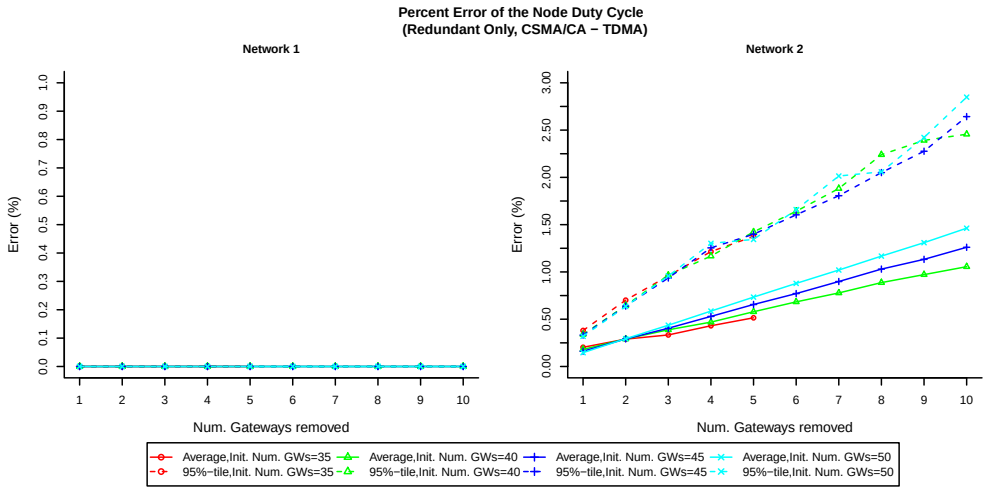


Figure 6.6: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and TDMA MAC protocols.

For the TDMA - TDMA case, the prediction error displays the same 'erratic' behaviour as in the random-flows scenario. Although the prediction error can clearly be seen to rise with the number of 'removed gateways' it also varies significantly with even minor

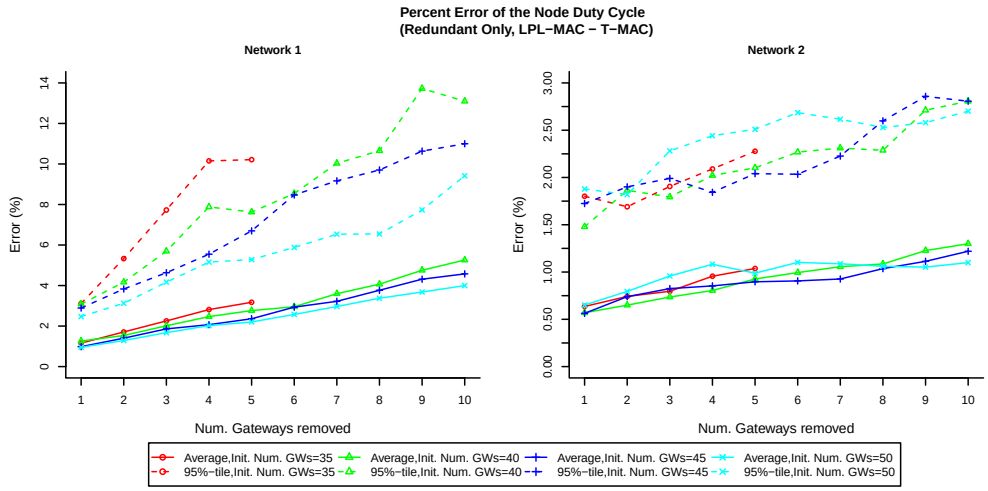


Figure 6.7: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using the LPL-MAC and T-MAC protocols.

changes to either the 'number of removed gateways' or the 'initial number of gateways' parameter. In addition, the prediction error also seems to rise when the 'initial number of gateways' parameter is reduced but this effect is less clear. When the magnitude of the prediction error is considered it is clear that the prediction error is also somewhat higher for the TDMA - TDMA case than it is for the other combinations of MAC protocols with the average error rising to around 8.5% and the 95%-tile error reaching nearly 15%. This somewhat more 'erratic' behaviour is most likely caused by the fact that, when both networks use a TDMA MAC protocol, the effect of the interference between these MAC protocols on the node duty cycle will for a significant portion depend on the specific timings and thus the slot allocation used by the nodes. As with the random-flows scenario, the fluctuations observed in figure 6.10 are thus most likely caused by the fact that these slot allocations depend on the exact number of nodes adhering to the timing of each MAC protocol and thus also on the specific number of virtual gateways used.

The results shown in figures 6.4 to 6.10 thus show that, apart from the TDMA - TDMA case, there is very little difference in the 'behaviour' of the prediction error between the random-flows and the node-to-sink scenario. Both the initial number and the number of removed gateways have a far greater influence on the prediction error than the specific application being used. This also means that, as for the random-flows scenario the 'number of removed gateways' can be used as a tuning parameter by the selection algorithm to control the size of the prediction error.

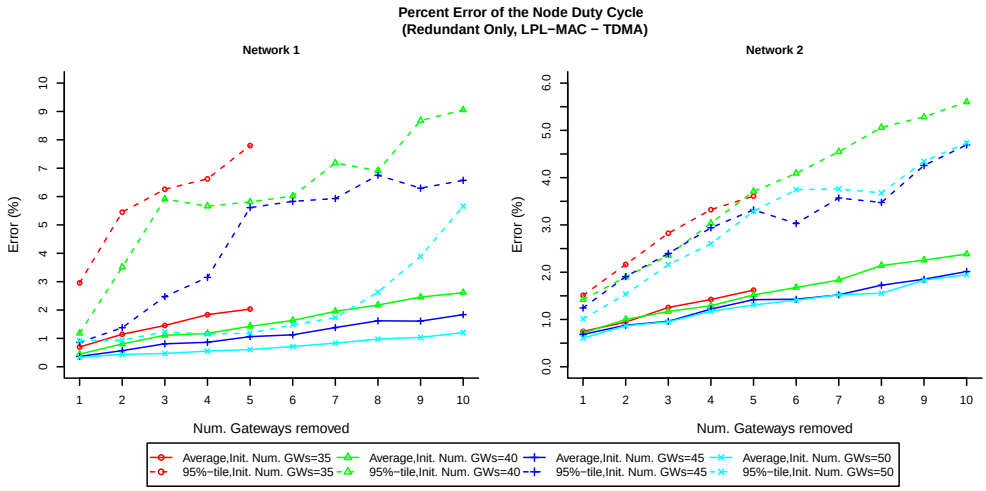


Figure 6.8: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using the LPL-MAC and TDMA MAC protocols.

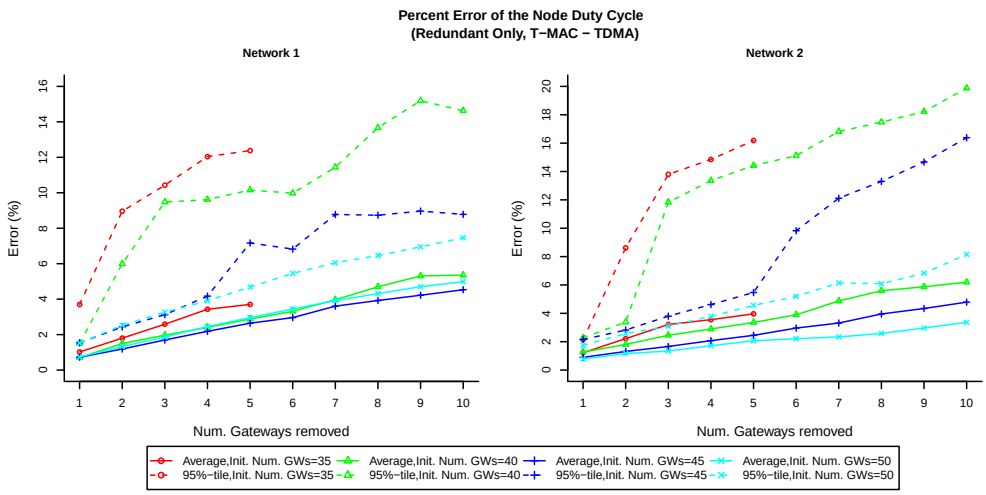


Figure 6.9: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using the T-MAC and TDMA MAC protocols.

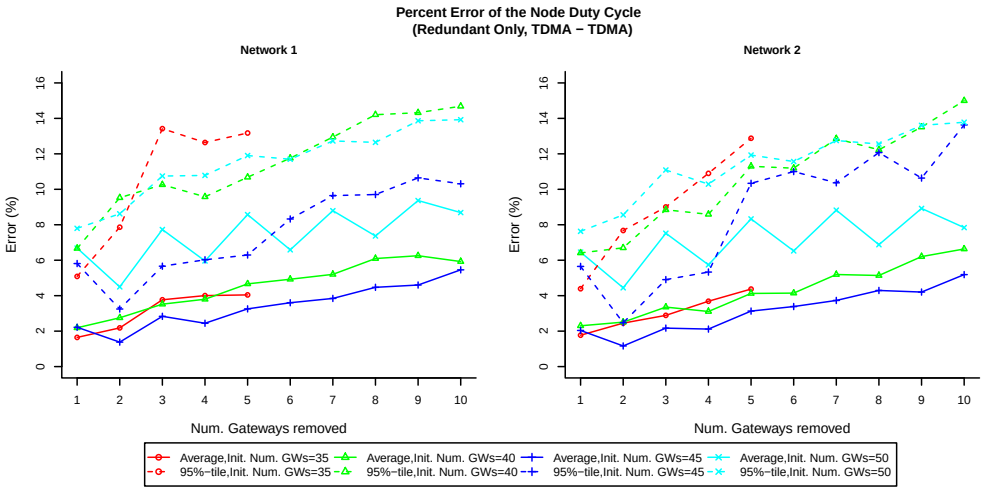


Figure 6.10: Average and 95%-tile of the node duty cycle prediction error when removing redundant gateways in the node-to-sink scenario when using two TDMA MAC protocols.

6.2.2.2 Hop Count

The prediction error for the node hop count metric when removing redundant gateways in the node-to-sink scenario is shown in figures 6.11 to 6.17. With the exception of the TDMA - TDMA case, the 'behaviour' of the prediction error in the node-to-sink scenario is remarkably similar to the behaviour encountered in the random-flows scenario and as with the random-flows scenario the prediction error behaves differently depending on whether or not the T-MAC protocol is used.

In all cases where the T-MAC protocol is used, the prediction error of the T-MAC network rises only slightly with the number of removed virtual gateways. For the other network this effect is less visible depending on the MAC protocol used. As with the random-flows scenario (see section 4.3.2.2), this rise in prediction error is most likely due to the fact that the reduction in interference resulting from the removal of one or more gateways may cause the routing protocols of the network to establish slightly different paths even if no paths were broken by the removal of these gateways. There is however a significant difference in the size of the prediction error between the random-flows and the node-to-sink scenario. In the random-flows scenario the prediction error is in general relatively low (averages between 2% and 3%, 95-percentiles between 4% and 10%). In the node-to-sink scenario however the prediction error is significantly higher. Depending on the set of MAC protocols used and the number of removed gateways, the average prediction error of the T-MAC network hovers somewhere between 5% and 8.5% while the 95-percentile hovers between 14% and 22%. This higher error is mostly due to the fact that, as discussed in section 6.2.1, the T-MAC protocol suffers from scalability issues which cause the path length to increase dramatically with the number of T-MAC nodes deployed in the wireless environment. In addition, increasing the number of T-MAC nodes in the wireless environment also causes the *variation* in hop count between different test runs to increase substantially, which indicates that the specific timings of the T-MAC protocol also have a significant impact on the length of the used routing paths. (While this is also

true for the random-flows scenario, this effect is more visible in the node-to-sink scenario since in this scenario the traffic is more concentrated around the sink nodes.) Since neither of these factors are taken into account by the prediction algorithm, these variations are reflected in the prediction error. In addition, it should also be noted that the use of the T-MAC protocol also causes the prediction error for the other network involved to rise dramatically. This can be attributed to the fact that when "sink sharing" is enabled sensor readings from one network are also sent to the sink of the other network and that the prediction error is therefore influenced by the MAC protocols of both networks. Even so, the prediction for these networks is still mostly within the "5% average, 10% 95-percentile" bounds set in section 5.1.5.

When the T-MAC protocol is *not* used there is some variation in the prediction error between the different values of the 'initial number of gateways' and 'number of gateways removed' parameter, but these variations are relatively minor. Moreover, even with these variations the prediction error is generally very low with the average error hovering around 1% and the 95-percentile error remaining below 4% in most cases and never rising to more than 7%. The only exception is the TDMA - TDMA case. In this case the behaviour of the prediction error is much more erratic, with the average and especially the 95-percentile varying significantly even when only minor changes are made to the initial number of gateways or the number of removed gateways. Despite this, no clear upwards or downwards trend with either of these two parameters can be discerned.

This behaviour is most likely caused by the fact that for the TDMA - TDMA case the *link reliability* depends on the specific timings and slot allocation used by the TDMA MAC protocols which in turn depend on the specific number of virtual gateways used. As discussed in section 4.3.2.3, this can cause the link reliability to vary significantly with only moderate changes to the initial number or number of removed gateways. Moreover, as discussed in section 2.2.1, the 'ETX' route metric used by both networks automatically makes a tradeoff between reliability and hop count and given that none of this is taken into account by the prediction algorithm it should come as no surprise that the erratic behaviour of the link reliability results in the variations shown in figure 6.17. It should also be noted that, although suffering from the same variations in the link reliability (see section 4.3.2.2), the TDMA - TDMA case does not exhibit these fluctuations in the prediction error of the node hop count when the random-flows scenario is used instead of the node-to-sink scenario. This is most likely due to the fact that in the node-to-sink scenario most traffic is concentrated around the two sink nodes whereas in the random-flows scenario the traffic is more evenly distributed throughout the wireless environment. Despite these fluctuations, the prediction error is still quite small: the average prediction error is smaller than 2.3% and the 95-percentile is at most 6.7%.

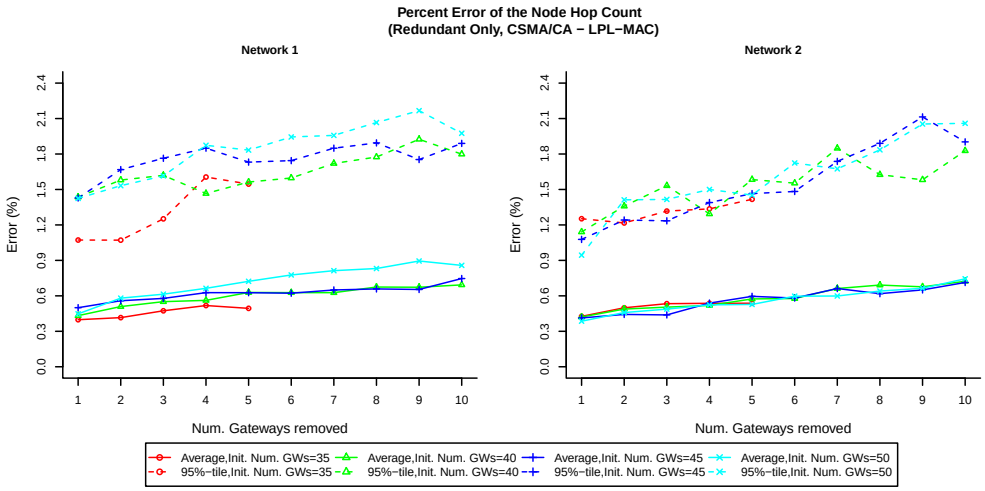


Figure 6.11: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and LPL-MAC protocols.

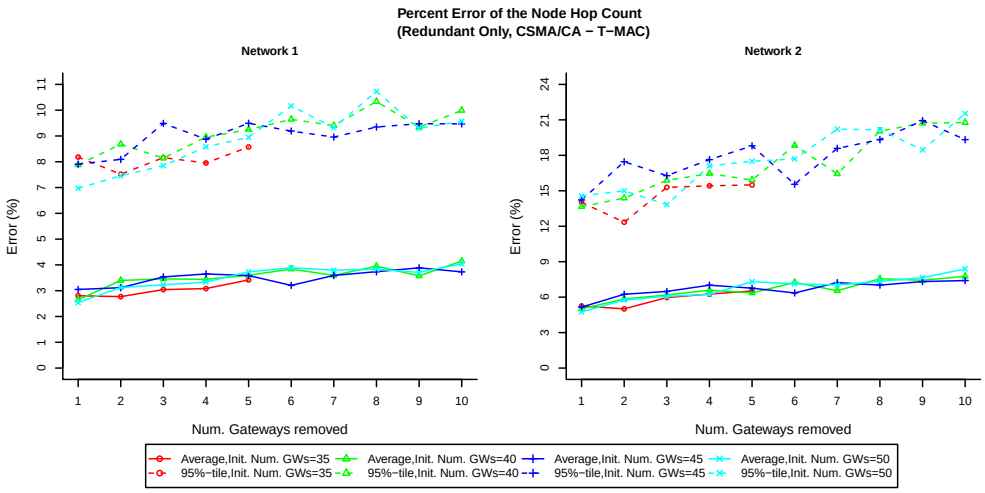


Figure 6.12: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and T-MAC protocols.

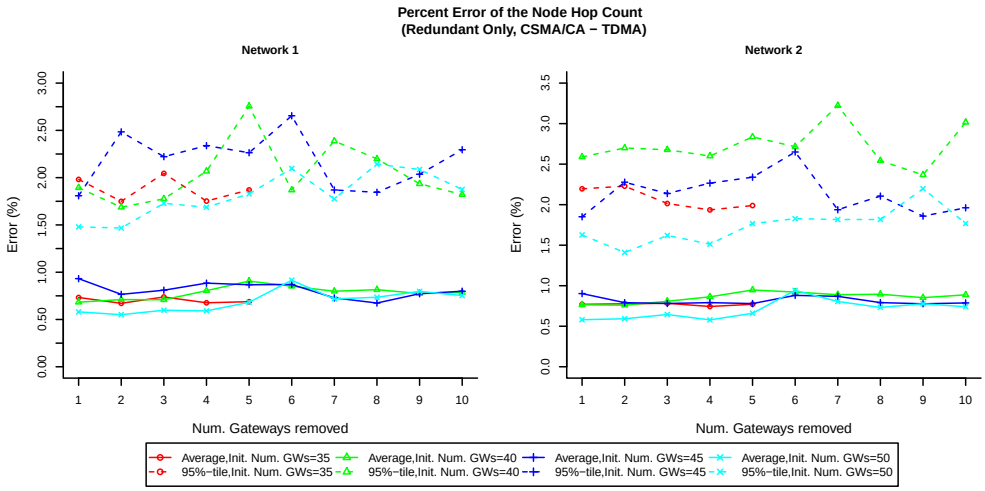


Figure 6.13: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and TDMA MAC protocols.

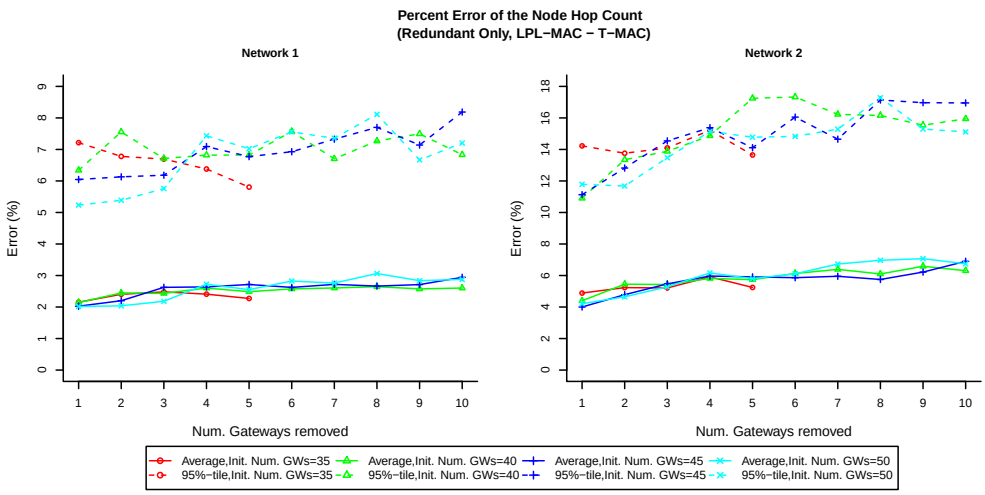


Figure 6.14: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using the LPL-MAC and T-MAC protocols.

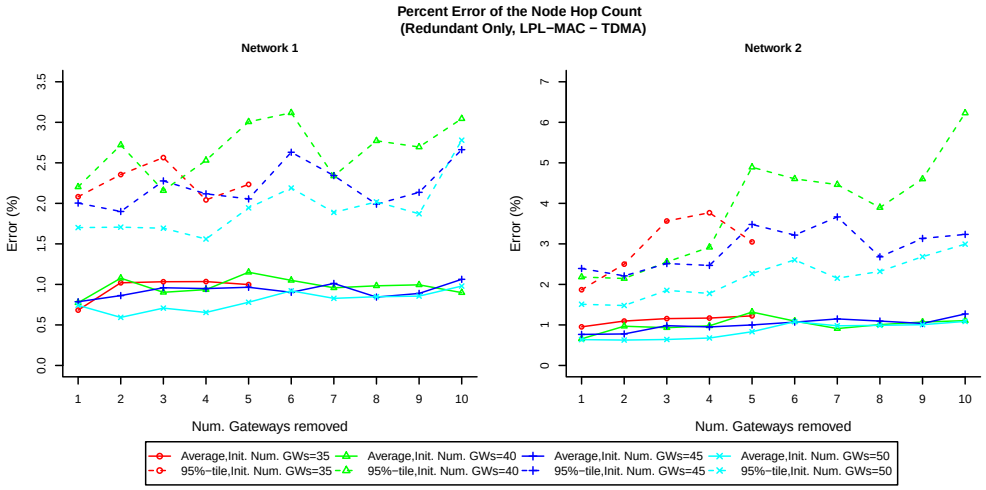


Figure 6.15: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using the LPL-MAC and TDMA MAC protocols.

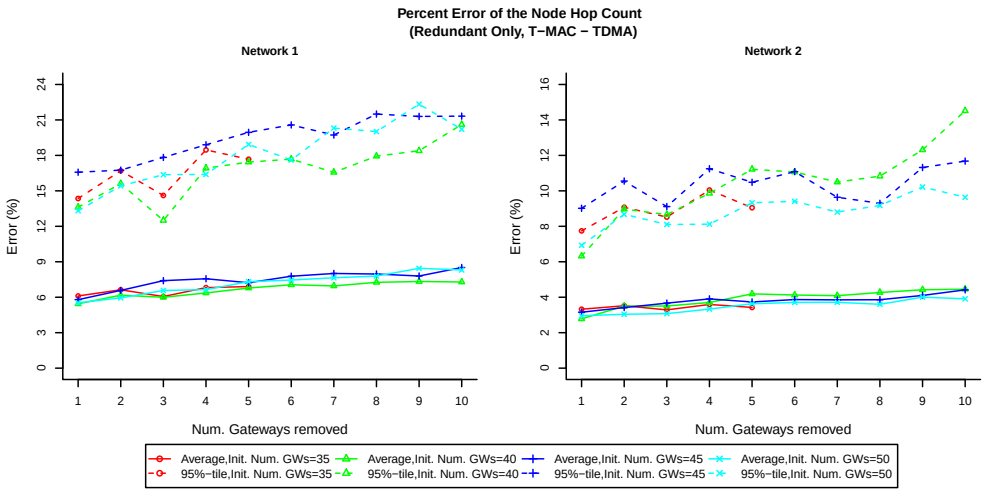


Figure 6.16: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using the T-MAC and TDMA MAC protocols.

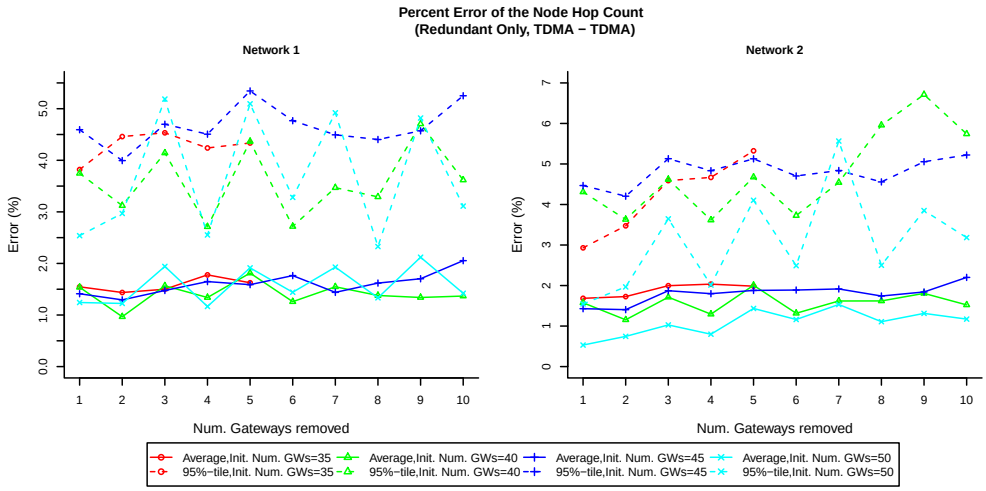


Figure 6.17: Average and 95%-tile of the node hop count prediction error when removing redundant gateways in the node-to-sink scenario when using two TDMA MAC protocols.

6.2.2.3 Reliability

The prediction error for the node reliability metric when removing redundant gateways in the node-to-sink scenario is shown in figures 6.18 to 6.24. These figures show that although there are some similarities in prediction error between the random-flows and node-to-sink scenario, the difference in traffic patterns between these two scenarios has a visible effect on the behaviour of the prediction error.

For the CSMA/CA - LPL-MAC case, the prediction error behaves in mostly the same manner as it did in the random-flows scenario. The prediction error rises with the number of removed and the initial number of gateways. Despite this the prediction error is very low with the average error topping out at 1.2% and the 95%-tile of the error being at most 4.5%. In the CSMA/CA - T-MAC and LPL-MAC - T-MAC cases, the prediction error also behaves similarly to the random-flows scenario except that the number of removed gateways does not affect the prediction error as much as it did in the random-flows scenario. This causes the prediction error to be significantly lower than in the random-flows scenario when a large number of gateways are removed at the same time. In both cases the average error is at most 4.2% while the 95-percentile error tops out at 8.5% and 6% for respectively the CSMA/CA - T-MAC and LPL-MAC - T-MAC case.

When either CSMA/CA or LPL-MAC is combined with the TDMA MAC protocol, the prediction error, again, behaves in much the same manner as it did in the random-flows scenario, except that for the TDMA network, the 95-percentile error shows a bit more fluctuation than in the random-flows scenario. This is most likely due to the fact that in the node-to-sink scenario most traffic is concentrated around the sink nodes and that this increased level of traffic also increases the interference between the MAC protocols (which is not taken into account by the prediction algorithm). Despite this the error is quite low in both cases. The average prediction error tops out at 2.1% and .175% for respectively the CSMA/CA - TDMA and LPL-MAC - TDMA case while the 95-percentile error can

reach respectively 6% and 4.5%.

In the T-MAC - TDMA case, the prediction error behaves quite different in the node-to-sink scenario than it did in the random-flows scenario. In contrast to the random-flows scenario, the number of removed gateways does not affect the average prediction error at all and while the 95-percentile error does seem to rise slightly with the number of removed gateways, this effect is very small. In addition, the 95-percentile error also varies quite a bit more with the initial number of gateways than it did in the random-flows scenario. For the T-MAC network the average error is still quite low (less than 5%), but the 95%-tile error can rise as high as 13%. For the TDMA network, the error is somewhat lower with the average hovering between 2 and 3.5% and most 95%-tile error values below 10%, with one outlier at 14%.

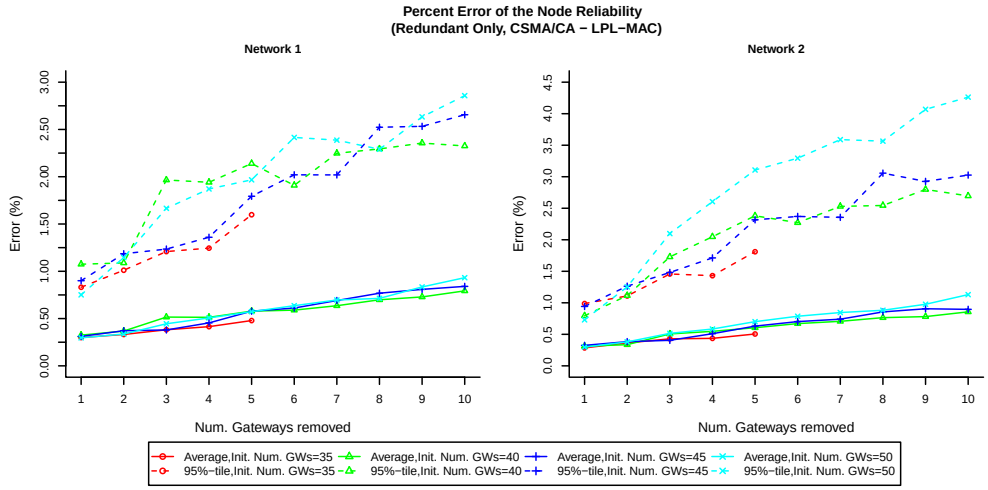


Figure 6.18: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and LPL-MAC protocols.

In the TDMA - TDMA case, the prediction error for the node-to-sink scenario behaves very similarly to the random-flows scenario in the sense that it fluctuates wildly with even the smallest changes to either the 'initial number of gateways' or the 'number of gateways removed'-parameter. As with the random-flows scenario, this is most likely caused by the fact that the *link reliability* depends on the specific timings and slot allocation used by the TDMA MAC protocols which in turn depend on the specific number of virtual gateways used. Despite this the error can still be considered to be relatively small with the average error hovering between 2.5% and 3.5% and the 95%-tile error reaching at most 10%.

Overall, the most significant difference in the 'behaviour' of the prediction error is that the 'number of removed gateways' has a much smaller effect on the prediction error in the node-to-sink scenario than it did in the random-flows scenario. While this does make this parameter less suitable as a 'tuning parameter' for the prediction error, in most cases the prediction error is sufficiently low to prevent this from being an issue for the selection algorithm. The only exception is perhaps the T-MAC network in the T-MAC - TDMA

case, for which the 95-percentile error is already relatively high (10-12%) even when only a few gateways are removed.

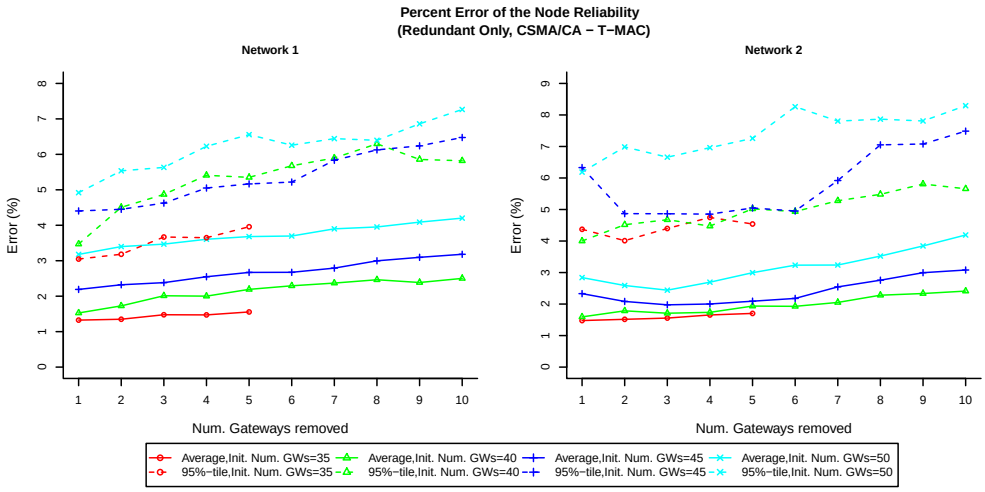


Figure 6.19: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and T-MAC protocols.

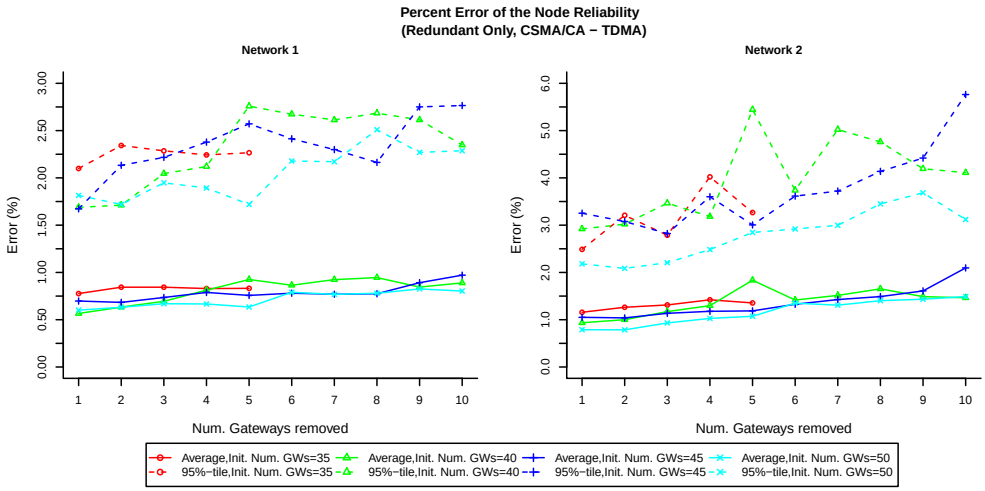


Figure 6.20: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using the CSMA/CA and TDMA MAC protocols.

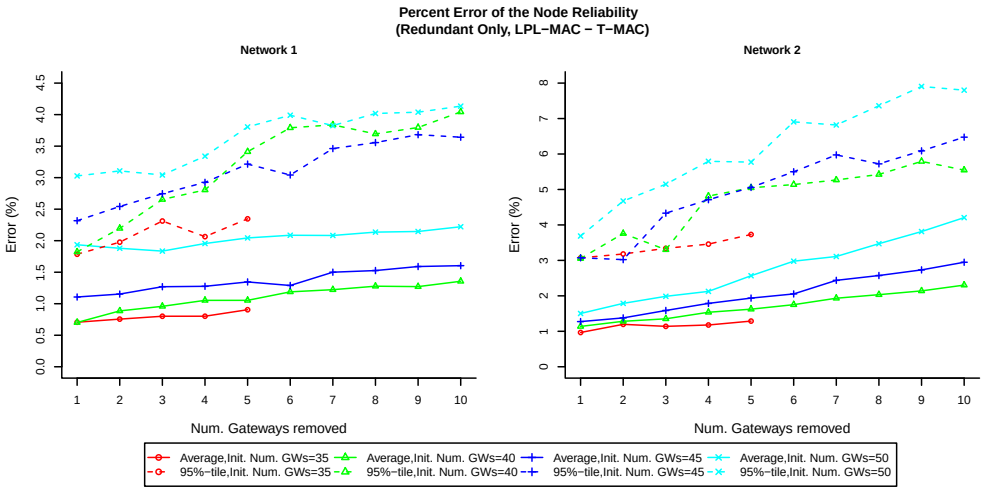


Figure 6.21: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using the LPL-MAC and T-MAC protocols.

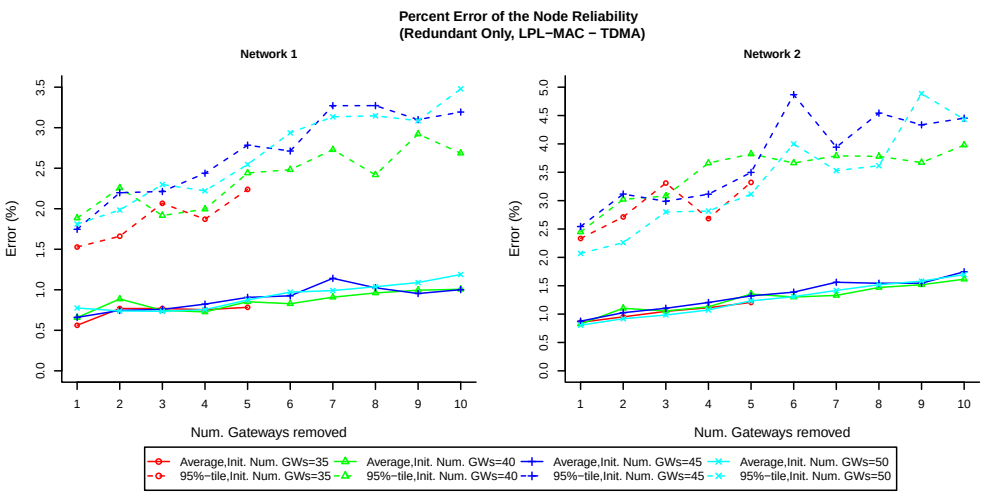


Figure 6.22: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using the LPL-MAC and TDMA MAC protocols.

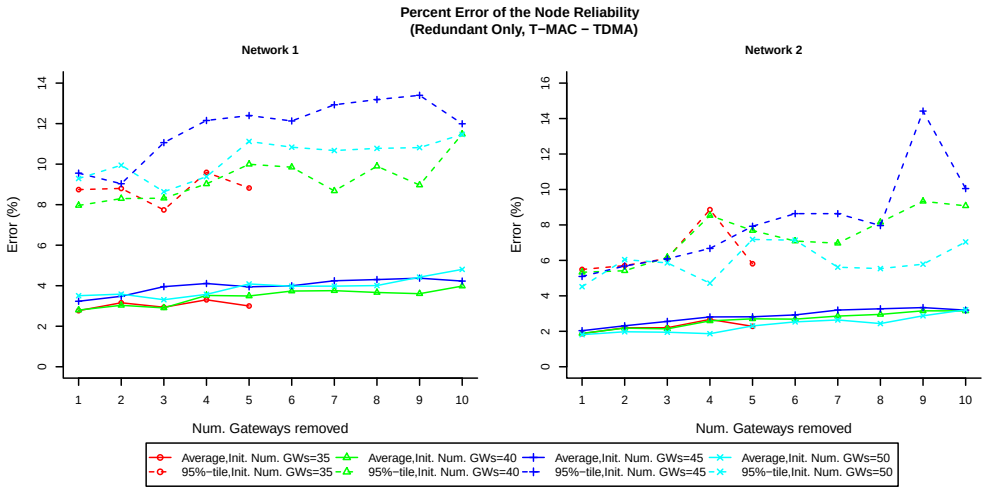


Figure 6.23: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using the T-MAC and TDMA MAC protocols.

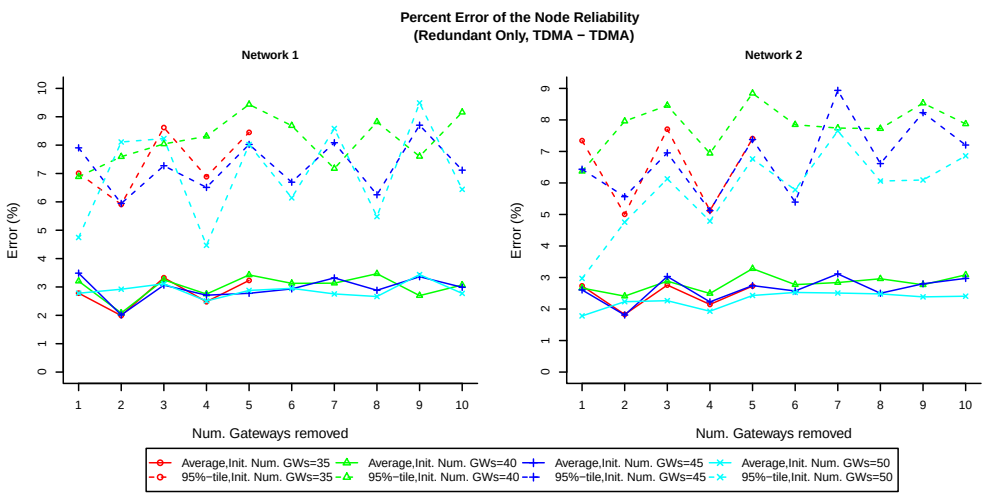


Figure 6.24: Average and 95%-tile of the node reliability prediction error when removing redundant gateways in the node-to-sink scenario when using two TDMA MAC protocols.

6.2.3 Removal of non-redundant gateways

As with the evaluation of the redundant gateways, the accuracy of the prediction algorithm when removing non-redundant gateways is evaluated in largely the same manner for the node-to-sink scenario discussed here as it was for the random-flows scenario. As with the tests discussed in section 4.3.3, the 'initial number of gateways' parameter is varied from 21 to 6 gateways in steps of 5 gateways and the 'number of gateways removed' is varied from 1 to 10 gateways (in steps of 1). For the case where the 'initial number of gateways' is 6, the 'number of gateways removed' is, in contrast to the random-flows scenario, at most 6 (rather than 5). This is because in the node-to-sink scenario the *minimum* gateway configuration is empty and it is therefore allowed to remove *all* gateways from the wireless environment. The accuracy of the prediction algorithm is discussed separately for each metric below.

6.2.3.1 Duty Cycle

The prediction error for the node duty cycle metric when removing non-redundant gateways in the node-to-sink scenario is shown in figures 6.25 to 6.31. These figures show that the prediction error 'behaves' in much the same way in the node-to-sink scenario as it did in the random-flows scenario. As with the random-flows scenario, the prediction error increases with the number of removed gateways and, depending on the specific combination of MAC protocols used, either increases or decreases with the initial number of gateways.

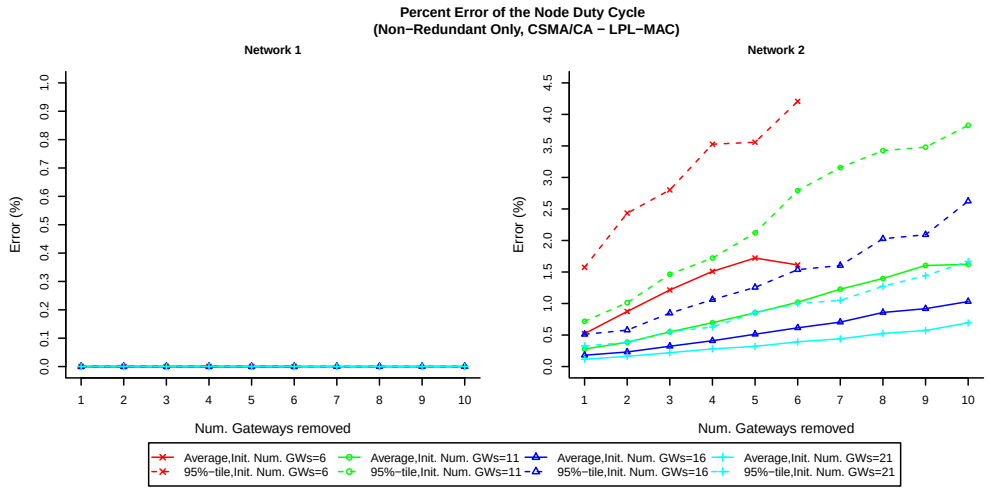


Figure 6.25: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and LPL-MAC protocols.

This behaviour can be explained by the fact that removing a non-redundant gateway affects the node duty cycle through two separate mechanisms, each of which may have a stronger (or weaker) effect on the duty cycle depending on the specific set of MAC protocols used (see section 4.3.3.1 for more information). In most cases, the prediction

error reacts in the same manner to the reduction of the initial number of gateways but there are some minor differences. In the LPL-MAC - T-MAC case for instance, the prediction error of both the LPL-MAC and the T-MAC network also rises steeply with the number of removed gateways when the 'initial number of gateways' parameter is set to 6. In the random-flows scenario this was only the case for the T-MAC network. These minor differences indicate that the traffic patterns in the wireless environment and thus the application scenario used, do have a minor effect on the behaviour of the prediction error.

When the magnitude of the prediction errors is examined it quickly becomes clear that, as in the random-flows scenario, the prediction error rises quite rapidly with the number of removed gateways and can, depending on the specific combination of MAC protocol used, become quite significant when a large number of gateways are removed at the same time. For networks using the CSMA/CA MAC protocol the prediction error is, unsurprisingly always zero. For networks using the LPL-MAC protocol the average error is in most cases quite small (less than 3%) except when LPL-MAC is combined with T-MAC. In that case the average prediction can reach as high as 10% but is generally still lower than 4%. The 95-percentile error is in most cases well below 10% but can in some cases reach as high as 13%. For networks using the T-MAC protocol, the average prediction error can reach as high as 6% but is in most cases well below the 5% mark. The 95-percentile prediction error for the T-MAC error is relatively low when T-MAC is combined with either CSMA/CA or LPL-MAC (at most 8%) but when T-MAC is combined with TDMA, the 95-percentile error can reach as high as 16%. The prediction error is even higher for networks using the TDMA network. The average and 95-percentile error rise, in all three cases, above respectively 10% and 20%. Moreover, the average prediction error can become as high as 14% while the 95-percentile error can rise to around 27%.

Although the prediction error can thus become quite high it should be noted that for most combinations of MAC protocols the maximum errors encountered in the node-to-sink scenario are still somewhat lower than those recorded for the random-flows scenario. Moreover, like in the random-flows scenario, the prediction error is in most cases also quite small when only a few gateways are removed. The only exceptions are the T-MAC - TDMA and TDMA - TDMA cases where the 95-percentile error is already quite large (respectively 8% and 12%) even when only a single gateway removed. Despite the fact that the selection algorithm will therefore have to cope with the fact that the prediction error will sometimes be higher than expected, the 'number of removed gateways'-parameter can still be used to 'tune' the size of the prediction error.

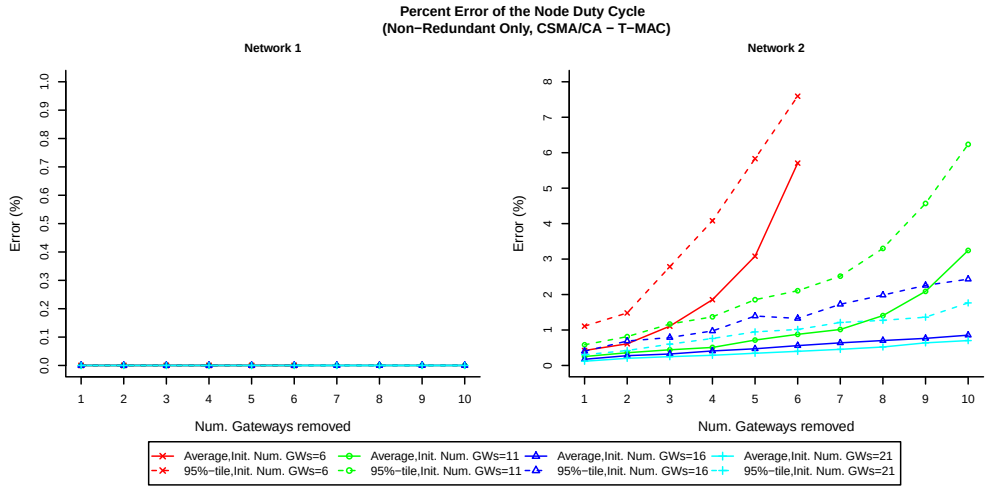


Figure 6.26: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and T-MAC protocols.

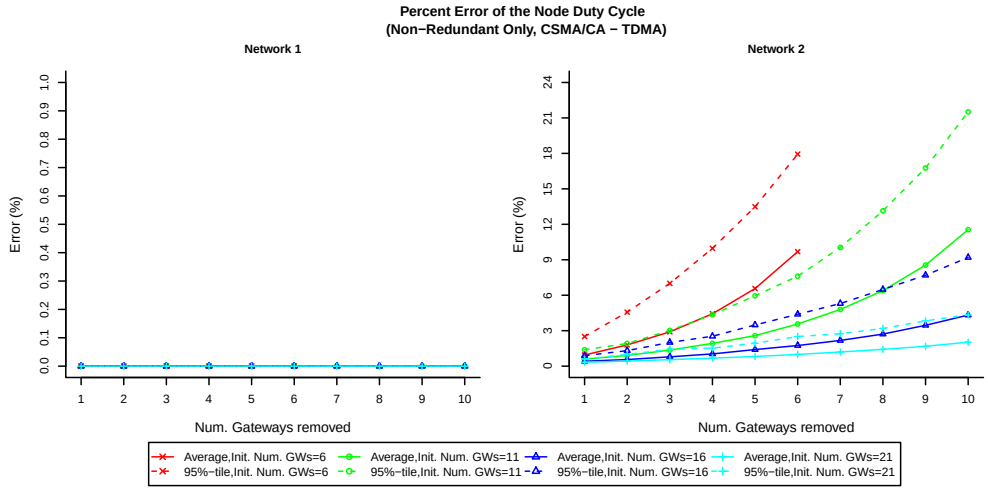


Figure 6.27: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and TDMA MAC protocols.

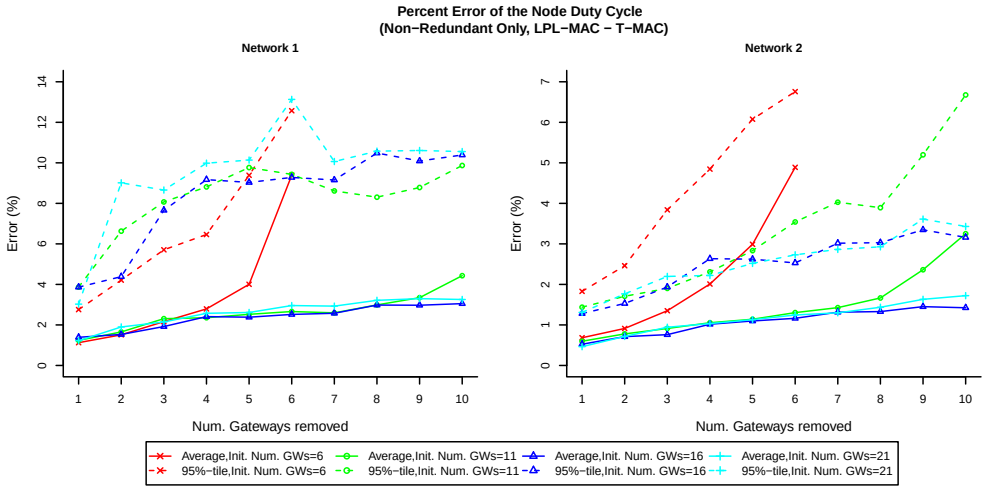


Figure 6.28: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using the LPL-MAC and T-MAC protocols.

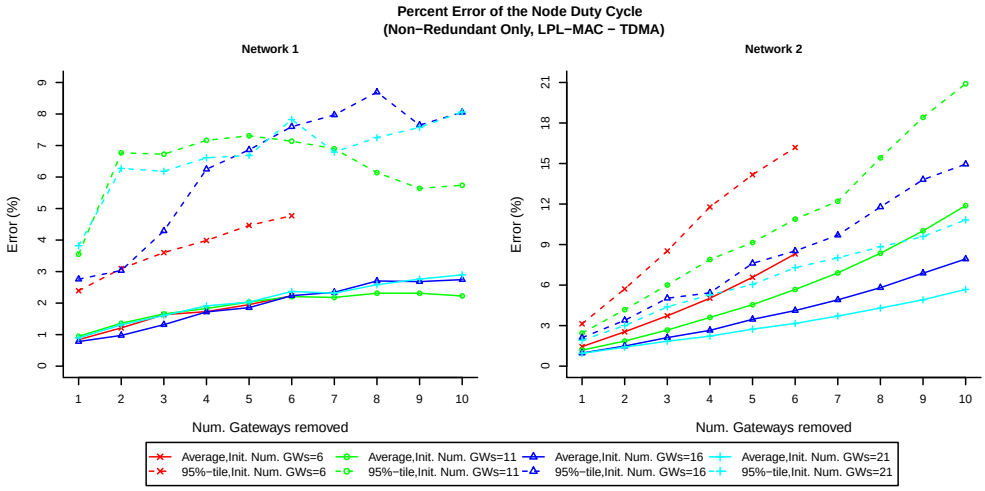


Figure 6.29: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using the LPL-MAC and TDMA MAC protocols.

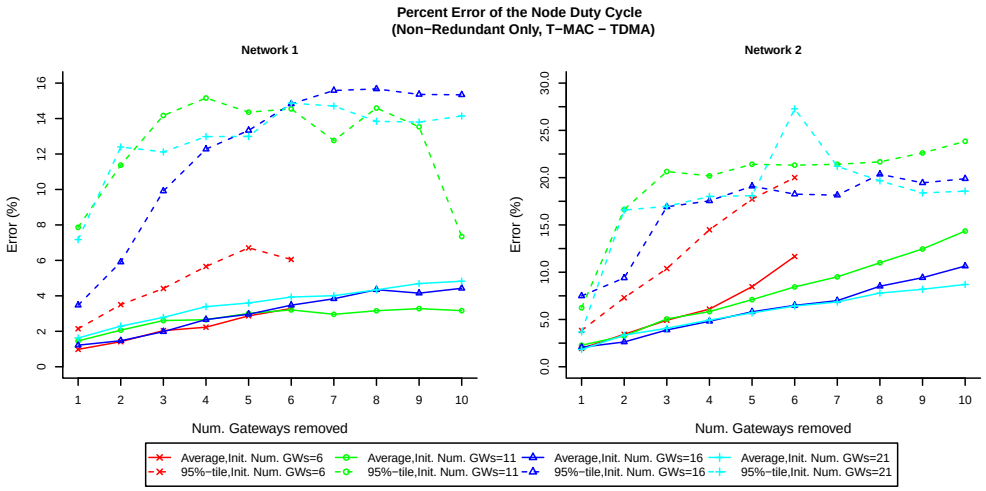


Figure 6.30: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using the T-MAC and TDMA MAC protocols.

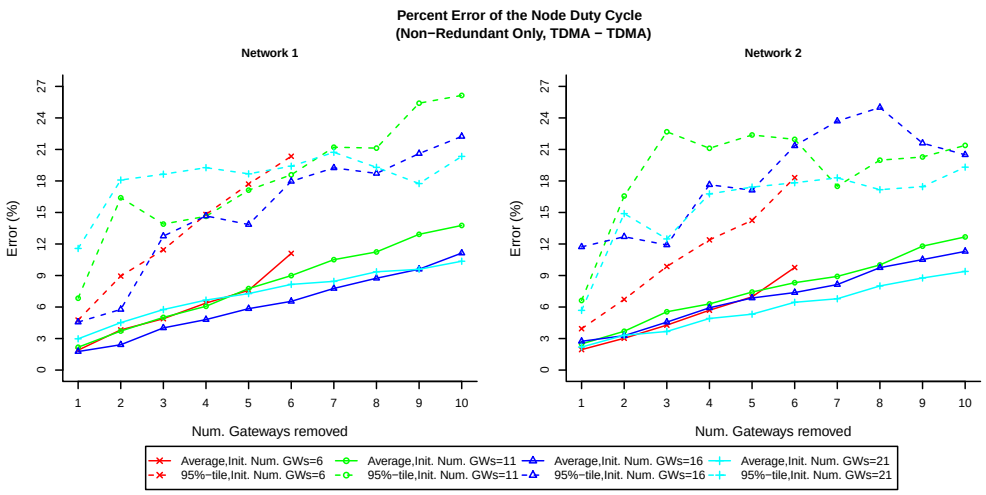


Figure 6.31: Average and 95%-tile of the node duty cycle prediction error when removing non-redundant gateways in the node-to-sink scenario when using two TDMA MAC protocols.

6.2.3.2 Hop Count

The prediction error for the node hop count metric when removing non-redundant gateways in the node-to-sink scenario is shown in figures 6.32 to 6.38. As is the case when removing redundant gateways, the prediction error reacts somewhat differently to changes in the number of initial and removed gateways depending on whether or not the T-MAC protocol is used.

When the T-MAC protocol is *not* used, the prediction error behaves in broadly the same manner as it did in the random-flows scenario. When the number of initial gateways is high (16 or 21), the prediction error rises with the number of removed gateways while for lower values of the 'initial number of gateways'-parameter (11 or 6) the prediction error first rises with the number of removed gateways but then drops again when the number of removed gateways is increased even further. It should be noted however that, depending on the specific set of MAC protocols used, this effect may be less visible in the node-to-sink scenario than it was in the random-flows scenario. As with the random-flows scenario this effect is most likely caused by the fact that when a large number of gateways is removed from a small initial configuration, only very few (if any), gateways will remain in the final configuration. This causes the route topology to become increasingly similar to the *minimum* configuration. Since, as discussed in section 4.3.3.2, the replacement policies rely on, amongst others, the topology information recorded for the minimum configuration to replace any broken paths is should come as no surprise that the prediction error will be lower as a result.

When one of the networks uses the T-MAC protocol however, the prediction error behaves somewhat differently than it did in the random-flows scenario. For the T-MAC network itself, the prediction error rises when the initial number of gateways is reduced. It also rises with the number of removed gateways but this effect is mostly visible when the initial number of gateways is large (16 or 21). The prediction error does drop when 6 gateways are removed from an initial configuration of 6 gateways but, as discussed in section 4.3.3.2, this is merely the result of the final configuration being equal to the minimum configuration. For the network accompanying the T-MAC network however, the prediction error only rises when the initial number of gateways is reduced, it does not rise (or fall) with the number of removed gateways. This behaviour is especially visible when considering the average prediction error. For the 95-percentile error there is a bit more variation but even then, no clear upwards or downwards trend can be discerned.

When the magnitude of the prediction error is considered, it is immediately clear that in the node-to-sinks scenario, the prediction error is in most cases significantly lower than it is for the random-flows scenario. The average prediction is usually less than 3% and the 95-percentile error is less than 9% (except when TDMA is used in which case it can rise to around 12%). The only exception is when the T-MAC protocol is used. In that case the prediction error of the network accompanying the T-MAC network is still quite low, but the average prediction error of the T-MAC network can rise to around 11% while the 95-percentile error can top 25%.

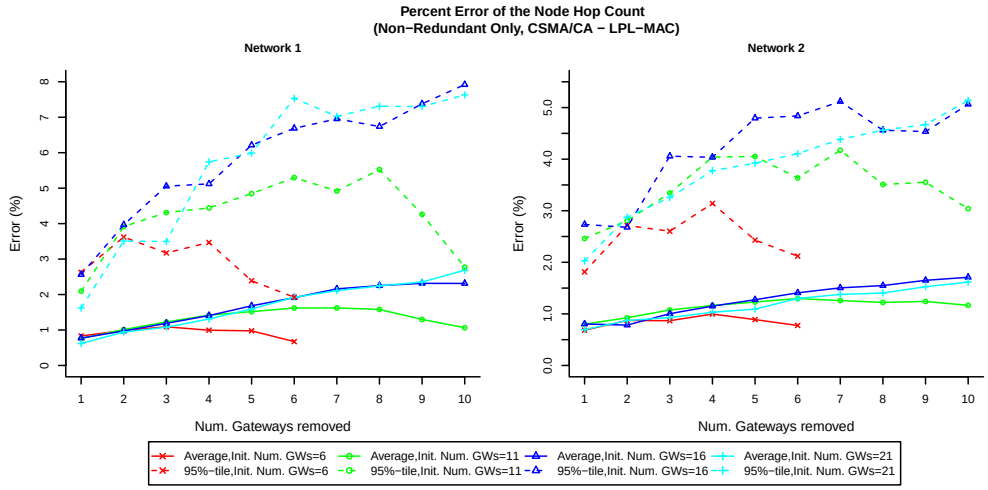


Figure 6.32: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and LPL-MAC protocols.

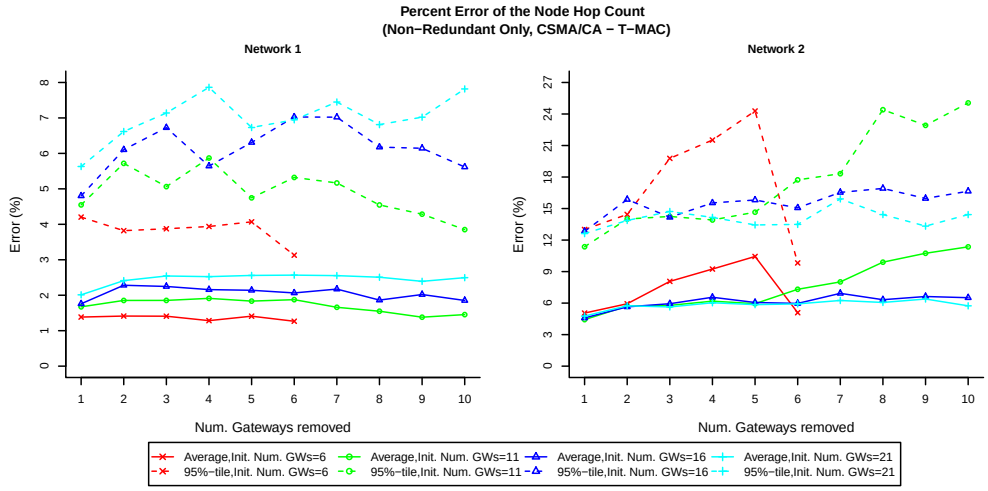


Figure 6.33: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and T-MAC protocols.

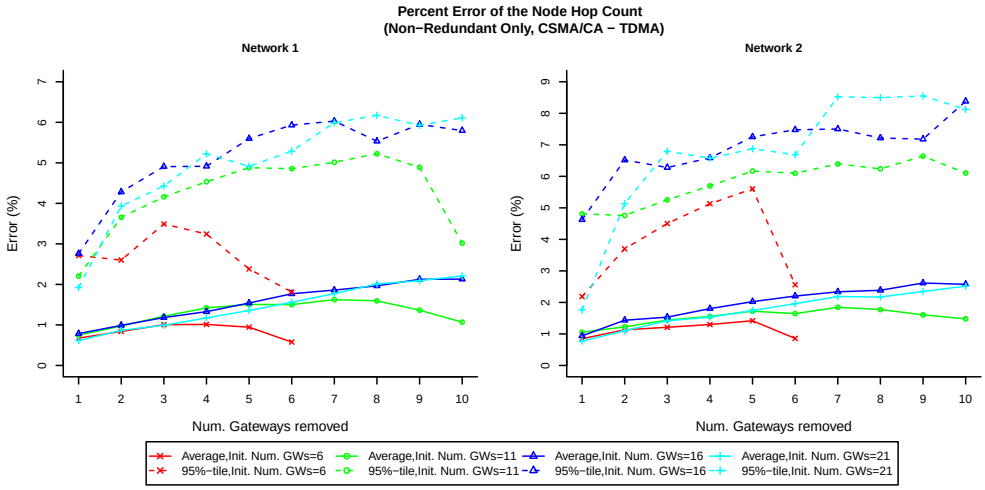


Figure 6.34: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and TDMA MAC protocols.

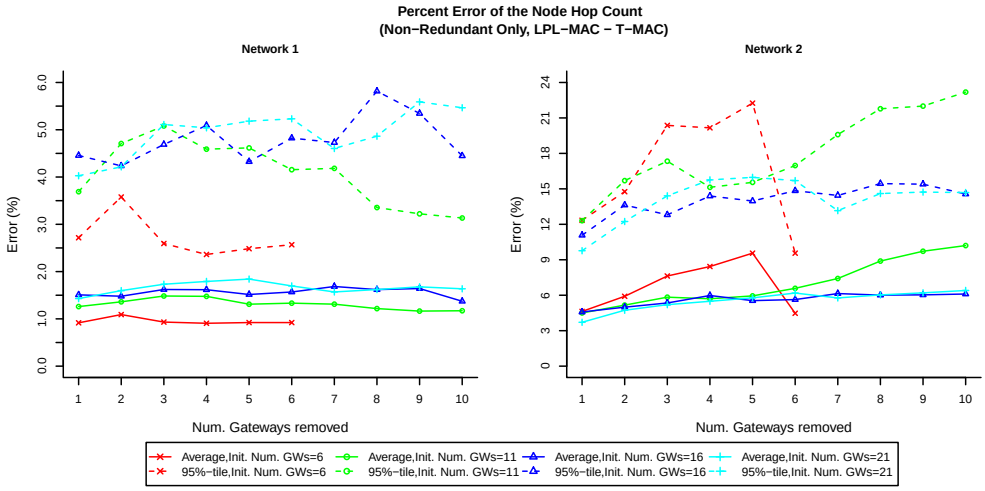


Figure 6.35: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using the LPL-MAC and T-MAC protocols.

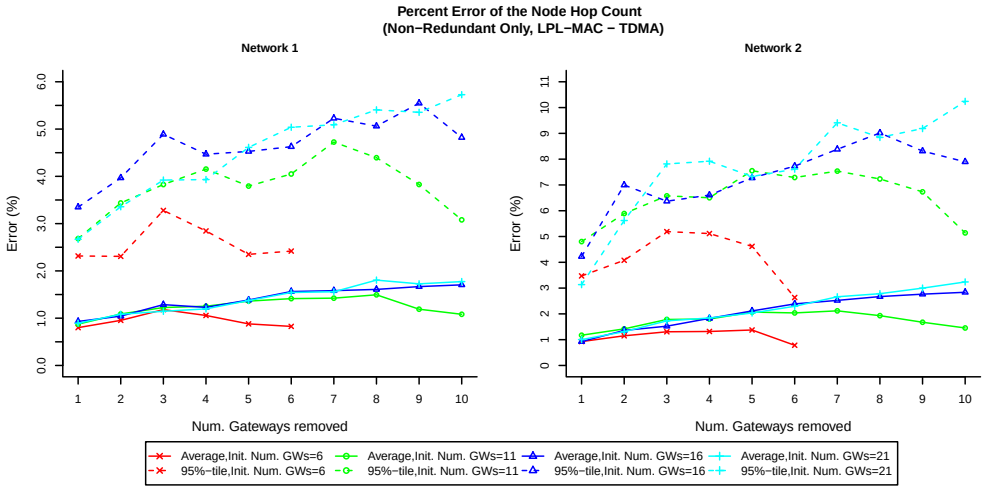


Figure 6.36: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using the LPL-MAC and TDMA MAC protocols.

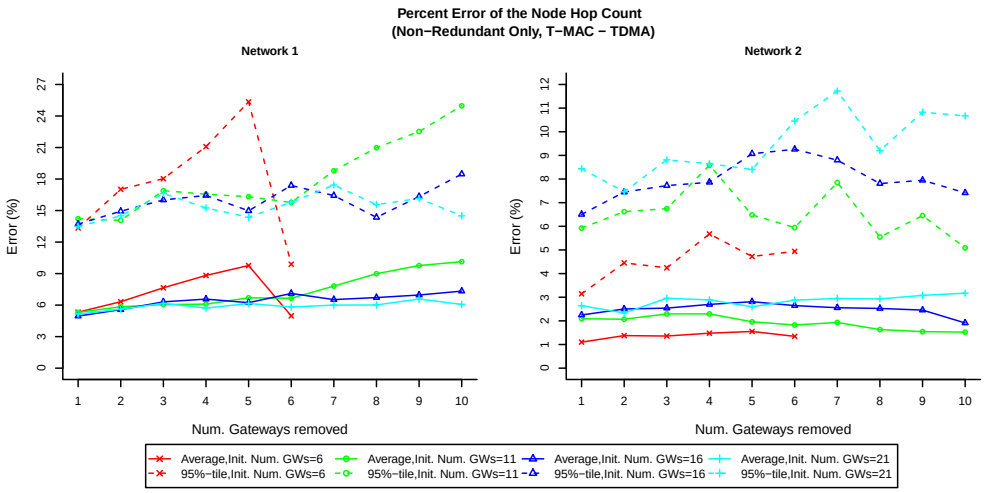


Figure 6.37: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using the T-MAC and TDMA MAC protocols.

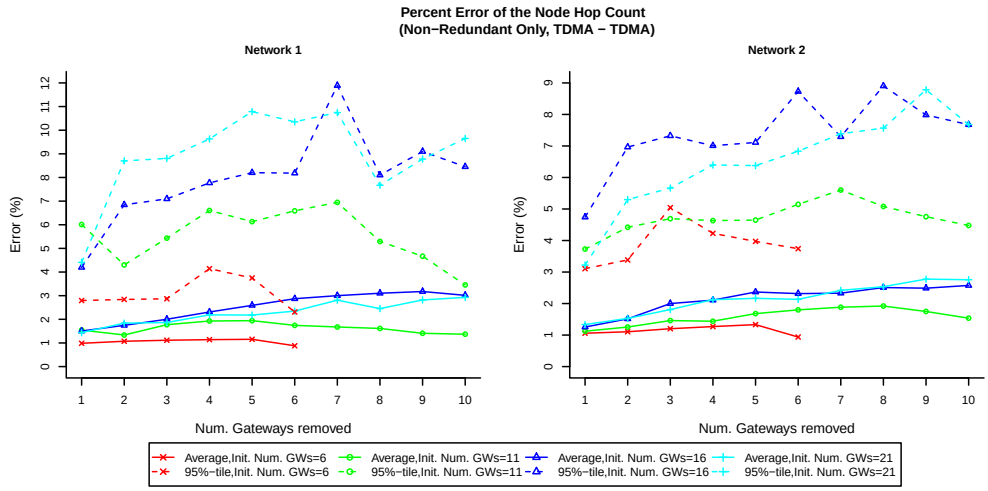


Figure 6.38: Average and 95%-tile of the node hop count prediction error when removing non-redundant gateways in the node-to-sink scenario when using two TDMA MAC protocols.

6.2.3.3 Reliability

The prediction error for the node reliability metric when removing non-redundant gateways in the node-to-sink scenario is shown in figures 6.39 to 6.45. These figures show that, as with the random-flows scenario the prediction error rises with the number of removed gateways and that for most (combinations of) MAC protocols, it also rises when the initial number of gateways is reduced. For those cases where it does not, the prediction error is unaffected by the initial number of gateways. As discussed in section 4.3.3.3 this behaviour is most likely caused by the fact that, in contrast to when redundant gateways are removed, paths broken by the removal of a non-redundant gateway cannot be repaired by the local repair mechanism. Although the 'Equivalent Flow' path replacement policy introduced in section 6.1 allows the prediction algorithm to cope with the fact that the removal of a virtual gateway may cause data to be sent to a different sink, it should also be noted that, as for the random-flows scenario, the prediction algorithm still relies on relatively 'general' assumptions to create a set of plausible replacement paths when a flows is broken. Given that the routing protocol is moreover not bound by the predictions of the prediction algorithm, this also incurs a certain error in the predicted reliability.

Figures 6.39 to 6.45 also show that, in contrast to the random-flows scenario, the networks are only minimally affected by the MAC protocol used by the other network. When the T-MAC protocol is *not* used, the prediction error observed for a network using a specific MAC protocol behaves in largely the same way, regardless of the MAC protocol used by the other network. When for instance CSMA/CA is used by one network, the prediction error of that network when combined with the LPL-MAC network is very similar to the behaviour of the prediction error when the CSMA/CA network is instead combined with TDMA. Moreover, an analogous observation can be made for the LPL-MAC and TDMA MAC protocols. The same is also true for the T-MAC protocol except that when CSMA/CA or TDMA are combined with T-MAC, the prediction error for these networks is less sensitive to changes in the initial number of gateways than when they are combined

with a different MAC protocol. This 'larger independence' between the networks is most likely caused by the fact that in the node-to-sink scenario a smaller portion of a network's traffic crosses the network boundary and that, as a result, the node reliability of one network is not influenced as much by the MAC protocol of the other network as in the random-flows scenario. The only exception to all this is the TDMA - TDMA case for which, as in the random-flows scenario, the behaviour of the prediction error is more erratic.

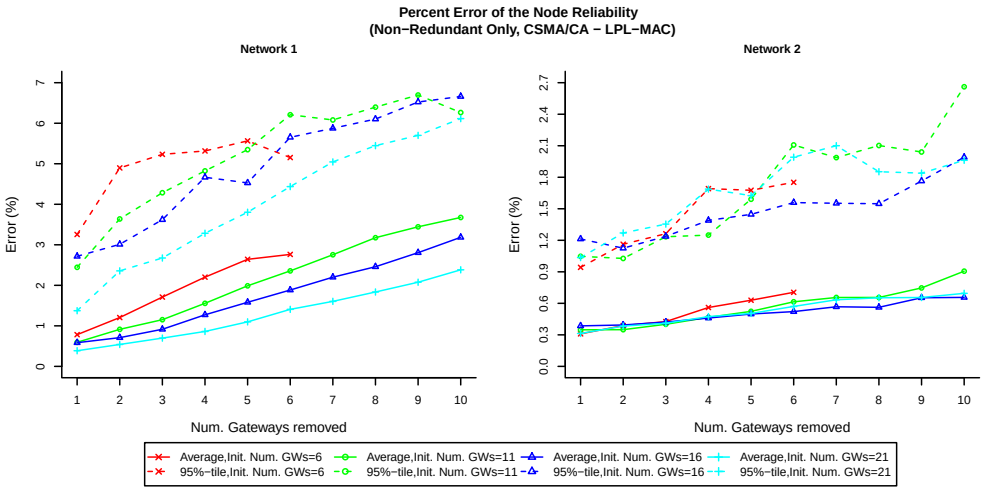


Figure 6.39: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and LPL-MAC protocols.

When the magnitude of the prediction error is considered it quickly becomes clear that, due to the 'larger independence' between the networks, this is mostly dependent on the MAC protocol of the network itself and while it is still influenced by the MAC protocol of the other network, this influence is less significant than it was in the random-flows scenario. For the LPL-MAC protocol the error is very low overall and never rises to more than 3.5%. When using CSMA/CA, the prediction error is also generally quite low with averages hovering between 1% and 3% and 95-percentiles varying between 3% and 7%. For the TDMA MAC protocol the average prediction error is in most cases still quite low but when a large number of virtual gateways are removed, it can become as high as 6% when combined with another MAC protocol and as 8% in the TDMA - TDMA case. The 95-percentile is, likewise in most cases quite low when only a few virtual gateways are removed but even then it can, in the T-MAC - TDMA case, still exceed 10%. Moreover, when more virtual gateways are removed at the same time, the 95-percentile error rises rapidly and can, at worst, reach nearly 26%. Finally, as with the random-flows scenario, the prediction error for the T-MAC network is even more substantial with the average error rising to, at worst, 30% and the 95-percentile error rising to around 45%. When only a single gateway is removed, the prediction error for the T-MAC protocol is substantially lower and while in that case the average error is never more than 5%, the 95-percentile error can in some cases still reach as high as 15%.

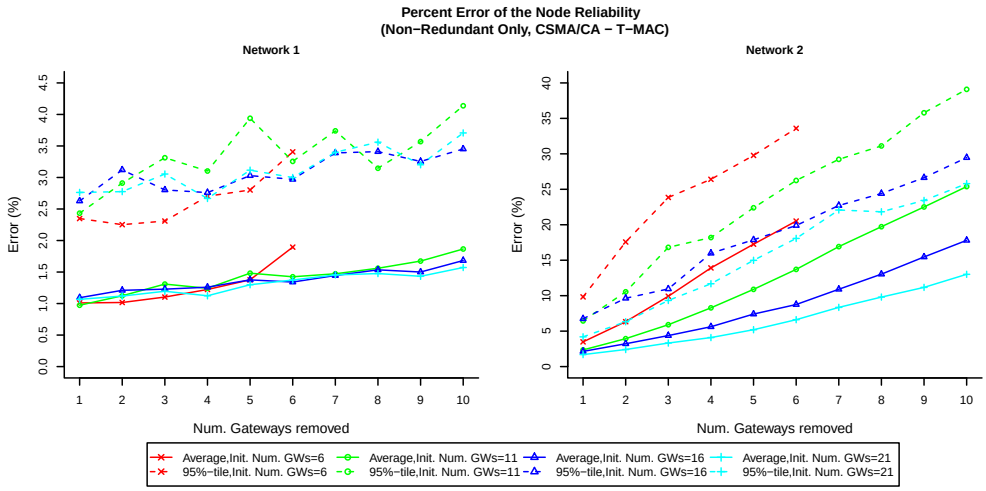


Figure 6.40: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and T-MAC protocols.

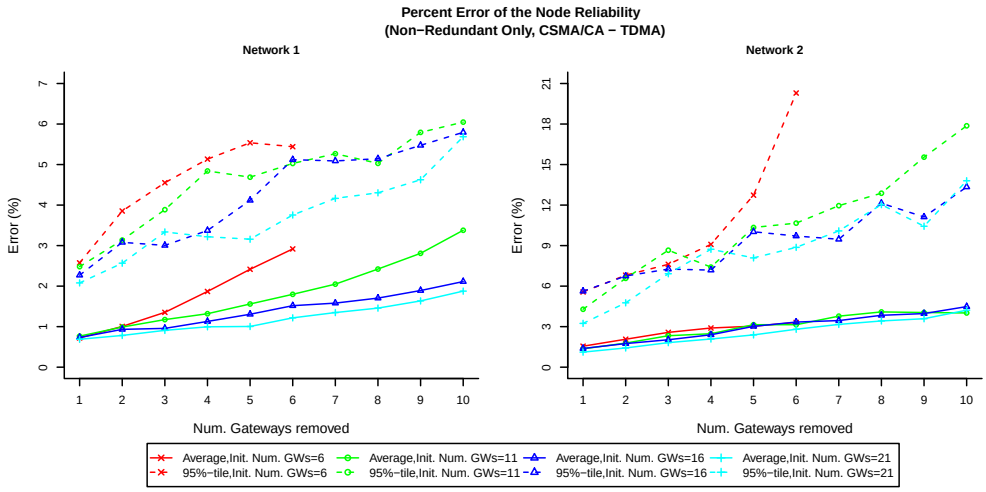


Figure 6.41: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using the CSMA/CA and TDMA MAC protocols.

Depending on the specific MAC protocols used, the error is in most cases either already fairly low or can otherwise be sufficiently reduced by choosing a conservative value for the 'number of gateways removed'-parameter. As with the random-flows scenario however, there are some cases where the prediction error can become quite high even when only a single gateway is removed. Despite the fact that even in those cases the error is not as substantial as in the random-flows scenario, the selection algorithm will still have to cope with the fact that the prediction error will sometimes be higher than expected.

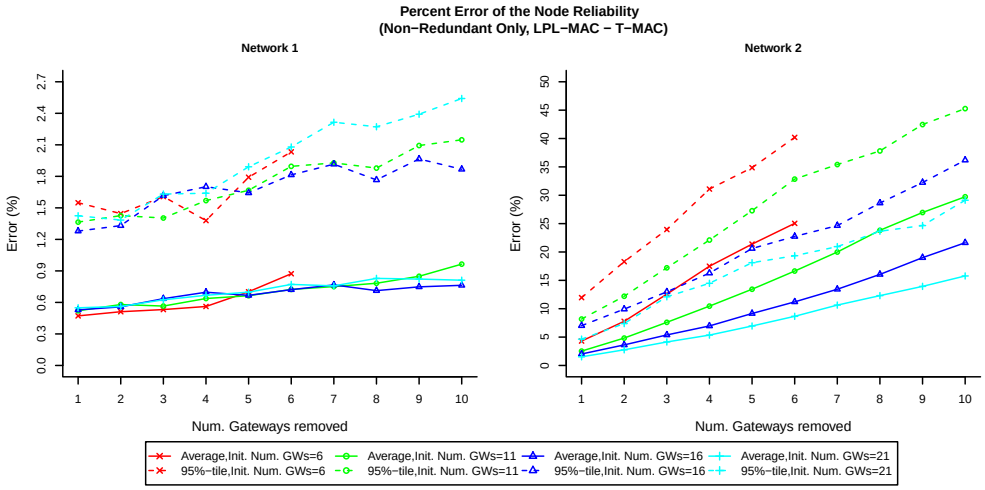


Figure 6.42: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using the LPL-MAC and T-MAC protocols.

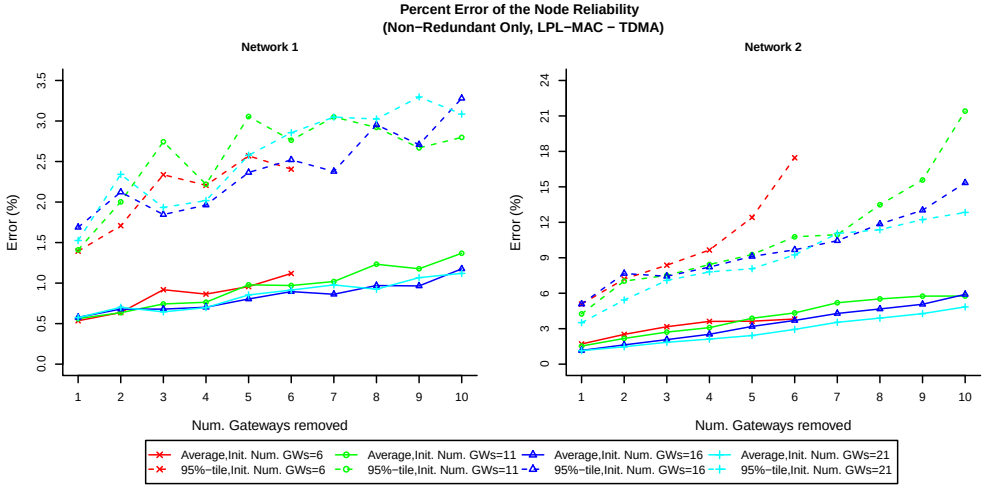


Figure 6.43: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using the LPL-MAC and TDMA MAC protocols.

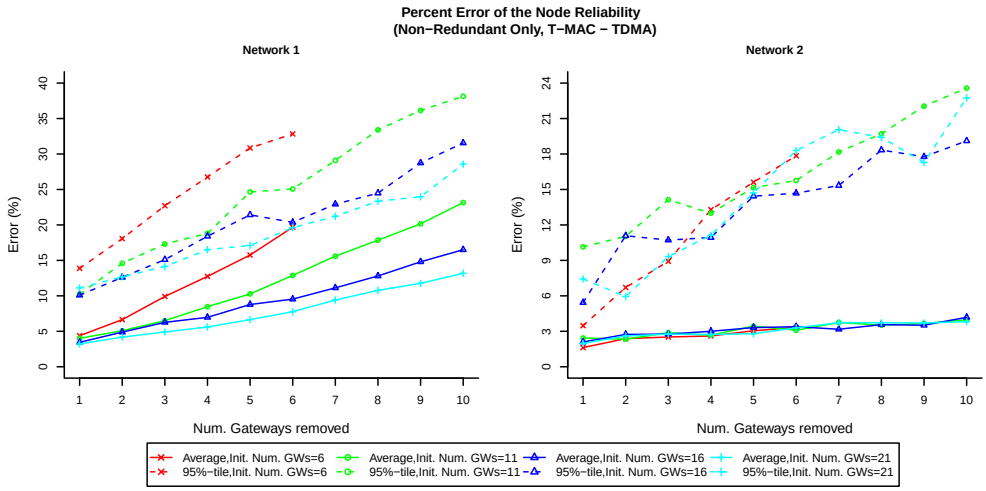


Figure 6.44: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using the T-MAC and TDMA MAC protocols.

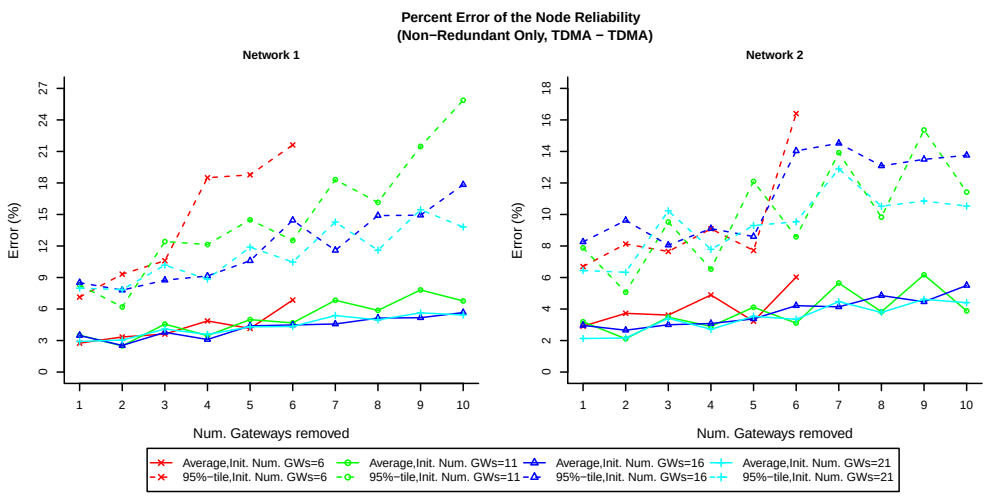


Figure 6.45: Average and 95%-tile of the node reliability prediction error when removing non-redundant gateways in the node-to-sink scenario when using two TDMA MAC protocols.

6.3 Modifications made to the Selection Algorithm

From the results discussed in the previous section it is clear that in most cases the prediction error of the prediction algorithm 'behaves' in much the same way for the node-to-sink scenario as it did for the random-flows scenario. As with the random-flows scenario the prediction error varies with the specific set of MAC protocols used by the networks and with the specific type of virtual gateways being removed (non-border, redundant or non-redundant). Moreover, when removing redundant or non-redundant gateways the prediction error also reacts in mostly the same way to changes in both the 'initial number of gateways' and the 'number of removed gateways' parameter as it did for the random-flows scenario. When the magnitude of the observed prediction error is considered it is clear that, again, in most cases the prediction error is either very similar to or even significantly lower than the ones encountered in the random-flows scenario and as with the random-flows scenario, the prediction algorithm is in most cases thus capable of accurately predicting the impact of removing one or more gateways on the duty cycle, node hop count and node reliability of the networks.

Despite this, the results of section 6.2 also make it clear that the prediction error can sometimes be considerably higher than in the random-flows scenario. This is predominantly the case when removing non-border gateways or when predicting the node hop count and one of the MAC protocols involved is the T-MAC protocol. As a result, the prediction error will in some cases be noticeably higher than the 'guideline' set for the prediction error in section 5.1.5 (5% average, 10% 95-percentile). Although it might seem that these larger errors would require some major modifications to be made to the selection algorithm this is not the case. The larger errors observed when removing non-border gateways in fact do not require any change to be made to the selection algorithm. The reason for this is that, as discussed in section 5.1.2, *all* non-border gateways are removed in a *single* iteration at the end of the preparation phase. This means that, when removing non-border gateways, the selection algorithm does not have to choose between multiple gateway configurations based on the predictions of the prediction algorithm. As a result, the larger prediction error encountered when removing non-border gateways does not affect the outcome of the selection algorithm.

The only alteration made to the selection algorithm is that during the redundant removal phase only three gateways are removed per iteration instead of four (the reason for using this 'step size' is further discussed below). As for the random-flows scenario, gateways are removed one at a time during the non-redundant removal phase of the selection algorithm. The prediction errors observed for these chosen 'step sizes' are summarised in figures 6.46 to 6.48. In all these figures the values shown are the *maximum* average, 5-percentile and 95-percentile errors encountered for the node-to-sink scenario over the different values for the 'initial number of gateways' parameter.

When removing redundant gateways 3 at a time, the average and 95-percentile error are, in most cases, within the respective 5% and 10% boundaries. The main exception is when predicting the node hop count when one of the networks involved uses the T-MAC protocol. In that case the average prediction error can reach nearly 8% while the 95-percentile error can reach nearly 18%. Although these errors are significantly higher than the guideline 'error-boundaries' it should be noted that even in those cases this high prediction error only occurs for the T-MAC network. For the other network involved,

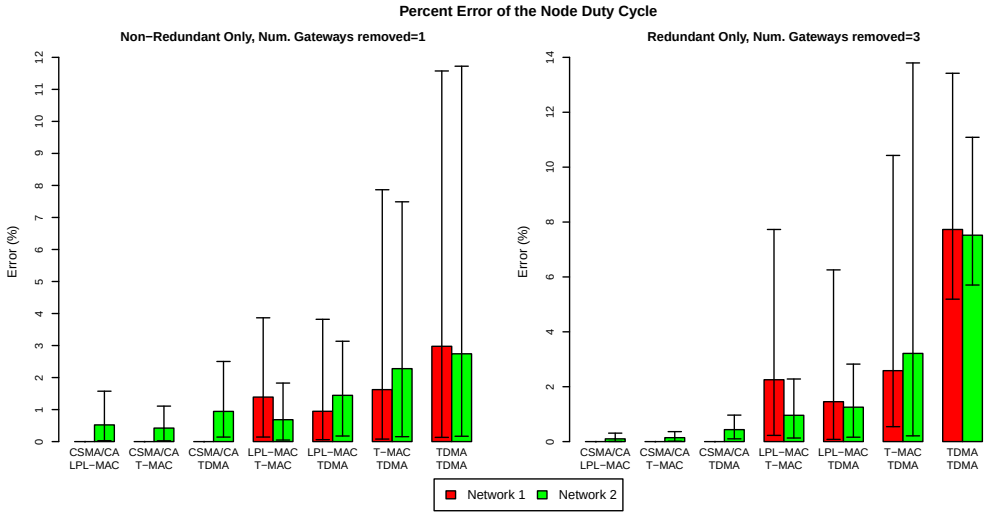


Figure 6.46: Maximum average, 5%-tile & 95%-tile error encountered for the node duty cycle when either removing 1 non-redundant gateway or 3 redundant gateways.

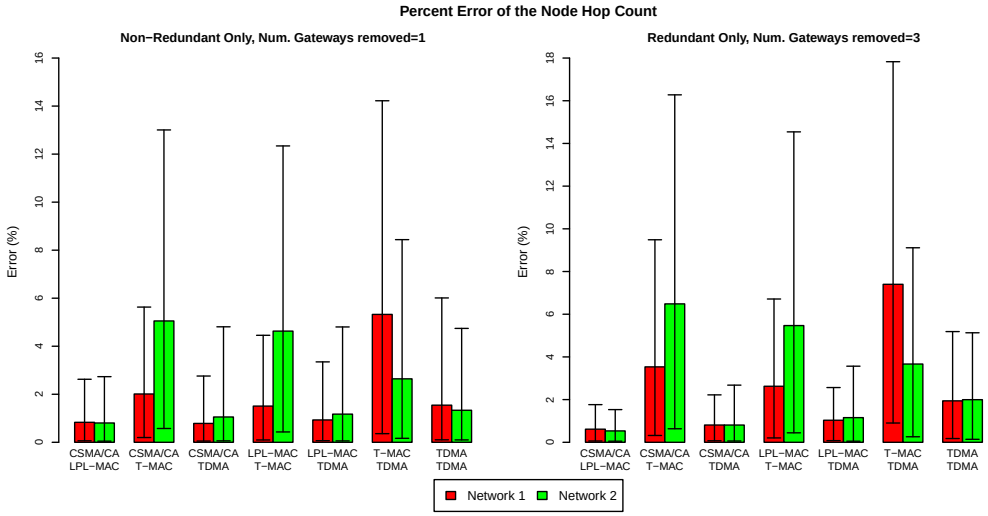


Figure 6.47: Maximum average, 5%-tile & 95%-tile error encountered for the node hop count when either removing 1 non-redundant gateway or 3 redundant gateways.

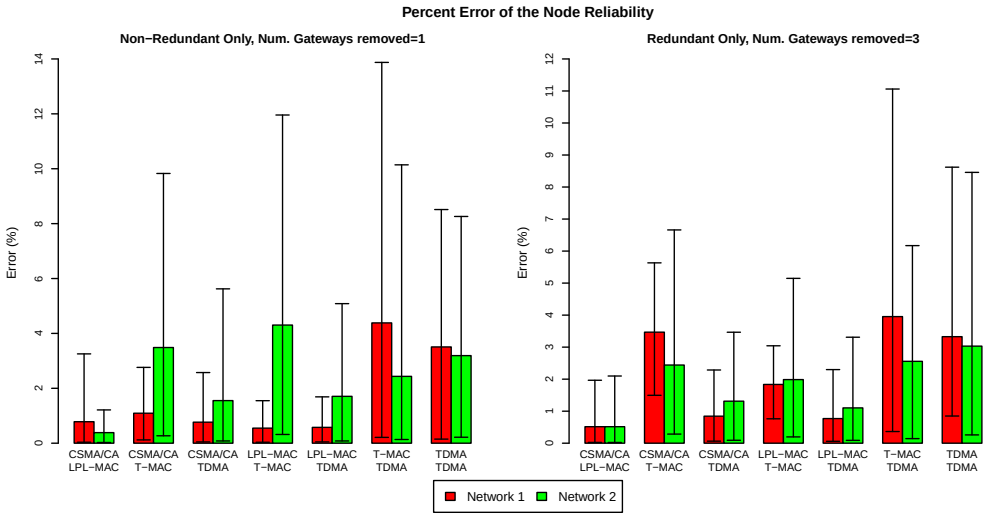


Figure 6.48: Maximum average, 5%-tile & 95%-tile error encountered for the node reliability when either removing 1 non-redundant gateway or 3 redundant gateways.

the observed prediction error remains within the "5% average, 10% 95-percentile" error boundary. Given that, as discussed in section 6.2.2.2, the number of gateways removed in a single iteration has little effect on the prediction error of the T-MAC network, there is no point in reducing the step size any further in an attempt to reduce these larger errors.

The only three other cases where the prediction error is higher than the error boundary set in section 5.1.5, are the T-MAC - TDMA and TDMA - TDMA cases when predicting the node duty cycle and the T-MAC - TDMA case when predicting the node reliability. Even so for the T-MAC - TDMA case the average prediction error does not rise above the 5% boundary. Only the 95-percentile error is out of bounds and even then it is still lower than some of those encountered in the random-flows scenario. For the TDMA - TDMA case both the average and the 95-percentile error are higher than the "5% average / 10% 95-percentile" boundary but even then the 95-percentile error is lower than for the random-flows scenario. The larger 95-percentile errors encountered for the random-flows scenario however do not seem to have negatively affected the selection-algorithm. Given that for all three cases the 'step size' would have to be reduced to 1 for the 95-percentile error to drop below the 10% boundary, the single spike in average prediction error encountered for the TDMA - TDMA case therefore does not warrant reducing the 'step size' any further than the currently chosen value.

When removing non-redundant gateways, the average and 95-percentile error are, again, in most cases within the respective 5% and 10% boundaries. The only exceptions are when predicting the node duty cycle for the TDMA - TDMA case and when predicting either the node hop count or the node reliability when one of the networks involved used the T-MAC protocol. In those cases, the 95-percentile error can rise to around 14%. Given however that this is still lower than some of the prediction errors encountered for the random-flows scenario, the *whitelisting* mechanism of the selection-algorithm (see

section 5.1.2) should be able to compensate for those cases where the prediction error is somewhat higher than expected.

6.4 Evaluation of the selection algorithm for the node-to-sink scenario

The test setup used to evaluate the selection algorithm of IRVG for the node-to-sink scenario is very similar to the one described in section 5.2. The main difference is that, as with the evaluation of the prediction algorithm (see section 6.2), a different application is running on the nodes of both networks. In addition, 'sink sharing' is enabled for both networks and the prediction algorithm is initialised with the same set of replacement policies as used in the evaluation of the prediction algorithm.

As with the evaluation of the selection algorithm for the random-flows scenario, 100 independent test runs are performed for each parameter set and a number of additional simulations are performed per test run to determine amongst others, the 'No Interference' and the 'Same MAC' performance of the networks. The performance for the 'No Interference' case is once again used to set the *goals* of the individual metrics while the *weights* once again vary with the specific test case.

One major difference between the tests performed here and the ones discussed in section 5.2 however is that the 'Same MAC' performance is *not* used as a 'baseline' performance to compare the performance of the IRVG-generated configurations to. This role is instead performed by the performance resulting from the *minimum* gateway configuration. The reason for doing so is that virtual gateways are used for a different purpose in the node-to-sink scenario than in the random-flows scenario. In the random-flows scenario inter-network communication, and thus virtual gateways, are essential to the operation of both networks. Because of this, the evaluation in section 5.2 was mainly focussed on determining the overhead incurred by the use of virtual gateways (and IRVG) relative to the idealised 'Same MAC' case. In the node-to-sink scenario however, inter-network communication is not essential to the operation of the networks involved. Virtual gateways are used solely as a network optimisation technique. Rather than using the 'Same MAC' case as a baseline, it would therefore be much more interesting to determine how much performance (in terms of duty cycle, hop count or reliability) is actually gained (or lost) by the use of virtual gateways in combination with 'sink sharing' and IRVG. To do so, the performance resulting from the use of IRVG needs to be compared to the case where no virtual gateways are used. Given that in the node-to-sink scenario the *minimum* gateway configuration is empty, this corresponds to using the performance resulting from the minimum gateway configuration as a baseline.

The performance of the 'Same MAC' case is still used in the evaluation of IRVG, but only as a 'best case' performance figure rather than the baseline performance. In addition, the performance of IRVG is also compared to the performance of the same 'random-selection' algorithms that were used in the evaluation of IRVG for the random-flows scenario. As with the evaluation discussed in section 5.2 however, not all 'random configurations' are shown in the graphs below. Only the 'fixed-size' random configurations with a size of 5, 10 and 15 gateways and the random configuration containing the same number of nodes

as the configuration as IRVG are shown in the graphs below. For those interested the full data set is posted online [139].

6.4.1 Single-metric optimisation

This section investigates the performance of IRVG separately for each individual metric. This is done by, for each network, assigning a weight of 1.0 to the relevant metric while the weights of the other metrics are set to 0.0. As for the random-flows scenario (see section 5.2.1), this is done to prevent IRVG from having to trade-off between the performance of multiple metrics and thereby allows the performance of the metric being investigated to be compared between different virtual gateway configurations (such as the one generated by IRVG and the ones generated by the various random-selection algorithms).

As with the test results discussed in section 5.2.1, the obtained performance measurements are expressed as a '*relative difference*' compared to the baseline performance. As discussed above however, the performance of the *minimum* gateway configuration is used as a baseline rather than the 'Same MAC' performance. The main advantage of using this relative representation is that it makes it immediately clear how much performance is gained by the use of virtual gateways in combination with IRVG. The difference between the more commonly used 'absolute' representation and the relative representation used in this section is exemplified by figures 6.49, 6.53 and 6.57 each showing the two representations side by side for the same data set.

6.4.1.1 Node Duty Cycle

The performance of IRVG when optimising only for node duty cycle is shown in figures 6.49 to 6.52. These figures show that, as far as the node duty cycle is concerned, there is very little benefit to using virtual gateways and IRVG in the node-to-sink scenario. When either LPL-MAC or T-MAC are combined with CSMA/CA, IRVG is able to reduce the node duty cycle for these networks but this reduction is relatively small (on average 3%-points for LPL-MAC, 7%-points for T-MAC). In the LPL-MAC - TDMA and T-MAC - TDMA cases, IRVG is able to slightly improve the node duty cycle of respectively the LPL-MAC and the T-MAC network (by respectively 1.25 and 0.8%-points) at the cost of a slight increase (max 3.2%-points) in the node duty cycle of the TDMA network. It is worth noting at this point that, IRVG encourages but does not enforce 'fairness' between the networks. The reward function defined in section 5.1.3 allows the performance of one network to be optimised at the cost of the other network as long as (1) it causes the overall performance to increase and (2) it does not cause the discrepancy between the performance of these networks to become too large (the tradeoff between these two is controlled by the β-parameter). From the results shown in figures 6.51 and 6.52 it would therefore seem that for these two test cases, on average the (relative) performance-gain for the LPL-MAC and T-MAC network of using virtual gateways outweighs the duty cycle-wise cost incurred by the TDMA network. For all other combinations of MAC protocols, the performance of IRVG is exactly the same as when no virtual gateways are used. This is caused by IRVG reverting to the (empty) *minimum* gateway configuration at the end of the optimisation process, indicating that for these cases either no improvement in duty cycle can be achieved or that this would have incurred to-great-a performance cost for one of the networks involved.

The limited node duty cycle-wise benefits of using IRVG and virtual gateways in this scenario however are not entirely unexpected. As discussed in chapter 3, there is a not-unsubstantial duty cycle-wise cost associated with the use of virtual gateways. Since these gateways also make alternative (and in this case shorter) paths available, they can also reduce the amount of traffic sent in the networks and thereby reduce the duty cycle of the other nodes in the network. While, depending on the specific MAC protocols and node deployment used it may therefore be possible for the duty cycle wise cost of running a virtual gateway to be compensated for by a reduction in duty cycle of the other nodes in the networks, the results shown in figures 6.49 to 6.52 make it clear that this is generally not the case for the specific application scenario and node deployment considered in this chapter. In the CSMA/CA - LPL-MAC and CSMA/CA - T-MAC cases IRVG is able to achieve a net performance improvement, but this is due to IRVG placing all virtual gateways in the CSMA/CA network for which the duty cycle wise cost of running one or more virtual gateways is exactly zero. For the node deployment and application scenario considered here, the use of virtual gateways is thus in general only beneficial if the duty cycle-wise cost of running a virtual gateway can be offset by an increased performance in one of the other metrics (such as node reliability or node hop count). This however will depend on the *goals* and *weights* assigned to each metric by the individual networks.

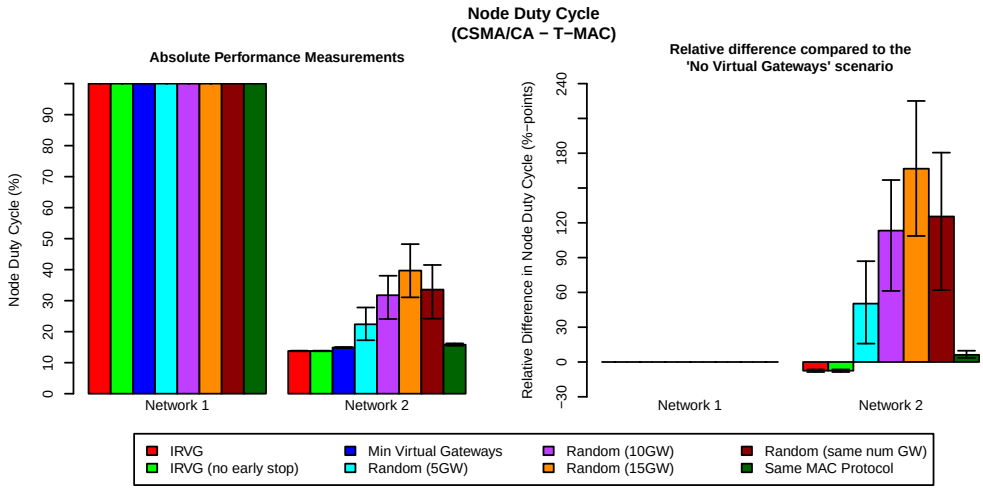


Figure 6.49: Average, 5- and 95%-tile of the node duty cycle in the 'single metric' optimisation case when using CSMA/CA and T-MAC. The left graph shows the absolute node duty cycle measurements while the graph on the right shows the relative difference of the node duty cycle when compared to the case where no virtual gateways are used. The duty cycle of the CSMA/CA network is unaffected by the use of IRVG while for the T-MAC network, IRVG is able to reduce the duty cycle by on average 7.5%-points.

When comparing the performance resulting from the use of IRVG against the other test cases it is immediately clear that IRVG performs significantly better than any of the 'randomly generated' virtual gateway configurations. The cost of using virtual gateways can be a few orders of magnitude larger when using one of the 'fixed-sized' randomly generated configurations than when using IRVG. This is hardly surprising given that IRVG can

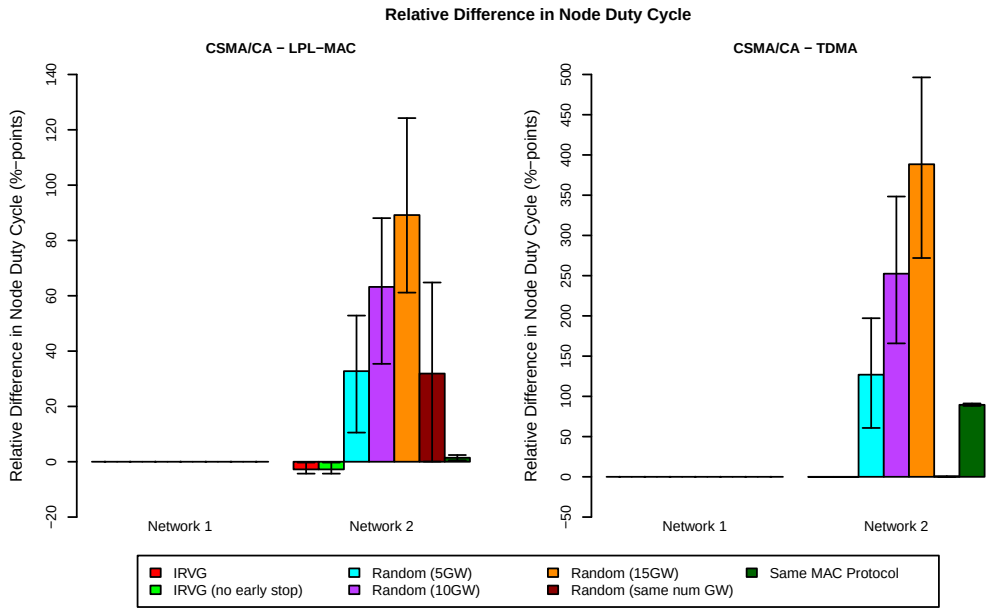


Figure 6.50: Average, 5- and 95%-tile of the relative difference in node duty cycle in the 'single metric' optimisation case when using either CSMA/CA and LPL-MAC or CSMA/CA and TDMA.

intelligently decide how many gateways to use whereas this is not the case for the 'fixed-size' random-configurations. When IRVG reverts back to the *minimum* gateway configuration, the configuration generated by IRVG performance of IRVG equals that of the "Random (same num GW)" virtual gateway configuration. This is also hardly surprising since in those cases both configurations are empty and thus equal to one another. In all other cases IRVG outperforms the "Random (same num GW)" configuration indicating that IRVG is also better at deciding which (rather than how many) virtual gateways to enable. Finally, IRVG in most cases also outperforms the 'Same MAC'-scenario. The only exception is the LPL-MAC network in the LPL-MAC - TDMA case and even then the difference is relatively small.

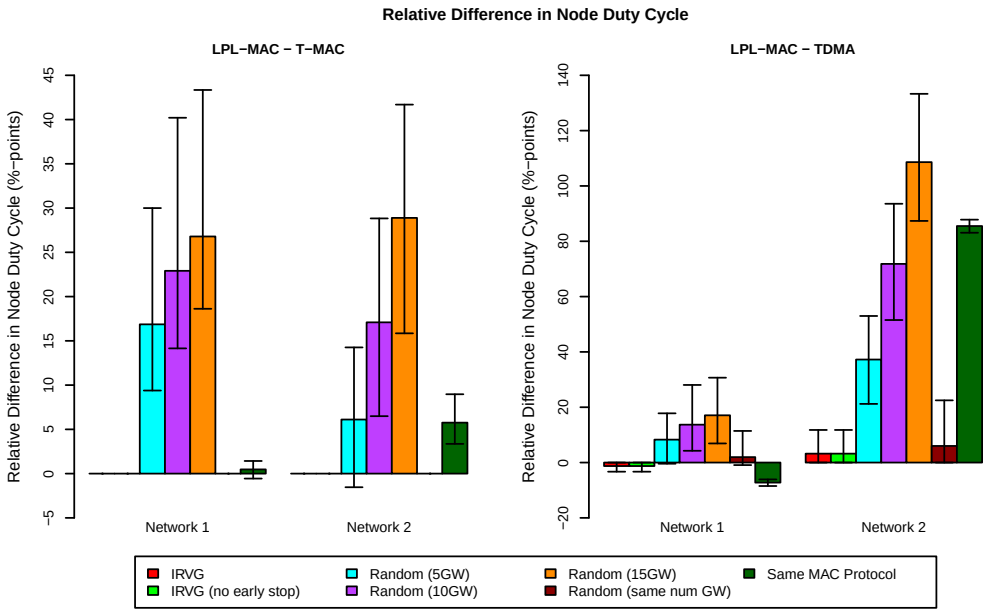


Figure 6.51: Average, 5- and 95%-tile of the relative difference in node duty cycle in the 'single metric' optimisation case when using either LPL-MAC and T-MAC or LPL-MAC and TDMA.

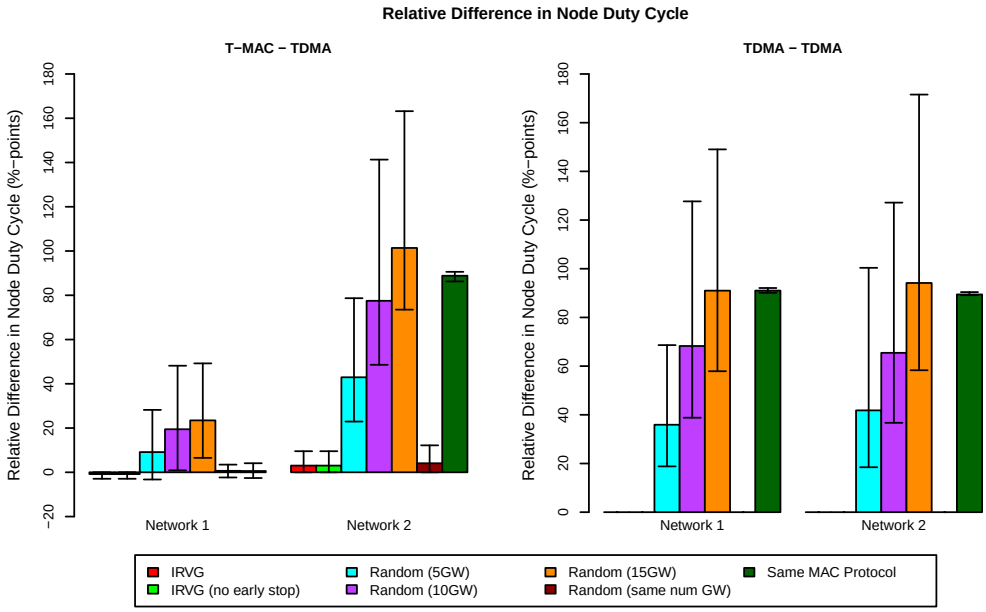


Figure 6.52: Average, 5- and 95%-tile of the relative difference in node duty cycle in the 'single metric' optimisation case when using either T-MAC and TDMA or two TDMA MAC protocols.

6.4.1.2 Node Hop Count

The performance of IRVG when optimising only for node hop count is shown in figures 6.53 to 6.55. These figures show that, when it comes to the node hop count, there is definite benefit to using virtual gateways in combination with 'sink sharing'. This is clear from the performance recorded for the various 'random' gateway configurations alone which show that, even with a pure random selection strategy the introduction of virtual gateways into the wireless environment results in most cases in a lower node hop count for both networks. In addition, the gateway configurations generated by IRVG perform significantly better than any of the randomly generated configurations, which shows that there is an added benefit to using IRVG *in combination* with virtual gateways and 'sink sharing' when hop count is considered. These results should not come as a surprise given that the 'sink sharing' mechanism in essence doubles the number of sink nodes available to both networks and thus allows the routing paths used to be dramatically shortened once inter-network communication is enabled.

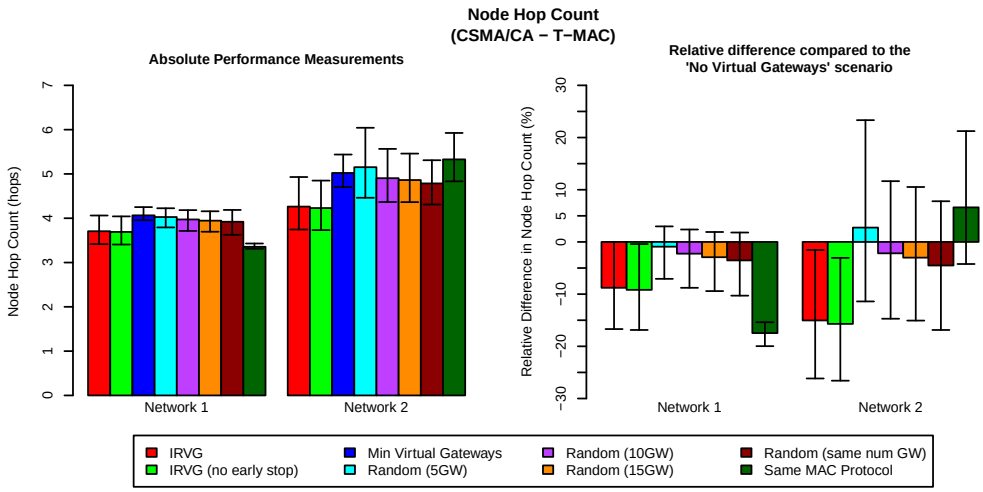


Figure 6.53: Average, 5- and 95%-tile of the node hop count in the 'single metric' optimisation case when using CSMA/CA and T-MAC.

These figures however also make it plain that there is a significant difference in performance depending on whether or not the T-MAC protocol is used. When one of the networks uses the T-MAC protocol, IRVG is able to achieve a relatively high performance improvement for the T-MAC network (between 15% and 18%), but for the other network the performance gain is noticeably lower (between 5% and 9%). Interestingly, this difference in performance is caused by the fact that the limitations of the T-MAC network hamper the performance of the other network involved. As exemplified by the left graph of figure 6.53, the baseline ('Min Virtual Gateways') node hop count is significantly higher for the T-MAC protocol than it is for the other MAC protocols. Moreover, this problem is only made worse when virtual gateways are introduced since the node hop count for the T-MAC protocol rises with the number of T-MAC nodes in the wireless environment (see section 6.2.1). Given that traffic crossing the network boundary is affected by the MAC protocols of both networks, this causes 'foreign' traffic sent to the sink node of the T-

MAC network to travel over a comparatively longer path than if another MAC protocol were used. This not only elongates the path between the source and the sink node but also causes more 'foreign' source nodes to prefer their own sink to the one of the T-MAC network. Given that in the node-to-sink scenario performance is optimised by redirecting traffic to a different sink node, this causes the relative performance of the network combined with the T-MAC network to be comparatively lower. For traffic originating in the T-MAC network the exact opposite happens and as a result it should come as no surprise that the performance of the T-MAC network is comparatively higher.

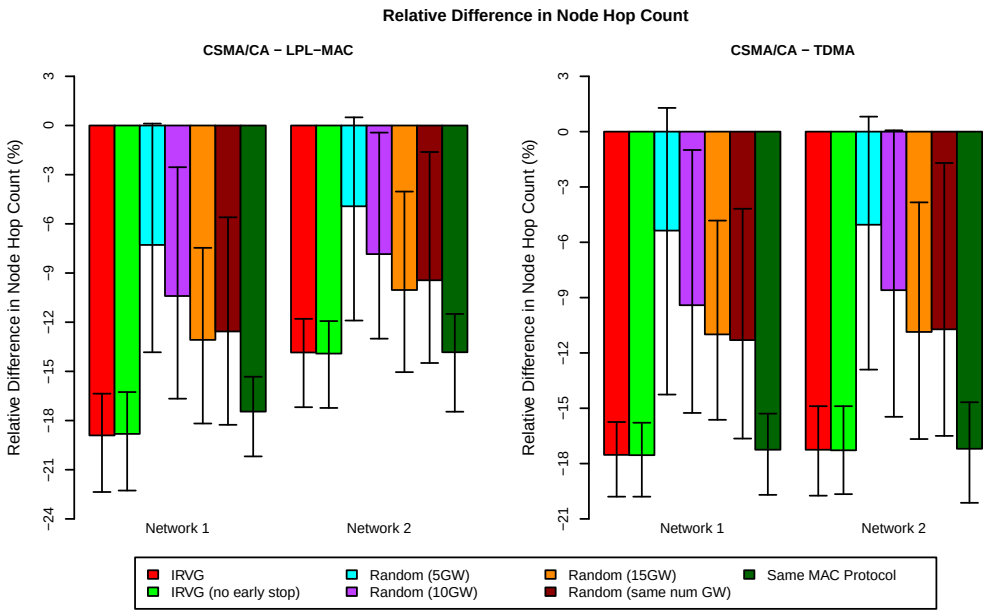


Figure 6.54: Average, 5- and 95%-tile of the relative difference in node hop count in the 'single metric' optimisation case when using either CSMA/CA and LPL-MAC or CSMA/CA and TDMA.

A similar argument can be made when comparing the performance of IRVG to the 'Same MAC'-performance. Figures 6.53, 6.55 and 6.56 show that despite there being twice as many sink nodes available in the 'Same MAC'-scenario, the node hop count of the T-MAC network is *higher* (rather than lower) for the 'Same MAC'-scenario than it is for the case where no virtual gateways are used. Given that using virtual gateways and IRVG yields a significantly better performance than this baseline scenario it should therefore not come as a surprise that for the T-MAC network the performance of IRVG far exceeds that of the 'Same MAC'-scenario. For the other network involved IRVG is not able to match the 'Same MAC'-performance but this is once again due to the fact that when sending data to the sink node, more hops are required to do so when using the T-MAC protocol than when another MAC protocol is used.

This discrepancy between the performance of the two networks however, *only* occurs when one of the networks involved uses the T-MAC protocol. If the T-MAC protocol is *not* used, the average performance gain achieved by IRVG is, again, relatively high (between

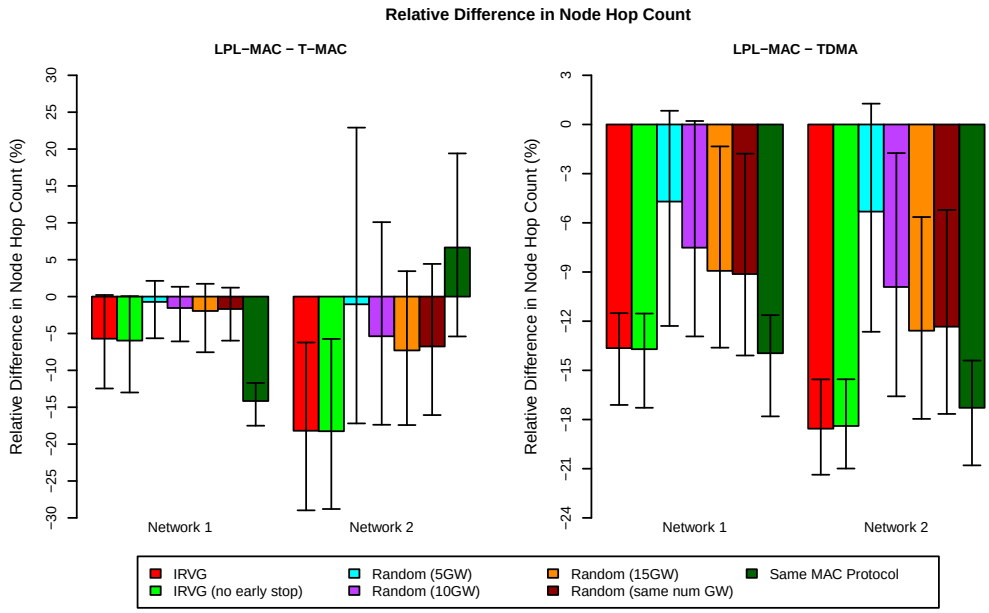


Figure 6.55: Average, 5- and 95%-tile of the relative difference in node hop count in the 'single metric' optimisation case when using either LPL-MAC and T-MAC or LPL-MAC and TDMA.

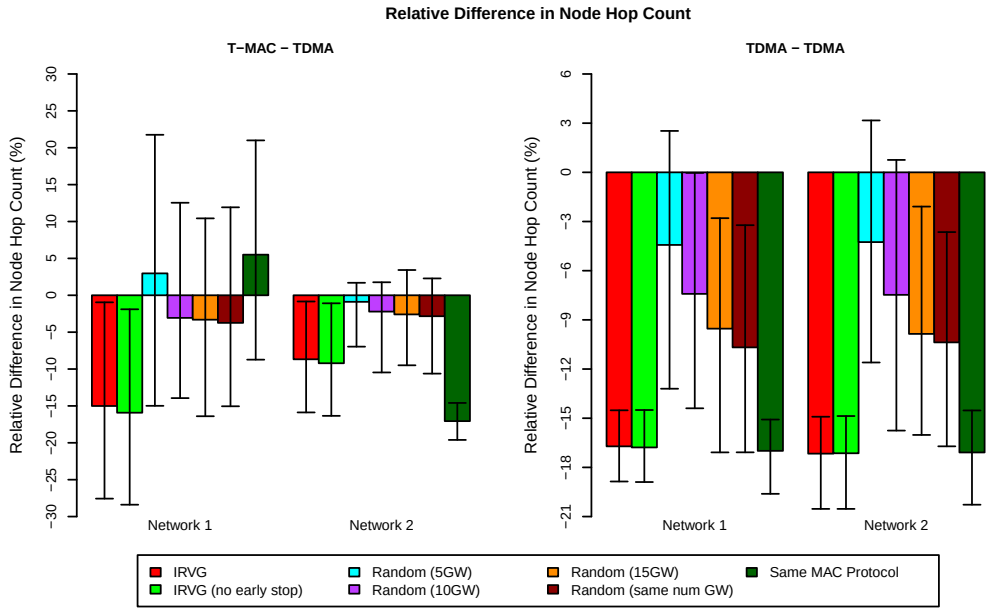


Figure 6.56: Average, 5- and 95%-tile of the relative difference in node hop count in the 'single metric' optimisation case when using either T-MAC and TDMA or two TDMA MAC protocols.

15% and 18%) but in contrast to the case where T-MAC is used, IRVG is able to achieve the same performance gain for both networks. More importantly IRVG is also able to match (and sometimes even exceed) the performance of the 'ideal' Same MAC scenario. Given that the 'Same MAC'-scenario benefits in the same manner from the 'sink sharing' mechanism as the various virtual gateway configurations but does not suffer from the interference between heterogeneous MAC protocols, the fact that IRVG is able to match the performance of the 'Same MAC' scenario is thus a significant result.

6.4.1.3 Node Reliability

The performance of IRVG when optimising only for node reliability is shown in figures 6.57 to 6.60. These figures show that, as far as node reliability is concerned, the performance gain that can be achieved by having the networks cooperate through the use of virtual gateways and 'sink sharing' depends heavily on the specific MAC protocols used. The large difference between the 5- and 95-percentiles recorded for each of the test cases also indicate that there is a very significant variation in node reliability between different runs of the same test case. This in turn indicates that the node reliability is also influenced significantly by the specific timings used by in the application and the various layers of the network stack.

When considering the individual test cases, it is clear that networks using either the T-MAC or TDMA MAC protocol have the most to gain from cooperating with another network while for networks using either the CSMA/CA or LPL-MAC protocol, the benefits are much less significant. When CSMA/CA is combined with LPL-MAC IRVG is, on average, able to achieve a small performance improvement for both networks involved (4%-points and 1%-point for respectively the CSMA/CA and LPL-MAC network). The fact that for the LPL-MAC network the 5-percentile node reliability is lower than that of the baseline scenario however, does indicate that the performance improvement achieved for the CSMA/CA network can come at the cost of a small performance reduction for the LPL-MAC network. When T-MAC is combined with either CSMA/CA or with LPL-MAC, the network using T-MAC benefits greatly from this cooperation. In the case that T-MAC is combined with CSMA/CA the node reliability of the T-MAC rises by, on average, 42%-points while for the LPL-MAC - T-MAC case a performance improvement of 48%-points is achieved. For the other network involved this cooperation does come at the cost of a slight reduction in node reliability but this effect is very small (on average less than 1.4%-points). A similar observation can be made for the cases where CSMA/CA or LPL-MAC are combined with TDMA. In those cases, the node reliability of the TDMA network is improved by on average 10%-points when combined with CSMA/CA and on average 20%-points when combined with LPL-MAC. As with the T-MAC protocol, this improvement of reliability for the TDMA network comes at the cost of a slight drop in node reliability for the CSMA/CA and LPL-MAC networks (respectively 1 and 1.5%points).

Interestingly, IRVG is able to achieve a significant performance benefit for both networks in the T-MAC - TDMA case: the node reliability of the T-MAC network is improved by, on average, 26%-points while the node reliability of the TDMA network is improved by on average 10%-points. In addition, IRVG is also able to achieve a net performance improvement for both networks in the case where two TDMA protocols are used but

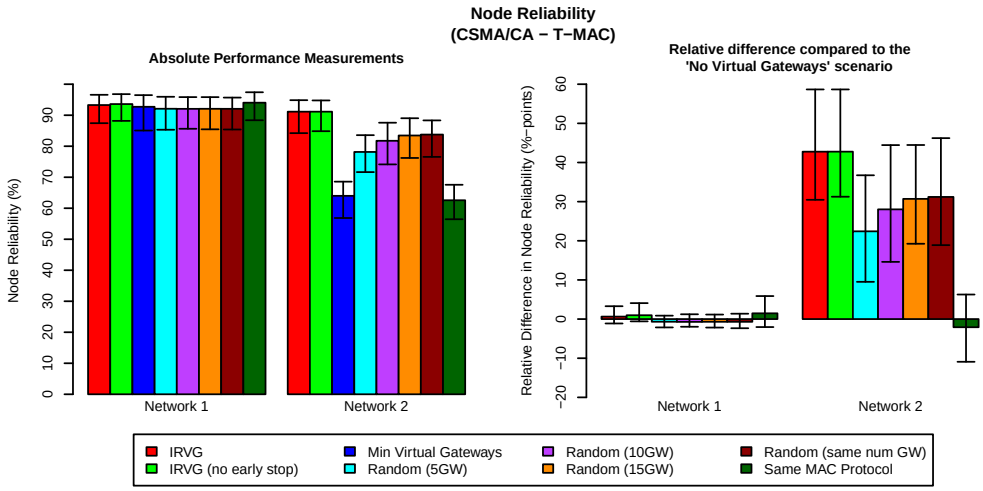


Figure 6.57: Average, 5- and 95%-tile of the node reliability in the 'single metric' optimisation case when using CSMA/CA and T-MAC.

in that case the performance gain is less spectacular than in the T-MAC - TDMA case (around 4% for each network). This would seem to indicate that the rather poor performance gain achieved for the CSMA/CA and LPL-MAC protocols is not due to the T-MAC and TDMA MAC protocols having a negative effect on the performance of the other network but is instead due to the properties of the CSMA/CA and LPL-MAC protocols themselves. A closer inspection of the *baseline* node reliability reveals this to be indeed the case. For the node-to-sink scenario and node deployment considered in this chapter, the baseline node reliability of the CSMA/CA and LPL-MAC protocols is already well above 90%. While this indicates that these MAC protocols are able to cope very well with inter-MAC interference, this also leaves very little room for any additional improvement in node reliability. In contrast, the *baseline* node reliability of the TDMA and T-MAC protocols varies between 60% and 80% depending on the specific combination of MAC protocols and as a result, IRVG and 'sink sharing' are able to achieve a much more significant performance improvement.

When comparing the performance resulting from the use of IRVG against the other test cases it is immediately clear that IRVG performs significantly better than any of the 'randomly generated' virtual gateway configurations. When comparing against the performance of the 'Same MAC' scenario it is clear that the performance of IRVG varies with the specific MAC protocols used. In many cases IRVG is able to substantially improve on the performance of the 'Same MAC' scenario (this is predominantly the case for networks using the T-MAC or TDMA MAC protocol). Despite this, there are also a number of cases where IRVG is not able to match the 'Same MAC' performance. This is predominantly the case for networks using the CSMA/CA or LPL-MAC protocol and is mainly due to IRVG 'giving priority' to the performance of the other network involved.

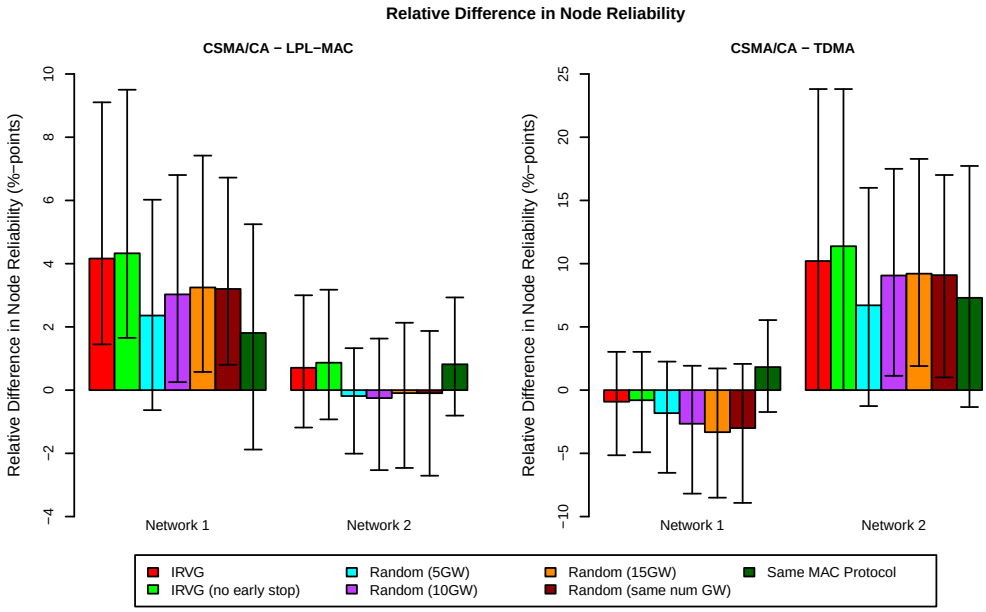


Figure 6.58: Average, 5- and 95%-tile of the relative difference in node reliability in the 'single metric' optimisation case when using either CSMA/CA and LPL-MAC or CSMA/CA and TDMA.

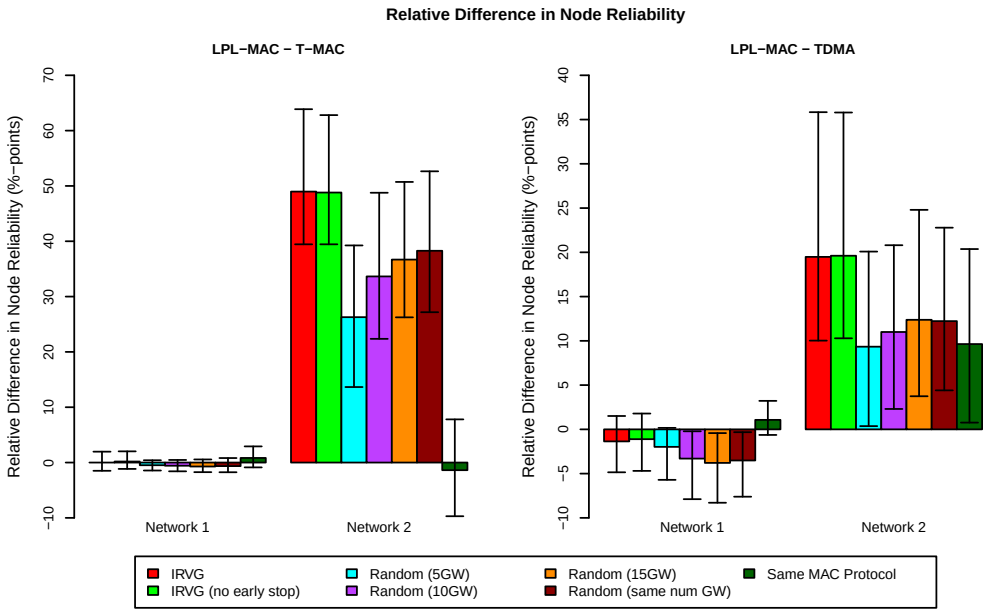


Figure 6.59: Average, 5- and 95%-tile of the relative difference in node reliability in the 'single metric' optimisation case when using either LPL-MAC and T-MAC or LPL-MAC and TDMA.

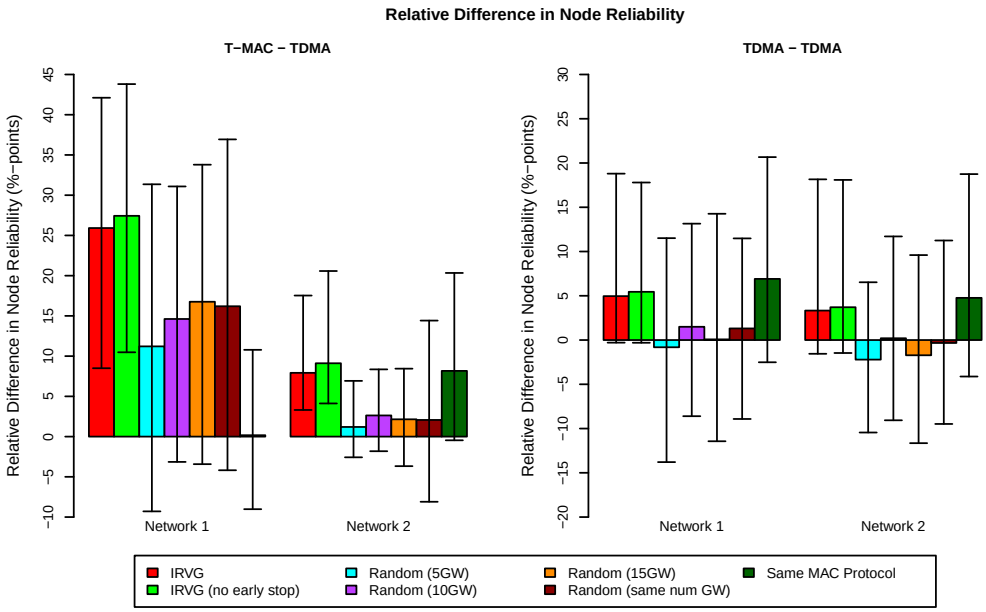


Figure 6.60: Average, 5- and 95%-tile of the relative difference in node reliability in the 'single metric' optimisation case when using either T-MAC and TDMA or two TDMA MAC protocols.

6.4.2 Multi-metric optimisation

This section investigates the behaviour of IRVG when multiple metrics need to be taken into consideration. Since each network may assign different weights to the individual metrics, this means that IRVG also needs to find a balance between the possibly conflicting requirements of these networks. As for the multi-metric tests performed for the random-flows scenario (see section 5.2.2), the *goals* specified for each of the metrics are the performance values measured for these metrics in the 'No Interference' scenario while the *weights* assigned to these metrics are decided entirely by the default MAC protocol of the network. The different metric weights used for each MAC protocol are the same as the ones used for the random-flows scenario and are listed in table 5.1.

6.4.2.1 Reward & Performance tradeoffs

The Total Reward achieved by IRVG for the various combinations of MAC protocols is shown in figure 6.61. As with the presentation in section 5.2.2.1, these rewards are plotted along a non-linear Y-axis to compensate for the fact that the exponential function used to calculate the reward, tends to 'squish' the rewards together as the performance metrics near their goals. To illustrate the trade-offs made by IRVG to attain these total rewards, figures 6.62 and 6.64 also show the performance obtained for the individual metrics in the CSMA/CA - T-MAC and the T-MAC - TDMA cases. As for the multi-metric tests performed for the random-flows scenario, these three figures all show the performance of IRVG (both with and without 'early stop') alongside the performance achieved by the various 'random configurations' and the performance of the 'No Interference' scenario. The performance of the 'Same MAC'-scenario is once again not shown as it is less relevant

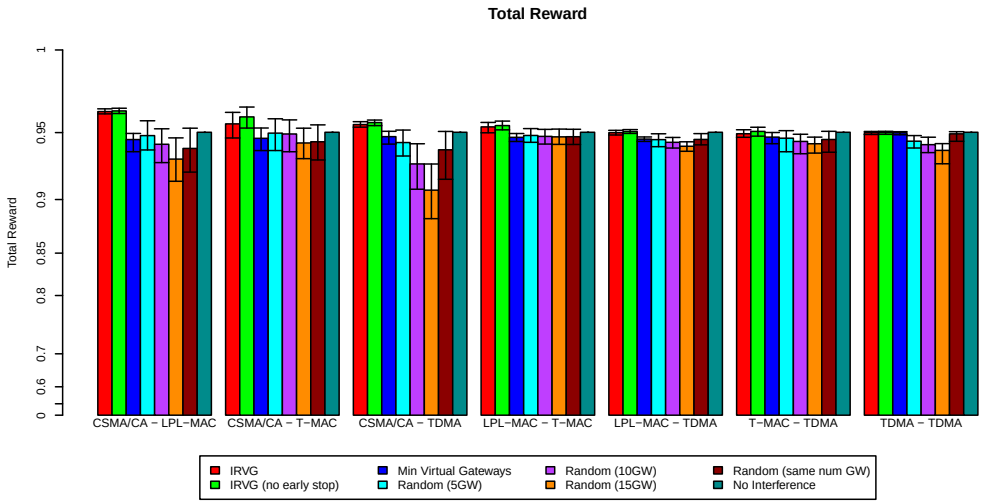


Figure 6.61: Average, 5- and 95%-tile of the Total Reward achieved by IRVG when optimising for multiple metrics in the node-to-sink scenario.

to the tradeoffs made by IRVG.

Figure 6.61 shows that for every combination of MAC protocols considered here, IRVG yields a better reward than the baseline 'Min Virtual Gateways' configuration. This is true both in the case that the 'early stop' check is enabled and in the case that this check is disabled and shows that IRVG is thus able to achieve a definite performance improvement over the case where no virtual gateways are used. Although this is also true for the case where two TDMA MAC protocols are combined, it should be noted that in that case the average performance improvement achieved by IRVG is very small. The reason for this is that in the TDMA - TDMA case, both networks assign most of their *weight* to the node duty cycle metric (0.8) while only a small weight is attached to the node reliability metric (0.2). Given that for the TDMA - TDMA case the duty cycle-wise cost of running a virtual gateway is too large to allow IRVG to optimise the node duty cycle of the networks through the use of virtual gateways (see section 6.4.1.1), this causes IRVG to return an empty gateway configuration in 81% of all test runs performed for this test case. As a result, the total reward achieved by IRVG is, on average, nearly identical to that of the baseline 'Min Virtual Gateways'-case.

When the performance of the various 'random configurations' are considered, figure 6.61 shows that for every combination of the MAC protocols considered here, IRVG yields a better reward than any of these 'random configurations'. In addition it should also be noted that these 'random configurations' often yield a reward that is lower than that of the baseline 'Min Virtual Gateways' configuration, which shows that the mere use of virtual gateways in combination with 'sink sharing' does not suffice to guarantee an improved performance. The specific nodes to be used as a virtual gateway also need to be selected carefully. It is also interesting to note that in a number of test cases, IRVG performs better than the 'No Interference'-scenario. This shows that for the node-to-sink scenario considered here, having the two networks cooperate through the use of IRVG and 'sink

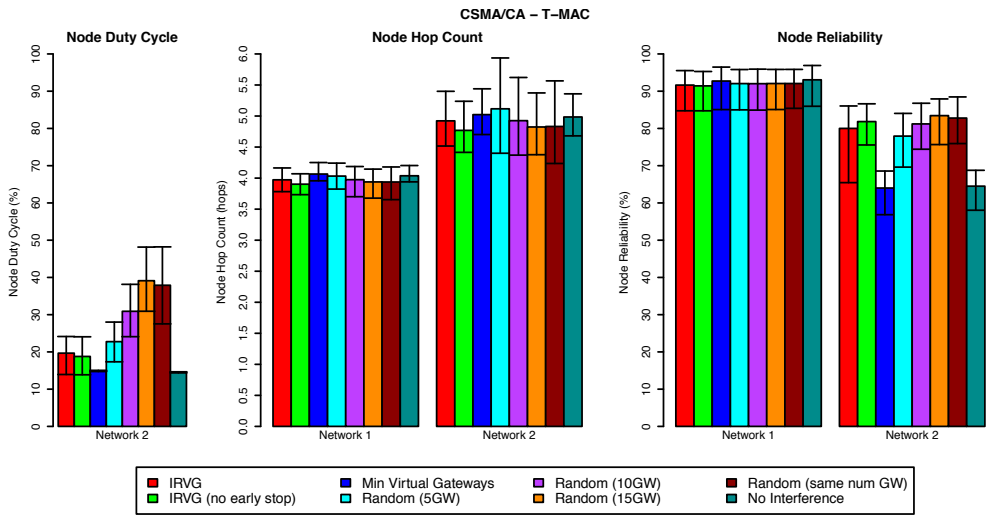


Figure 6.62: Average, 5- and 95%-tile of the node duty cycle, node hop count and node reliability for the CSMA/CA - T-MAC case. The node duty cycle of the CSMA/CA network is not shown as it is always 100%.

sharing' not only minimises the impact of the interference between the two networks but can, depending on the specific MAC protocols used, also result in a better performance than if there were only one network present in the wireless environment.

Figure 6.61 also shows that the 'gap' in reward between IRVG and the various other configurations varies quite a bit from one combination of MAC protocols to another. As discussed in section 5.2.2.1 however, these differences are not representative of the difference in performance for the individual metrics. To give a better insight into the actual performance underlying the total reward and the tradeoffs made by IRVG to achieve this reward, the CSMA/CA - T-MAC and the T-MAC - TDMA case are therefore discussed in more detail.

The performance of the individual metrics for the CSMA/CA - T-MAC case is shown in figure 6.62. In this particular case, the cooperation between the networks benefits mostly the T-MAC network. With 'sink sharing' enabled, IRVG is able to substantially improve the node reliability of the T-MAC network at the cost of a slight drop in reliability for the CSMA/CA network. Given that this drop in reliability is very small and that CSMA/CA assigns a relatively low *weight* (0.3) to the node reliability metric, this will only have a negligible effect on the reward. At the same time, IRVG is also able to lower the node hop count of both networks by a small amount. The fact that IRVG is not able to achieve a more substantial optimisation of the node hop count is most likely due to the fact that, as discussed in section 6.4.1.2, the 'sink sharing' mechanism is hampered in its ability to reduce the node hop count by the fact that the baseline node hop count for the T-MAC protocol is so much higher than that of the CSMA/CA MAC protocol. While this does not matter for the T-MAC network, it will have a significant effect on the per network reward that the CSMA/CA network can achieve since this network assigns a very high weight to the node hop count metric (0.7). When the node duty cycle metric

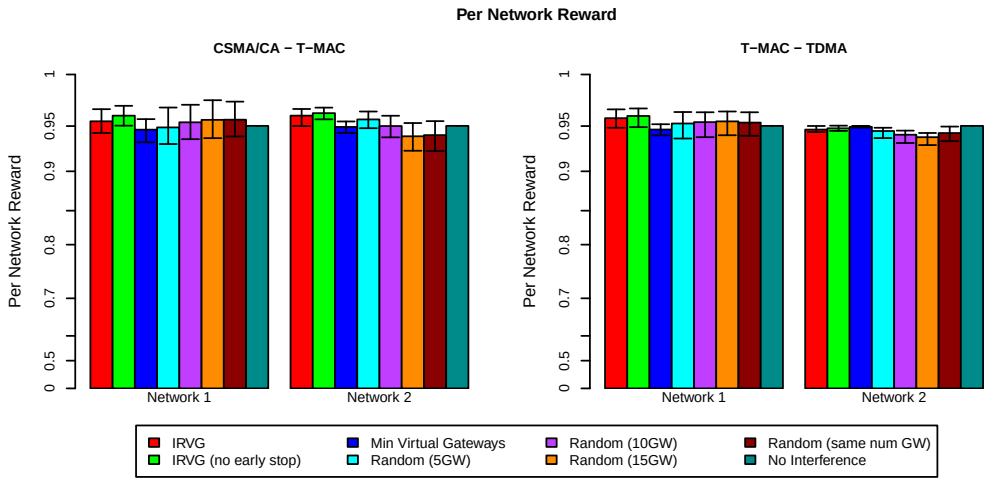


Figure 6.63: Average, 5- and 95%-tile of the per network reward for the CSMA/CA - T-MAC and T-MAC - TDMA cases.

is considered, it is clear that the significant boost in reliability achieved for the T-MAC network does come at the cost of a noticeable increase in duty cycle. For the T-MAC network, the node duty cycle (after optimisation) is around 19% whereas it is only around 15% when no virtual gateways are used. (The node duty cycle of the CSMA/CA network is not shown since it is not affected at all by the number of virtual gateways.) Moreover, when the performance of the 'random configurations' are considered, it is immediately clear that both the node duty cycle and the node reliability rise with the number of virtual gateways. Given that, for the T-MAC network, the node reliability achieved for the 'Random (15GW)' configuration is slightly higher than that of IRVG, this would seem to indicate that IRVG could have optimised the node reliability of the T-MAC network even further, but refrained from doing so because it would have had too-large-an effect on the node duty cycle. To help understand how all these factors affect the overall performance of the two networks, the *per network reward* (see section 5.1.3) of these networks is shown in the left graph of figure 6.63. This graph shows that, despite the relatively small optimisation in node hop count, for both networks the per network reward achieved by IRVG is still higher than the one achieved for the 'baseline' (Min Virtual Gateways) scenario, which indicates that for the CSMA/CA - T-MAC case both networks benefit from cooperating with one another.

The performance of the individual metrics for the T-MAC - TDMA case is shown in figure 6.64. This figure shows that for both networks, IRVG increases the node reliability at the cost of an increased node duty cycle. As for the CSMA/CA - T-MAC case, the T-MAC network benefits the most from the cooperation between the networks. This should hardly come as a surprise given that the T-MAC network assigns a much higher *weight* to the node reliability metric than the TDMA network (0.6 versus 0.2) and that the increase in node reliability is also more substantial for the T-MAC (from 62% to 77%) network than it is for the TDMA network (from 74% to 79%). In addition, the increased node duty cycle also affects the TDMA network more strongly than it affects the T-MAC

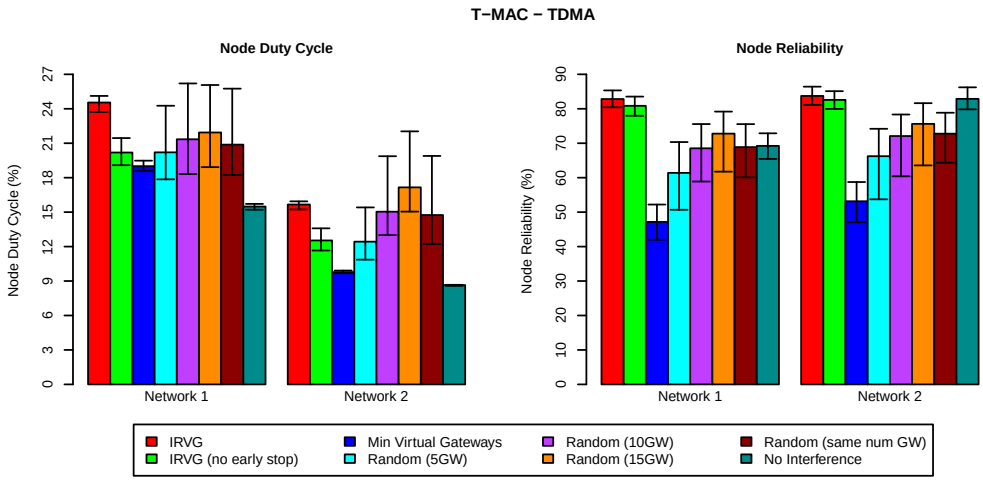


Figure 6.64: Average, 5- and 95%-tile of the node duty cycle and node reliability for the T-MAC - TDMA case. In this particular case, IRVG optimises for both networks the node reliability at the cost of an increase in node duty cycle.

network. This is partially because the TDMA network assigns a higher *weight* to the duty cycle metric (0.8 versus 0.4), but also because the relative increase in node duty cycle is higher for the TDMA network than it is for the T-MAC network (8% to 11% instead of 15% to 19%). It is interesting to note at this point that for the TDMA network, the node duty cycle of IRVG is lower than that of the "Random (same num GW)"-case while the opposite is true for the T-MAC network. This shows that IRVG does manage to somewhat limit the impact on the node duty cycle for the TDMA network (most likely by shifting the majority of the virtual gateways to the T-MAC network). To get an idea of the effect these tradeoffs have on the overall performance of the network involved, the *per network reward* of the respective networks are shown in the right graph of figure 6.63. These show that, in contrast to the CSMA/CA - T-MAC case discussed above, for the T-MAC - TDMA case only the T-MAC network benefits from the cooperation between the networks. For the TDMA network, the per network reward is slightly lower when using IRVG than in the baseline 'Min Virtual Gateways'-scenario which indicates that the performance improvement achieved for the T-MAC network comes at the cost of a slight performance overhead for the TDMA network. It should be noted at this point that the reward function of IRVG both optimises the total performance of the networks and minimises the difference between the performance of the individual networks. As discussed in section 5.1.3, the trade-off between these two optimisation goals is controlled by the β-parameter and while the value used in these tests ($\beta = 0.5$) does allow a certain imbalance between the performance of the two networks to arise, this behaviour can easily be changed depending on the requirements of the specific use case.

6.4.2.2 Number of Iterations

As discussed in section 5.1.2, a new virtual gateway configuration is applied to the networks during each iteration of IRVG. Given that there is a definite cost associated with

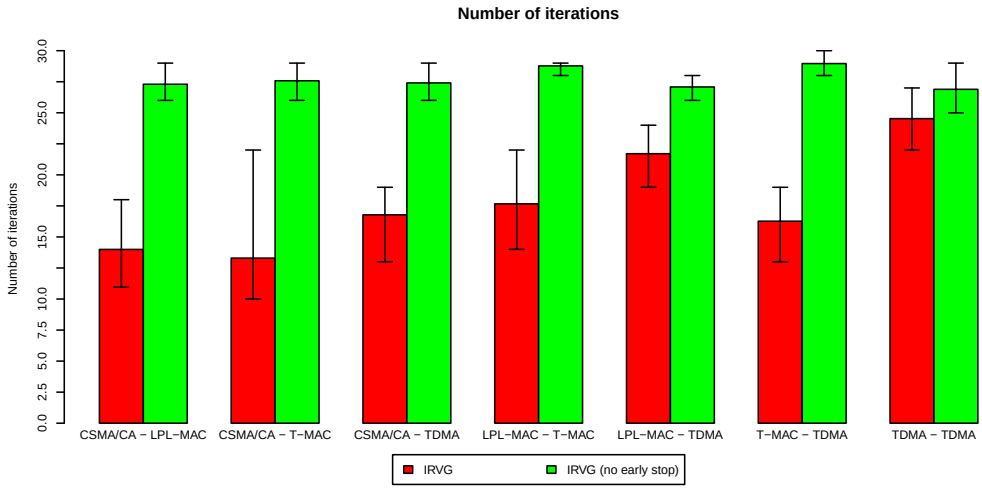


Figure 6.65: Average, 5- and 95%-tile of the number of iterations required by IRVG to complete the optimisation process for each of the combinations of MAC protocols.

doing so, the 'early stop check' of IRVG limits the number of iterations required by halting the optimisation process as soon as the removal of another gateway is no longer predicted to yield any additional increase in performance. Given however that the predictions made by the prediction algorithm are never 100% accurate, this may also cause IRVG to stop the optimisation process too early. Therefore, this section investigates the effect that this 'early stop check' has on the number of iterations required by IRVG as well as the overall performance and the number of gateways in the final configuration.

The number of iterations required for IRVG to finish, both with and without the 'early stop check' enabled, is shown in figure 6.65. While the effect of the 'early stop check' is not as dramatic for the node-to-sink scenario as it is for the random-flows scenario (see section 5.2.2.2), it still has a significant effect on the number of iterations required. When the 'early stop check' is disabled, IRVG requires on average between 27 and 28 iterations to complete. Given that IRVG starts from a *maximum configuration* of 50 nodes, the number of required iterations is thus quite considerable. As with the random-flows scenario, this high number of required iterations is due to the fact that with the 'early stop check' disabled, IRVG will try to remove every single non-redundant gateway that remains after the redundant removal phase. When the 'early stop check' is enabled, the average number of required iterations is, in most cases, between 35% and 50% lower than when this check is disabled. The only two exceptions are the LPL-MAC - TDMA and TDMA - TDMA cases for which the average number of iterations are reduced by respectively 16% and 8%. In the TDMA - TDMA case the reduced effectiveness of the 'early stop check' can be attributed to the fact that node duty cycle is very important for both networks and that IRVG will therefore, on average, remove more virtual gateways than in the other cases. Given that only a single non-redundant gateway is removed per iteration, this also causes the number of iterations to increase. A similar argument can be made for the LPL-MAC - TDMA case. Even though the LPL-MAC only assigns a weight of 0.5 (instead of 0.8) to the node duty cycle metric, this metric is still fairly important to both networks and

as a result the number of iterations needed to complete the optimisation process will also be higher. Finally, it should also be noted that the 'early stop check' is less effective in reducing the number of iterations for the node-to-sink scenario considered here than it is for the random-flows scenario discussed in section 5.2.2.2. This is most likely due to the fact that in the node-to-sink scenario, fewer gateways are needed to achieve the *goals* set for the node reliability and node hop count. This allows IRVG to further optimise the node duty cycle metric. Since this is done by removing more gateways, the number of iterations needed to complete the optimisation process will also be higher.

To allow the effect of 'early stop check' on the overall performance to be evaluated, all the graphs in sections 6.4.1 and 6.4.2 show the performance of IRVG both with and without the 'early check enabled'. For the single-metric optimisation test cases, the performance difference between IRVG with and without the 'early stop check' enabled is in most cases either very minimal or even so small as to be negligible. The only two exceptions are when optimising node reliability in the case that either CSMA/CA or T-MAC is combined with TDMA. In those cases, the difference in performance is a little bit larger but even then the difference in the average improvement of the node reliability is less than 1.5%. When optimising for multiple metrics there is, again, in most cases hardly any difference between the performance of IRVG with and without the 'early stop check' enabled. The only two cases for which the difference in performance is somewhat noticeable are the CSMA/CA - T-MAC and T-MAC - TDMA cases. For the CSMA/CA - TMAC case there is a small but noticeable difference between the *Total Reward* achieved between IRVG with and without the 'early stop check' enabled. When the performance of the individual metrics is considered however it turns out that this difference is still quite small. Using the 'early stop check' only causes the node duty cycle to drop from 19.6% to 18.7% (which is less than 1% difference) while for the node reliability metric the difference is less than 2%. This difference is still very small, especially considering that the average node reliability is above 80%. For the node hop count metric, the 'early stop check' only has a tiny effect on the performance of the CSMA/CA network. For the T-MAC network the difference in node hop count is completely irrelevant given that this network does not optimise for this metric. For the T-MAC - TDMA case, the 'early stop check' only has a noticeable effect on the performance of the node duty cycle metric and even then the difference in performance is quite small. As shown in figure 6.64, the node duty cycle drops from 19.3% to 18% for the T-MAC network while for the TDMA network the average node duty cycle is reduced from 10.9% to 10.2%. For the node reliability metric there is also a tiny difference between the performance of IRVG with and without the 'early stop check' enabled but this difference is negligible.

Finally, the effect of the 'early stop check' on the number of gateways is shown in figure 6.66. In contrast to the random-flows scenario, the effect of the 'early stop check' on the number of gateways in the final configuration varies noticeably with the specific set of MAC protocols used. When either CSMA/CA or LPL-MAC are combined with T-MAC the difference in final number of gateways is very small (on average less than 1 gateway). For the CSMA/CA - LPL-MAC and LPL-MAC - TDMA cases this difference is somewhat larger (on average around 2 gateways) while for the other combinations of MAC protocols the difference between the final configurations generated with and without the 'early stop check' enabled can become quite significant (on average more than 5 gateways in the T-MAC - TDMA case). Despite these large differences however, the 'early stop check'

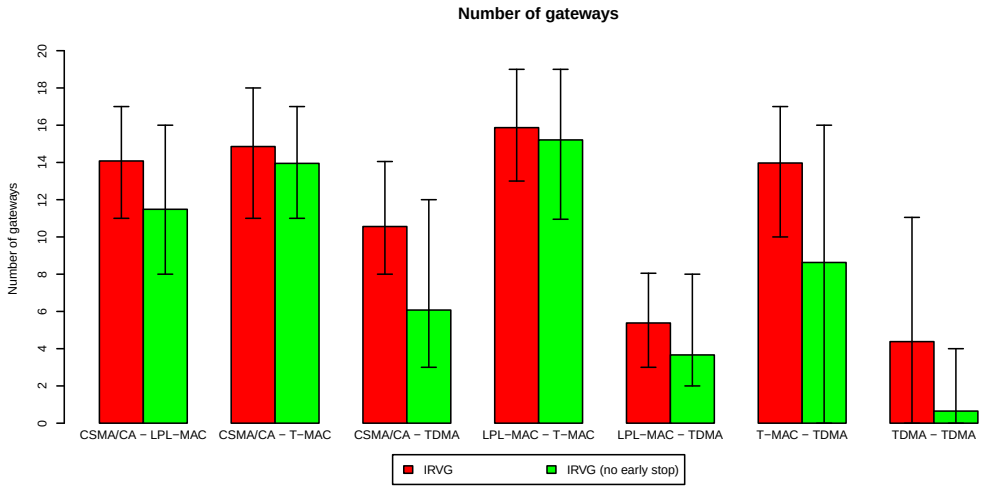


Figure 6.66: Average, 5- and 95%-tile of the number of gateways present in the *final* configurations generated by IRVG.

only has a small effect on the overall performance of the final configurations generated by IRVG. As with the TDMA - TDMA case in the random-flows scenario, this therefore demonstrates that the 'early stop check' does exactly what it is designed to do, namely to stop optimising as soon as there is no further benefit to (trying to) remove any additional gateways.

6.5 Conclusion

The previous two chapters introduced IRVG and evaluated its use in the random-flows scenario. This chapter applied the IRVG mechanism to the node-to-sink scenario. As discussed at the beginning of this chapter, the goal is no longer to enable communication between the networks with the lowest possible performance overhead but instead to optimise the performance of these networks through the use of virtual gateways and the 'sink sharing' mechanism. After first discussing the changes made to the prediction algorithm to allow it to cope with the effects of the 'sink sharing' mechanism (see section 6.1), this modified prediction algorithm was extensively evaluated in section 6.2.

The evaluation of the prediction algorithm showed that, as discussed in section 6.3, the prediction error 'behaves' in mostly the same manner for the node-to-sink scenario as it did in the random-flows scenario. In most cases, the prediction error is at least as low in the node-to-sink scenario as it was in the random-flows scenario and it also reacts in largely the same manner to changes in the MAC protocols used, the type and the number of virtual gateways removed. Despite this, there are some cases in which the worst case prediction error can be considerably higher than in the random-flows scenario. As discussed in section 6.3 however these only required minimal changes to be made to the selection algorithm and should not have too-large-an effect on the performance of IRVG itself.

The selection algorithm of IRVG was evaluated in section 6.4. This evaluation shows that although there are some cases where no improvement in performance can be achieved, the use of IRVG and virtual gateways can in the overall majority of all test cases achieve a significant optimisation in the operation of the networks. Whether or not IRVG is able to optimise the performance as well as the level of optimisation achieved however, depends for a very significant portion on the specific metrics being optimised and thus on the requirements imposed by the networks. This is made clear by the *single metric* optimisation tests discussed in section 6.4.1. When only the node duty cycle is being optimised, IRVG is generally not able to achieve any performance improvement at all and even when it is able to do so the performance gain is relatively small. This however is not because of any limitation of IRVG itself and is solely due to the fact that for the node deployment considered here, the 'radio-on'-time saved by using shorter routing paths generally does not outweigh the energy-wise cost of running a virtual gateway. In contrast to the node duty cycle metric, the use of virtual gateways and IRVG is highly beneficial when optimising for (only) node hop count. For this metric, IRVG is able not only to significantly improve the performance of the networks involved but is in most cases also capable of either matching or even exceeding the performance of the "ideal" Same MAC scenario. For the node reliability metric, the performance gain achieved by IRVG depends on the specific MAC protocols. IRVG yields the best results for the T-MAC and TDMA MAC protocols while for the CSMA/CA and LPL-MAC protocols the achieved performance improvement is noticeably lower. As discussed in section 6.4.1.3 however, this is not due to any limitation of IRVG but rather due to the fact that the 'baseline' node reliability of CSMA/CA and LPL-MAC are already very high, which leaves little room for improvement. The *single metric* tests discussed in section 6.4.1 also reveal that IRVG consistently outperforms any of the randomly generated virtual gateway configurations. As with the random-flows scenario, this shows that IRVG is able to make good use of the information collected from the networks and that there is a definite benefit to using IRVG instead of more primitive selection techniques.

When optimising for multiple metrics, IRVG once again consistently outperforms any of the randomly generated configurations and, despite having to balance between multiple and possibly conflicting network requirements, IRVG is in most test cases also able to achieve a definite improvement in the overall performance of the networks. (The only exception to this is the TDMA - TDMA case where the node duty cycle metric is too important to make the use of virtual gateways worthwhile.) While the actual performance gain achieved depends on the MAC protocols used, and thus also on the requirements of the networks, IRVG is in most cases also able to exceed the performance of the 'No Interference'-scenario, which indicates that the use of virtual gateways and IRVG can result in a better performance than if there were only one network present in the wireless environment. Despite this it should be noted that, at least for the test performed here, using IRVG is not always beneficial to both networks involved. The T-MAC - TDMA case is a prime example of this as in this case the performance of the T-MAC network is optimised at the cost of the TDMA network. As discussed in section 6.4.2.1 however this is due to the fact that the reward function allows, to a certain extent, the total performance to be optimised at the cost of a disparity between the performance of the individual networks. As discussed in section 5.1.3 however, this behaviour can easily be changed by tweaking the β-parameter of the reward function.

When considering the number of iterations required by IRVG, the results shown in section 6.4.2.2 make it clear that, as for the random-flows scenario, it takes a considerable number of iterations to complete the optimisation process when the 'early stop check' is disabled. Despite the 'early stop check' not having as-dramatic an effect on the number of iterations required, it is still capable of significantly reducing the number of iterations required by IRVG while only having a minimal impact on the performance of the final configuration.

The results of chapter 3 already showed that, although virtual gateways offer a viable method of enabling inter-network communication, great care should be taken in selecting the specific nodes to be used as a virtual gateway. The results of chapter 5 subsequently showed that IRVG is able to choose these gateways very carefully for the random-flows scenario. The results of this chapter show this also to be the case for the node-to-sink optimisation scenario. Although virtual gateways are not particularly well suited for optimising the node duty cycle, IRVG is able to significantly optimise both the node hop count and node reliability and is in many cases capable of nearing (or even exceeding) the performance of the idealised 'Same-MAC' scenario. At the same time, IRVG is capable of balancing between the performance improvement achieved for these two metrics and the performance overhead incurred for the node duty cycle metric based on the requirements of the network administrators. Although it is not always possible to improve the performance of the networks this shows that, with the proper support from the rest of the network stack (i.e., the 'sink sharing mechanism'), IRVG is thus also a viable tool for optimising the operation of the networks in the node-to-sink scenario.

CHAPTER 7

Conclusions & Future Work

This thesis presented and evaluated a novel strategy for enabling link level communication between sensor networks using heterogeneous MAC protocols. In contrast to the technique used by sensor network developers to enable interoperability at the routing layer of the network stack (i.e., forcing everyone to use a single standardised protocol) the approach of this thesis is to allow each sensor network to continue to use its own MAC protocol and to use *virtual gateway* nodes (regular nodes running multiple MAC protocols at the same time) to bridge the communication gap between the networks. While, as discussed in section 1.2, this approach has a number of practical advantages it also required three separate research questions to be answered.

The first of these questions is whether or not it is possible for two sensor networks using different (incompatible) MAC protocols to even coexist in the same wireless environment. As discussed in section 1.2, the reason this question needed answering is that there is little point in enabling communication between two MAC-heterogeneous sensor networks if the interference that exists between them has such a detrimental effect on the network performance that they cannot possibly coexist in the same wireless environment. To answer this question, chapter 2 investigated how the performance of these networks is affected, under a number of different traffic conditions, by the interference that exists between their respective MAC protocols. From the performance tests discussed in this chapter it is clear that, as one might expect, the effect of inter-MAC interference on the network performance largely depends on the traffic load and on the number of nodes deployed in the same area (i.e., the node density). While under certain extreme circumstances (high traffic load, large number of nodes deployed in a small area) inter-MAC interference did indeed have a dramatic effect on the performance of the network involved, the tests discussed in chapter 2 also revealed that under less strenuous (and more realistic) conditions, the effect of this interference on the overall network performance is small enough to allow

these networks to co-exist in the same wireless environment without much issue.

The second research question that needed answering is whether or not it is feasible to use (existing) low-power sensor nodes as virtual gateway devices. To answer this question a network stack architecture that allows multiple MAC protocols to be run simultaneously on top of a single radio interface (the 'MultiMAC' stack) was presented in chapter 3. To verify that this architecture can be used on *low power* nodes, it was implemented for and evaluated using the extremely resource constrained Tmote Sky sensor node platform. This evaluation not only revealed that the proposed architecture is flexible and extensible enough to support a wide range of sensor network MAC protocols but also that the performance overhead of supporting multiple MAC protocols is insignificant compared to the performance benefits that can be achieved by being performance-conscious both in the design and implementation of the network stack. The performance evaluation using multiple MAC protocols (see section 3.2.2) did reveal that there is a noticeable duty cycle wise cost associated with configuring a sensor node as a virtual gateway. While this is not unexpected (running multiple MAC protocols causes the radio to be active more of the time), this duty cycle wise cost does need to be taken into account when choosing the (number of) virtual gateway to use. Finally the large-scale tests performed using the w-iLab.t testbed (see section 3.3), revealed that although virtual gateways are a feasible way of enabling link-level communication between MAC-heterogeneous sensor networks, the specific nodes to be configured as a virtual gateway need to be carefully selected based on the requirements of the individual networks.

The final research question to be answered was how to select the nodes to use as a virtual gateway based on the topology and the requirements of the networks. This research question turned out to be somewhat more complicated than the other two and as a result this issue was covered by chapters 4, 5 and 6. To answer this question, this thesis introduced IRVG (Iterative Removal of Virtual Gateways): a heuristic algorithm for the selection of virtual gateway nodes that operates based on topology- and performance information collected from the wireless environment as well as the performance requirements specified by the administrators of the respective networks. As discussed in chapter 4, IRVG operates by configuring all possible nodes as a virtual gateway and then iteratively removing the redundant ones. IRVG consists of two main parts: an algorithm for predicting how the performance of the networks is affected by the removal of one or more gateways and an algorithm for selecting the next gateway(s) to remove based on these predictions.

The prediction algorithm of IRVG was introduced in section 4.2 and its accuracy was evaluated in sections 4.3 and 6.2 for respectively the random-flows and node-to-sink scenario. From these evaluations it is clear that although for some test-scenarios the worst case prediction error can become quite high, for most combinations of MAC protocols the prediction algorithm is able to accurately predict the impact of removing one or more gateways on the performance of the networks. In addition, the accuracy of the prediction algorithm also varies with the combination of MAC protocols used and also depends on the number of gateways being removed (with the prediction being more accurate when fewer gateways are removed at the same time). The graphs in sections 4.3 and 6.2 however, also make it clear that in a few cases there can be occasional spikes in the prediction error of the prediction algorithm. In the end this proved to be not too much of an issue as the selection mechanism of IRVG has been designed to be able to cope with these occasional spikes (see section 5.1.5).

The selection algorithm of IRVG is discussed in section 5.1 and was evaluated for respectively the random-flows and node-to-sink scenario in sections 5.2 and 6.4. For the random-flows scenario the presence of virtual gateways is required for the networks to operate and as a result the main focus of the evaluation was to determine the performance overhead of using the virtual gateway configurations generated by IRVG in comparison to an idealised 'Same MAC' scenario. This evaluation showed that in most cases IRVG is able to match or even exceed the performance of the 'ideal' baseline (Same-MAC) scenario. Moreover, in those cases where IRVG is not quite able to meet the baseline performance, the performance 'overhead' of IRVG is fairly limited, considering that the 'ideal' baseline scenario used for these tests is actually impossible in real life (see section 5.2). In the node-to-sink scenario, IRVG (and virtual gateways) are used solely as a network optimisation technique and as a result the evaluation of IRVG for this scenario focussed on determining the performance gain achieved by the virtual gateway configurations generated by IRVG compared to the baseline case where no virtual gateways are used. The tests discussed in section 6.4 clearly show that IRVG is able to achieve a significant performance improvement when optimising for either node hop count or end-to-end reliability. When only the node duty cycle is being optimised IRVG is not able to achieve a significant performance improvement but, as discussed in section 6.4.1.1, this is due to the specific node deployment used in this thesis and not due to any limitation of IRVG itself. In addition, the tests discussed in sections 5.2 and 6.4 also revealed that for both scenarios IRVG is, within the limits of what is physically possible, able to balance between the performance of different metrics based on the *goals* and *weights* provided by the network administrators.

In summary: the results of chapter 2 showed that, except under extreme conditions, the interference between MAC-heterogeneous has a small enough impact on the network performance to allow these networks to coexist peacefully in the same wireless environment. The results of chapter 3 showed that the use of *virtual gateways* is a feasible way of enabling link-level communication between MAC-heterogeneous sensor networks, but that the selection of which nodes to use as virtual gateways needs to be done very carefully. The results of chapters 4 and 5 show that, for the random-flows scenario, IRVG is able to choose the gateways very carefully since it is able to near or even exceed the performance of the idealised 'Same-MAC' scenario and at the same time is capable of adjusting the selected gateways based on the requirements of the network administrators. Likewise, the results of chapter 6 show that this is also the case for the node-to-sink optimisation scenario. Although virtual gateways are not particularly well suited for optimising the node duty cycle, IRVG is able to significantly optimise both the node hop count and the end-to-end reliability while also keeping the duty cycle wise cost in check. This all shows that using virtual gateways in combination with IRVG is thus a viable method of enabling link-level inter-network communication and cooperation.

Future Work

A number of interesting areas for future work can be identified in the context of the work presented in this thesis.

Firstly, the virtual gateway selection problem offers a number of avenues for future research. The IRVG-algorithm presented in this thesis achieves good results in tuning the

selection of virtual gateway nodes to the requirements of the networks but, given that it is the first algorithm to tackle this specific problem it can best be considered to be a 'baseline' algorithm for other researchers to improve on. For the development of IRVG the choice was for instance made to (1) impose as few constraints on the routing protocol as possible and (2) to use an iterative removal strategy whereby virtual gateways are only ever removed from (and never added to) the wireless environment. One possible avenue for future research would therefore be to include additional information about (and thus impose additional constraints on) the routing protocol used by the networks. Incorporating knowledge about the routing metrics used might for example help the prediction algorithm to better predict the effect of removing a particular gateway on the route topology and thus result in more accurate predictions. In addition this information might also be used to enable the prediction algorithm to predict the effect of adding (rather than removing) a virtual gateway to the wireless environment and thus allow the selection algorithm more freedom in which changes to make to the virtual gateway configuration at each step of the algorithm.

Another area for future research is the network stack architecture for the virtual gateway nodes presented in chapter 3. Currently this architecture has one significant drawback: it only supports single-channel MAC protocols. This limitation is partly due to the fact that all MAC protocols have to share a single radio interface. Since this radio interface can only listen to a single channel at the same time this, essentially, means that all MAC protocols running on the virtual gateway node have to agree on which channel to tune the radio to at any given point in time. Given moreover that, within the scope of this work, the choice was made to isolate the different MAC protocols from one another (i.e., the MAC protocols are not aware of the fact that they have to share the radio interface with other MAC protocols), the only way to do so is to keep the radio channel "fixed" for all the MAC protocols. In recent years however, multi-channel MAC protocols have received renewed interest from sensor network developers and researchers both as a means to tackle crowded channels and as a means of minimising the effects of outside interference. For this reason, it would be interesting to investigate to what extent the virtual gateway approach could be applied to multi-channel MAC protocols by 'breaking' the isolation between the MAC protocols running on the same virtual gateway. Depending on the specific MAC protocols used, it might for instance be possible for the different MAC protocols to coordinate with one another about who has 'ownership' of the radio interface at any given time, but this is of course an open area of research.

It should also be noted that although this thesis focussed solely on the problem of enabling communication between MAC-heterogeneous networks, there are a number of other applications for which the virtual gateway concept may be useful. One possible application would for instance be *single* network optimisation. As discussed in chapters 1 and 2, different sensor network MAC protocols are optimised for different application requirements and also react differently to outside interference. As discussed in [107], it therefore makes sense, from an optimisation point-of-view, to dynamically vary the MAC protocol used based on traffic requirements and prevailing interference conditions (both of which may vary over time). Although [107] try to do just that, they are limited by the fact that all nodes have to use the same MAC protocol and that the MAC protocol used can thus only be varied on a network-wide (global) scale. In this case, the addition virtual gateways would also allow for the MAC protocol to be varied on a local scale (e.g., only change the

MAC protocol in areas experiencing elevated levels of interference) and would thus allow more fine-grained optimisations to be made.

Other applications for virtual gateways can be found in the (relatively) recent trend to extend cloud-computing concepts (such as advanced virtualisation of resources, on-the-fly resource-allocation and support for multi-tenancy) to network infrastructure and devices "beneath" the cloud. Edge Computing is a good example of this. Other examples include Software Defined Networking (which aims to make the management of traditional networks as flexible as that of cloud infrastructure), Network Function Virtualisation (which aims to replace 'traditional' network hardware, such as dedicated firewalls & load balancers, with to cloud-based virtual machines) and (5G) *network-slicing* (which takes the virtualisation principle even further by effectively multiplexing multiple virtualised networks onto a single physical network infrastructure). Although all of these technologies mainly focus on wired and cellular networks, a number of researchers have proposed to also apply these concepts to sensor networks [169]. Virtual gateways can be considered to be an enabling technology in that regard given that they *virtualise* the radio chip of the sensor node to the rest of the network stack. In this thesis, this functionality was only used to share the radio chip between separate MAC protocols but this separation could just as easily be carried through in the rest of the software running on the sensor node, which would essentially allow multiple independent network stacks (each supporting its own set of applications) to be run simultaneously on the same sensor node. This means that virtual gateways may for instance be used to enable network slicing in sensor networks. In addition they may, within the scope of multi-tenancy (cloud-like) in-network computing help to support the differing QoS requirements of different applications running on the same sensor node. Although virtual gateways may thus be used to enable cloud-like architectures in sensor networks it should be noted that doing so brings with it its own set of challenges. As discussed above, the current virtual gateway network stack is limited by the fact that all MAC protocols have to operate in the same channel. In the context of multi-tenancy in-network processing this means that whatever mechanisms are used to meet the QoS requirements need to consider the interactions between the different MAC protocols. Likewise it also means that in the context of network slicing, virtual gateways currently cannot offer perfect isolation between the different slices. This means that, as was the case for the interoperability problem that was investigated in this thesis, interference between different MAC protocols will have to be considered by whatever control mechanisms are used to manage such "cloud-like" sensor networks.

Bibliography

[1] Richard T. Lacoss. Distributed Sensor Networks. Final report ada204719, Massachusetts Institute of Technology, Lexington Lincoln Lab, September 1986. `https://apps.dtic.mil/docs/citations/ADA204719`.

[2] Chee-Yee Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.

[3] Zolertia RE-Mote platform. `https://github.com/Zolertia/Resources/wiki/RE-Mote`. Accessed: 2019-08-07.

[4] Byungrak Son, Yomg-sork Her, and Jung-Gyu Kim. A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea Mountains. *International Journal of Computer Science and Network Security*, 6(9):124–130, September 2006.

[5] Jirapon Sunkpho and Chaiwat Ootamakorn. Real-time flood monitoring and warning system. *Songklanakarin Journal of Science & Technology*, 33(2), March 2011.

[6] Juan López, Fulgencio Soto, Pedro Sánchez, Andrés Iborra, Juan Suardiaz, and Juan Vera. Development of a sensor node for precision horticulture. *Sensors*, 9(5):3240–3255, 2009.

[7] Antonio-Javier Garcia-Sanchez, Felipe Garcia-Sanchez, Fernando Losilla, Pawel Kulakowski, Joan Garcia-Haro, Alejandro Rodríguez, José-Vicente López-Bao, and Francisco Palomares. Wireless Sensor Network Deployment for Monitoring Wildlife Passages. *Sensors*, 10(8):7236–7262, 2010.

[8] Shiyong Wang, Jiafu Wan, Di Li, and Chunhua Zhang. Implementing Smart Factory of Industrie 4.0: An Outlook. *International Journal of Distributed Sensor Networks*, 12(1):10, 2016.

[9] Markus Becker, Bernd-Ludwig Wenning, Carmelita Görg, Reiner Jedermann, and Andreas Timm-Giel. Logistic Applications with Wireless Sensor Networks. In *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*, HotEmNets '10, pages 6:1–6:5, 2010.

[10] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.4-2003*, pages 1–680, October 2003.

[11] ZigBee Specification. `https://www.zigbee.org/download/standards-zigbee-specification/`. Accessed: 2019-08-07.

[12] Intrinsically Secure WirelessHART Field Device Networks and the Industrial Internet of Things (IIoT). Technical report, Fieldcomm Group, June 2017. `https://www.fieldcommgroup.org/sites/default/files/technologies/hart/WirelessHART%20security%20v1.0.pdf`.

[13] Maarten Weyn, Glenn Ergeerts, Raf Berkvens, Bartosz Wojciechowski, and Yordan Tabakov. DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication. In *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 54–59, October 2015.

[14] Z-Wave Alliance. `https://z-wavealliance.org/`. Accessed: 2019-08-07.

[15] Can Tunca, Sinan Isik, Mehmet Yunus Donmez, and Cem Ersoy. Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys Tutorials*, 16(2):877–897, October 2014.

[16] Tommaso Melodia, Mehmet C. Vuran, and Dario Pompili. The State of the Art in Cross-Layer Design for Wireless Sensor Networks. In *Wireless Systems and Network Architectures in Next Generation Internet*, pages 78–92, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[17] Khaled Arisha, Moustafa Youssef, and Mohamed Younis. *Energy-Aware TDMA-Based MAC for Sensor Networks*, pages 21–40. Springer US, Boston, MA, 2002.

[18] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2):70–87, April 2007.

[19] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 56–67, New York, NY, USA, 2000. ACM.

[20] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325 – 349, 2005.

[21] Wnjing Guo and Wei Zhang. A survey on intelligent routing protocols in wireless sensor networks. *Journal of Network and Computer Applications*, 38:185 – 201, 2014.

[22] Gabriel Montenegro, Jonathan Hui, David Culler, and Nandakishore Kushalnagar. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, September 2007.

[23] N. Sornin and A. Yegin. LoRaWAN 1.1 Specification. Technical report, LoRa Alliance, October 2017. https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf.

[24] Sigfox Radio Technology Keypoints. https://www.sigfox.com/en/sigfox-iot-radio-technology. Accessed: 2019-08-08.

[25] Standardization of NB-IOT completed. https://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete. Accessed: 2019-08-08.

[26] Pascal Thubert and Jonathan Hui. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, September 2011.

[27] Carsten Bormann, Zach Shelby, Samita Chakrabarti, and Erik Nordmark. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). RFC 6775, November 2012.

[28] Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, March 2012.

[29] Zach Shelby, Klaus Hartke, and Carsten Bormann. The Constrained Application Protocol (CoAP). RFC 7252, June 2014.

[30] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Commun. ACM*, 43(5):51–58, May 2000.

[31] Archived Symbionets project website. https://web.archive.org/web/20130613193401/http://symbionets.intec.ugent.be/. Accessed: 2019-08-08.

[32] Eli De Poorter, Benoît Latré, Ingrid Moerman, and Piet Demeester. Symbiotic Networks: Towards a New Level of Cooperation Between Wireless Networks. *Wireless Personal Communications*, 45(4):479–495, June 2008.

[33] Maher Abdelshkour. IoT, from Cloud to Fog Computing. https://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing, March 2015. Accessed: 2019-08-09.

[34] Weisong Shi, Jie Cao, Zhang Quan, Youhuizi Li, and Lanyu Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, October 2016.

[35] Mahadev Satyanarayanan. The Emergence of Edge Computing. *Computer*, 50(1):30–39, January 2017.

[36] Bengt Ahlgren, Markus Hidell, and Edith C.-H. Ngai. Internet of Things for Smart Cities: Interoperability and Open Data. *IEEE Internet Computing*, 20(6):52–56, November 2016.

[37] Sergios Soursos, Ivana Podvar Žarko, Patrick Zwickl, Ivan Gojmerac, Giuseppe Bianchi, and Gino Carrozzo. Towards the cross-domain interoperability of IoT platforms. In *2016 European Conference on Networks and Communications (EuCNC)*, pages 398–402, June 2016.

[38] Mahda Noura, Mohammed Atiquzzaman, and Martin Gaedke. Interoperability in Internet of Things: Taxonomies and Open Challenges. *Mobile Networks and Applications*, 24(3):796–809, June 2019.

[39] Research & Innovation in Internet of Things. `https://ec.europa.eu/digital-single-market/en/research-innovation-iot`, June 2019. Accessed: 2019-08-09.

[40] Ivan Gojmerac, Peter Reichl, Ivana Podnar Žarko, and Sergios Soursos. Bridging IoT islands: the symbIoTe project. *e & i Elektrotechnik und Informationstechnik*, 133(7):315–318, November 2016.

[41] Giancarlo Fortino, Claudio Savaglio, Carlos E. Palau, Jara Suarez de Puga, Maria Ganzha, Marcin Paprzycki, Miguel Montesinos, Antonio Liotta, and Miguel Llop. *Towards Multi-layer Interoperability of Heterogeneous IoT Platforms: The INTER-IoT Approach*, pages 199–232. Springer International Publishing, Cham, 2018.

[42] AGILE IoT Project Website. `http://agile-iot.eu/about/`. Accessed: 2019-08-09.

[43] Aref Meddeb. Internet of things standards: who stands out from the crowd? *IEEE Communications Magazine*, 54(7):40–47, July 2016.

[44] Erno Kovacs, Martin Bauer, Jaeho Kim, Jaeseok Yun, Franck Le Gall, and Mengxuan Zhao. Standards-Based Worldwide Semantic Interoperability for IoT. *IEEE Communications Magazine*, 54(12):40–46, December 2016.

[45] Pratikkumar Desai, Amit Sheth, and Pramod Anantharam. Semantic Gateway as a Service Architecture for IoT Interoperability. In *2015 IEEE International Conference on Mobile Services*, pages 313–319, June 2015.

[46] Hasan Derhamy, Jens Eliasson, and Jerker Delsing. IoT Interoperability.On-Demand and Low Latency Transparent Multiprotocol Translator. *IEEE Internet of Things Journal*, 4(5):1754–1763, October 2017.

[47] G. Aloi, G. Caliciuri, G Fortino, R. Gravina, P Pace, W Russo, and C. Savaglio. A Mobile Multi-Technology Gateway to Enable IoT Interoperability. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 259–264, April 2016.

[48] Shang Guoqiang, Chen Yanming, Zuo Chao, and Zhu Yanxu. Design and Implementation of a Smart IoT Gateway. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 720–723, August 2013.

[49] Santi Nuratch. The IIoT devices to cloud gateway design and implementation based on microcontroller for real-time monitoring and control in automation systems. In *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 919–923, June 2017.

[50] MoteIv. *Tmote Sky Low Power Wireless Sensor Module*. `https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf`.

[51] I. Howitt and J. A. Gutierrez. IEEE 802.15.4 low rate - wireless personal area network coexistence issues. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, volume 3, pages 1481–1486 vol.3, March 2003.

[52] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai. Harmful Coexistence Between 802.15.4 and 802.11: A Measurement-based Study. In *2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, pages 1–6, May 2008.

[53] A. Sikora and V. F. Groza. Coexistence of IEEE 802.15.4 with other Systems in the 2.4 GHz-ISM-Band. In *2005 IEEE Instrumentationand Measurement Technology Conference Proceedings*, volume 3, pages 1786–1791, May 2005.

[54] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving Wi-fi Interference in Low Power ZigBee Networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 309–322. ACM, 2010.

[55] W. Yuan, X. Wang, and J. M. G. Linnartz. A Coexistence Model of IEEE 802.15.4 and IEEE 802.11b/g. In *2007 14th IEEE Symposium on Communications and Vehicular Technology in the Benelux*, pages 1–5, November 2007.

[56] S. Pollin, M. Ergen, M. Timmers, A. Dejonghe, L. van der Perre, F. Catthoor, I. Moerman, and A. Bahai. Distributed cognitive coexistence of 802.15.4 with 802.11. In *2006 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 1–5, June 2006.

[57] Lieven Tytgat, Opher Yaron, Sofie Pollin, Ingrid Moerman, and Piet Demeester. Avoiding collisions between IEEE 802.11 and IEEE 802.15.4 through coexistence aware clear channel assessment. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):137, April 2012.

[58] Carlo Alberto Boano, Thiemo Voigt, Nicolas Tsiftes, Luca Mottola, Kay Römer, and Marco Antonio Zúñiga. Making Sensornet MAC Protocols Robust against Interference. In Jorge Sá Silva, Bhaskar Krishnamachari, and Fernando Boavida, editors, *Wireless Sensor Networks*, pages 272–288, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[59] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM.

[60] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, April 2006.

[61] Texas Instruments. *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. `http://www.ti.com/lit/gpn/cc2420`.

[62] W. L. Lee, A. Datta, and R. Cardell-Oliver. FlexiMAC: A flexible TDMA-based MAC protocol for fault-tolerant and energy-efficient wireless sensor networks. In *2006 14th IEEE International Conference on Networks*, volume 2, pages 1–6, September 2006.

[63] Jianlin Mao, Zhiming Wu, and Xing Wu. A TDMA scheduling scheme for many-to-one communications in wireless sensor networks. *Computer Communications*, 30(4):863 – 872, 2007. Nature-Inspired Distributed Computing.

[64] Sandeep S. Kulkarni and Mahesh Arumugam. SS-TDMA: A self-stabilizing MAC for sensor networks. In *Sensor networks operations*. IEEE Press, 2006.

[65] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 10 pp. vol.2–, January 2000.

[66] Guangyu Pei and Charles Chien. Low power TDMA in large wireless sensor networks. In *2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No.01CH37277)*, volume 1, pages 347–351 vol.1, October 2001.

[67] Tim Nieberg, Stefan Dulman, Paul Havinga, Lodewijk van Hoesel, and Jian Wu. *Collaborative Algorithms for Communication in Wireless Sensor Networks*, pages 271–294. Springer US, Boston, MA, 2003.

[68] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks. *Wireless Communications and Mobile Computing*, 7(7):863–875, 2007.

[69] Kurtis Kredo and Prasant Mohapatra. Medium access control in wireless sensor networks. *Computer Networks*, 51(4):961 – 994, 2007.

[70] L. Kleinrock and F. Tobagi. Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, December 1975.

[71] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999) - Redline*, pages 1–1238, June 2007.

[72] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. *TinyOS: An Operating System for Sensor Networks*, pages 115–148. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[73] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462, November 2004.

[74] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, and Mihail L. Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Trans. Netw.*, 16(3):511–524, June 2008.

[75] We Yei, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, June 2004.

[76] Tijs van Dam and Koen Langendoen. An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM.

[77] P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. In *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, volume 3, pages 1534–1539 Vol.3, March 2004.

[78] T. Zheng, S. Radhakrishnan, and V. Sarangan. PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. In *19th IEEE International Parallel and Distributed Processing Symposium*, pages 8 pp.–, April 2005.

[79] Zhenzhen Liu and Itamar Elhanany. RL-MAC: a Reinforcement Learning Based MAC Protocol for Wireless Sensor Networks. *International Journal of Sensor Networks*, 1(3/4):117–124, January 2006.

[80] S. Mehtha and K.S Kwak. H-MAC: A Hybrid MAC Protocol for Wireless Sensor Networks. *International Journal of Computer Networks and Communications*, 2:108–117, 2010.

[81] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 99(1):70–80, January 2002.

[82] A. El-Hoiydi and J. . Decotignie. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No.04TH8769)*, volume 1, pages 244–251 Vol.1, July 2004.

[83] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 95–107, New York, NY, USA, 2004. ACM.

[84] Philipp Hurni and Torsten Braun. MaxMAC: A Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks. In *Wireless Sensor Networks*, pages 289–305, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[85] Adam Dunkels. The ContikiMAC Radio Duty Cycling Protocol. Technical Report T2011:13, SICS, Kista, Sweden, December 2011.

[86] Yao-Win Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5):1085–1099, May 2005.

[87] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained Network Time Synchronization Using Reference Broadcasts. *SIGOPS Oper. Syst. Rev.*, 36(SI):147–163, December 2002.

[88] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. Timing-sync Protocol for Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 138–149, New York, NY, USA, 2003. ACM.

[89] Jana van Greunen and Jan Rabaey. Lightweight Time Synchronization for Sensor Networks. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, pages 11–19, New York, NY, USA, 2003. ACM.

[90] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The Flooding Time Synchronization Protocol. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 39–49, New York, NY, USA, 2004. ACM.

[91] Wen-Long Chin and Jiun-Lin Tzen. Clock synchronization for energy-constrained wireless sensor networks. In *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1371–1375, September 2009.

[92] R. Tjoa, K. L. Chee, P. K. Sivaprasad, S. V. Rao, and J. G. Lim. Clock drift reduction for relative time slot TDMA-based sensor networks. In *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)*, volume 2, pages 1042–1047 Vol.2, September 2004.

[93] Fasika Assegei. Decentralised Frame Synchronization of a TDMA-based Wireless Sensor Network. Master's thesis, Eindhoven University of Technology, Den Dolech 2, 5612 AZ Eindhoven, The Netherlands, August 2008.

[94] Ted Herman and Sébastien Tixeuil. A Distributed TDMA Slot Assignment Algorithm for Wireless Sensor Networks. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 45–58, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[95] Sinem Coleri Ergen and Pravin Varaiya. TDMA Scheduling Algorithms for Sensor Networks. Technical report, Department of Electrical Engineering and Computer Sciences, University of California, July 2005.

[96] I. Rhee, A. Warrier, J. Min, and L. Xu. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 8(10):1384–1396, October 2009.

[97] Lichun Bao and J. J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 210–221, New York, NY, USA, 2001. ACM.

[98] Jing Li and Georgios Y. Lazarou. A Bit-map-assisted Energy-efficient MAC Scheme for Wireless Sensor Networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, IPSN '04, pages 55–60, New York, NY, USA, 2004. ACM.

[99] Wim Torfs and Chris Blondia. Binary TDMA scheduler by means of egyptian fractions for real-time WSNs on TMotes. In *2010 The 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–8, June 2010.

[100] Bart Braem, Benôit Latré, Ingrid Moerman, Chris Blondia, and Piet Demeester. The Wireless Autonomous Spanning tree Protocol for Multihop Wireless Body Area Networks. In *2006 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops*, pages 1–8, July 2006.

[101] S. S. Kulkarni. TDMA service for sensor networks. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, pages 604–609, March 2004.

[102] L.F.W Van Hoesel and P.J.M Havinga. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks. In *First International Workshop on Networked Sensing Systems (INSS2004)*, June 2004.

[103] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, October 2000.

[104] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks. *Wirel. Netw.*, 12(1):63–78, February 2006.

[105] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher. RID: radio interference detection in wireless sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 891–901 vol. 2, March 2005.

[106] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low Duty Cycle MAC with Scheduled Channel Polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 321–334, New York, NY, USA, 2006. ACM.

[107] Mo Sha, Rahav Dor, Gregory Hackmann, Chenyang Lu, Tae-Suk Kim, and Taerim Park. Self-Adapting MAC Layer for Wireless Sensor Networks. In *2013 IEEE 34th Real-Time Systems Symposium*, volume 75, pages 192–201. IEEE, December 2013.

[108] Yi Chu, Paul D. Mitchell, and David Grace. ALOHA and Q-Learning based medium access control for Wireless Sensor Networks. In *2012 International Symposium on Wireless Communication Systems (ISWCS)*, pages 511–515. IEEE, August 2012.

[109] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018. `http://incompleteideas.net/book/RLbook2018.pdf`.

[110] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and

Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pages 1–320, September 2006.

[111] IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, September 2011.

[112] IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, April 2016.

[113] IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, April 2012.

[114] Qin Wang, Xavier Vilajosana, and Thomas Watteyne. 6TiSCH Operation Sublayer (6top) Protocol (6P). RFC 8480, November 2018.

[115] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 337–350, New York, NY, USA, 2015. ACM.

[116] Tengfei Chang, Thomas Watteyne, Qin Wang, and Xavier Vilajosana. LLSF: Low Latency Scheduling Function for 6TiSCH Networks. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 93–95, May 2016.

[117] Thang Phan Duy, Thanh Dinh, and Younghan Kim. Distributed cell selection for scheduling function in 6TiSCH networks. *Computer Standards & Interfaces*, 53:80 – 88, 2017.

[118] Glenn Daneels, Bart Spinnewyn, Steven Latr.., and Jeroen Famaey. ReSF: Recurrent Low-Latency Scheduling in IEEE 802.15.4e TSCH networks. *Ad Hoc Networks*, 69:100–114, 2018.

[119] Katina Kralevska, Dimitrios J. Vergados, Yuming Jiang, and Angelos Michalas. A Load Balancing Algorithm for Resource Allocation in IEEE 802.15.4e Networks. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 675–680, March 2018.

[120] Athanassios Boulis. Castalia: Revealing Pitfalls in Designing Distributed Algorithms in WSN. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 407–408, New York, NY, USA, 2007. ACM.

[121] H. N. Pham, D. Pediaditakis, and A. Boulis. From Simulation to Real Deployments in WSN and Back. In *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6, June 2007.

[122] András Varga and Rudolf Hornig. An Overview of the OMNeT++ Simulation Environment. In *Proceedings of the 1st International Conference on Simulation Tools and*

Techniques for Communications, Networks and Systems & Workshops, Simutools '08, pages 60:1–60:10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[123] Marco Zuniga and Bhaskar Krishnamachari. Analyzing the transitional region in low power wireless links. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, pages 517–526, October 2004.

[124] Daniel van den Akker, Bart Braem, and Chris Blondia. On the Effects of Interference between Heterogeneous Sensor Network MAC Protocols. *Mobile Ad-Hoc and Sensor Systems, IEEE International Conference on*, 0:560–569, 2011.

[125] Athanassios Boulis. Castalia - A simulator for Wireless Sensor Networks and Body Area Networks - Version 3.2 - User's Manual. Technical report, NICTA, March 2011. `https://github.com/boulis/Castalia/blob/3.2/Castalia-UserManual.doc`.

[126] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. *Wirel. Netw.*, 11(4):419–434, July 2005.

[127] Pascal Thubert. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). RFC 6552, March 2012.

[128] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, page 14, New York, New York, USA, 2003. ACM Press.

[129] Alec Woo and David Culler. Evaluation of Efficient Link Reliability Estimators for Low-Power Wireless Networks. Technical report, U.C. Berkeley Computer Science Division, September 2003.

[130] M. Younis, M. Youssef, and K. Arisha. Energy-aware routing in cluster-based sensor networks. In *Proceedings. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pages 129–136, October 2002.

[131] K. Akkaya and M. Younis. An energy-aware QoS routing protocol for wireless sensor networks. In *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, pages 710–715, May 2003.

[132] Ruay-Shiung Chang and Chia-Jou Kuo. An energy efficient routing mechanism for wireless sensor networks. In *20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06)*, volume 2, pages 5 pp.–, April 2006.

[133] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No.02TH8609)*, volume 1, pages 350–355 vol.1, March 2002.

[134] Tian He, J. A. Stankovic, Chenyang Lu, and T. Abdelzaher. SPEED: a stateless protocol for real-time communication in sensor networks. In *23rd International Conference on Distributed Computing Systems, 2003. Proceedings.*, pages 46–55, May 2003.

[135] P. K. K. Loh, S. H. Long, and Y. Pan. An efficient and reliable routing protocol for wireless sensor networks. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 512–516, June 2005.

[136] J. Teo, Y. Ha, and C. Tham. Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming. *IEEE Transactions on Mobile Computing*, 7(9):1124–1137, September 2008.

[137] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[138] Klaus Hartke. Observing Resources in the Constrained Application Protocol (CoAP). RFC 7641, September 2015.

[139] Daniel van den Akker. Result dataset of the PhD thesis "Enabling interoperability between MAC-heterogeneous sensor networks". `https://zenodo.org/record/3687155`, February 2020.

[140] Stefan Bouckaert, Wim Vandenberghe, Bart Jooris, Ingrid Moerman, and Piet Demeester. The w-iLab.t Testbed. In *Testbeds and Research Infrastructures. Development of Networks and Communities*, pages 145–154, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[141] JL Wong, Roozbeh Jafari, and Miodrag Potkonjak. Gateway placement for latency and energy efficient data aggregation. *Local Computer Networks, 2004. 29th Annual International Conference on*, 0:490 – 497, 2004.

[142] S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, volume 1, pages 377–381. IEEE, 2003.

[143] Alon Efrat, S Har-Peled, and JSB Mitchell. Approximation algorithms for two optimal location problems in sensor networks. *Broadband networks, 2005. 2nd International Conference on*, 0:714 – 723, 2005.

[144] Xu Xu and Weifa Liang. Placing Optimal Number of Sinks in Sensor Networks for Network Lifetime Maximization. *2011 IEEE International Conference on Communications (ICC)*, pages 1–6, June 2011.

[145] W. Youssef and M. Younis. Intelligent Gateways Placement for Reduced Data Latency in Wireless Sensor Networks. *2007 IEEE International Conference on Communications*, pages 3805–3810, June 2007.

[146] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward. Gateway Placement Optimization in Wireless Mesh Networks With QoS Constraints. *IEEE Journal on Selected Areas in Communications*, 24(11):2127–2136, November 2006.

[147] Maolin Tang. Gateways Placement in Backbone Wireless Mesh Networks. *Int'l J. of Communications, Network and System Sciences*, 02(1):44–50, 2009.

[148] Fan Li, Yu Wang, Xiang-Yang Li, Ashraf Nusairat, and Yanwei Wu. Gateway Placement for Throughput Optimization in Wireless Mesh Networks. *Mobile Networks and Applications*, 13(1-2):198–211, March 2008.

[149] Ping Zhou, Xudong Wang, B. S. Manoj, and Ramesh Rao. On Optimizing Gateway Placement for Throughput in Wireless Mesh Networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:7:1–7:12, January 2010.

[150] Antonio Capone, Matteo Cesana, Danilo De Donno, and Ilario Filippini. Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach. *Computer Communications*, 33(10):1151 – 1161, 2010.

[151] Raghavendra V Kulkarni, Senior Member, Anna Förster, and Ganesh Kumar Venayagamoorthy. Computational Intelligence in Wireless Sensor Networks : A Survey. *Communications Surveys & Tutorials, IEEE*, 13(1):68–96, 2011.

[152] Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, and Hwee-pink Tan. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. *IEEE Communications Surveys & Tutorials*, 16(4):1996–2018, 2014.

[153] Anna Forster and Amy L. Machine Learning across the WSN Layers. In *Emerging Communications for Wireless Sensor Networks*, pages 165–183. InTech, February 2011.

[154] Yong Wang, Margaret Martonosi, and Li-Shiuan Peh. A supervised learning approach for routing optimizations in wireless sensor networks. In *Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality - REALMAN '06*, page 79, New York, New York, USA, 2006. ACM Press.

[155] Jenn-long Liu and Chinya V Ravishankar. LEACH-GA: Genetic Algorithm-BasedEnergy-Efficient Adaptive Clustering Protocolfor Wireless Sensor Networks. *International Journal of Machine Learning and Computing*, 1(1):79–85, 2011.

[156] Mohamed Elhoseny, Xiaohui Yuan, Zhengtao Yu, Cunli Mao, Hamdy K. El-Minir, and Alaa Mohamed Riad. Balancing energy consumption in heterogeneous wireless sensor networks using genetic algorithm. *IEEE Communications Letters*, 19(12):2194–2197, 2015.

[157] Suat Özdemir, Bara'A A. Attea, and Önder A. Khalil. Multi-objective evolutionary algorithm based on decomposition for energy efficient coverage in wireless sensor networks. *Wireless Personal Communications*, 71(1):195–215, 2013.

[158] Sajid Hussain, Abdul W. Matin, and Obidul Islam. Genetic Algorithm for Energy Efficient Clusters in Wireless Sensor Networks. *Fourth International Conference on Information Technology (ITNG'07)*, pages 147–154, 2007.

[159] Julio Barbancho, Carlos León, F. Javier Molina, and Antonio Barbancho. A new QoS routing algorithm based on self-organizing maps for wireless sensor networks. *Telecommunication Systems*, 36(1-3):73–83, 2007.

[160] Tatiana Bokareva, Nirupama Bulusu, and Sanjay Jha. Learning sensor data characteristics in unknown environments. *2006 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous*, 2006.

[161] Raghavendra V Kulkarni and Ganesh K Venayagamoorthy. Neural network based secure media access control protocol for wireless sensor networks. In *2009 International Joint Conference on Neural Networks*, pages 1680–1687. IEEE, June 2009.

[162] Yu-Ju Shen and Ming-Shi Wang. Broadcast scheduling in wireless sensor networks using fuzzy Hopfield neural network. *Expert Systems with Applications*, 34(2):900–907, February 2008.

[163] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues. *Journal of Network and Computer Applications*, 35(1):253–267, January 2012.

[164] Ping Wang and Ting Wang. Adaptive Routing for Sensor Networks using Reinforcement Learning. In *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*, pages 219–219. IEEE, September 2006.

[165] Ying Zhang and Qingfeng Huang. A Learning-based Adaptive Routing Tree for Wireless Sensor Networks. *Journal of Communications*, 1(2):12–21, May 2006.

[166] Tiansi Hu and Yunsi Fei. QELAR: A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks. *IEEE Transactions on Mobile Computing*, 9(6):796–809, Jun 2010.

[167] Milos Rovcanin, Eli De Poorter, Ingrid Moerman, and Piet Demeester. A reinforcement learning based solution for cognitive network cooperation between co-located, heterogeneous wireless sensor networks. *Ad Hoc Networks*, 17:98–113, Jun 2014.

[168] Milos Rovcanin, Eli De Poorter, Daniel van den Akker, Ingrid Moerman, Piet Demeester, and Chris Blondia. Experimental validation of a reinforcement learning based approach for a service-wise optimisation of heterogeneous wireless sensor networks. *Wireless Networks*, 21(3):931–948, April 2015.

[169] Nikos Bizanis and Fernando A. Kuipers. SDN and Virtualization Solutions for the Internet of Things: A Survey. *IEEE Access*, 4:5591–5606, September 2016.