ORCHESTRA : supercharging wireless backhaul networks through multi-technology management

Journal of Network and Systems Management manuscript No.
(will be inserted by the editor)

ORCHESTRA: Supercharging wireless backhaul networks through multi-technology management

Tom De Schepper[1] · Patrick Bosch · Jakob Struye[1] · Carlos Donato · Jeroen Famaey[1] · Steven Latré[1]

Received: date / Accepted: date

Abstract Today's and tomorrow's networks are becoming increasingly complex and heterogeneous with a large diversity of devices and technologies. To meet growing demand, and support client mobility there is need for intelligent mechanisms like multi-technology load balancing and handovers. Current solutions, like Multipath Transmission Control Protocol (MPTCP), fail to provide a fine-grained, coordinated, and transparent answer to this heterogeneity, while the lower layers of the Open Systems Interconnection stack simply ignore it by providing full separation of layers. Therefore, we introduce ORCHESTRA, a data link layer framework for the management of multi-technology networks and devices, enabling packet-level dynamic handovers, load balancing, and duplication across network technologies. The framework is the first of its kind in providing fine-grained packet-level control across different technologies in a network-wide manner. Moreover, it works on top of existing standards without the need for hardware changes. This is achieved through a fully transparent virtual Medium Access Control layer and a Software-Defined Networking controller with global intelligence. The framework is implemented in a prototype running on off-the-shelf hardware and we demonstrate its features across different IEEE 802.11 technologies and 4G (Long Term Evolution). We demon-

✉ Tom De Schepper
E-mail: tom.deschepper@uantwerpen.be

Jakob Struye
E-mail: jakob.struye@uantwerpen.be

Jeroen Famaey
E-mail: jeroen.famaey@uantwerpen.be

Steven Latré
E-mail: steven.latre@uantwerpen.be

[1]University of Antwerp - imec
IDLab - Department of Computer Science
Sint-Pietersvliet 7, 2000 Antwerp, Belgium

strate that ORCHESTRA outperforms MPTCP and allows for real-time inter-technology handovers and that overall throughput and reliability are improved in wireless networks.

Keywords Heterogeneous network orchestration · Multi-technology handover · Virtual MAC layer · Load balancing

1 Introduction

Over the years, there has been a tremendous increase in data communication and in the capacity of wireless communication systems. For instance, monthly global mobile data traffic will reach 50 exabytes in 2021, while the number of connected devices is expected to at least reach 28.5 billion by 2022 [1, 2]. More and more (wireless) networking technologies have been developed and added to a plethora of consumer and infrastructure devices to cope with this explosive increase in traffic. In home networks, we can find Ethernet and a mix of wireless technologies like IEEE 802.11 (Wi-Fi), Bluetooth, and ZigBee. Similarly, satellite, Long-Term Evolution (LTE), and 802.11 technologies are being embedded, among others, into smart vehicles. This growth and diversity among technologies and devices will further increase with technological advancements and new technologies being standardized (e.g., IEEE 802.11ay and 802.11ax) [2, 3]. However, each of these technologies operates independently, isolated from each other. Cooperation between them is infeasible due to the current design of the lower layers of the Open Systems Interconnection (OSI) stack [4, 5, 6]. Uneven load distribution among these wireless technologies and spectrum scarcity leads to sub-optimal and inefficient use of the wireless resources. With the ever-increasing traffic demands, however, efficient use and management of wireless resources would allow devices to fully exploit the diversity of wireless access technologies and provide services with guaranteed Quality of Service (QoS) in terms of throughput, delay, and reliability. Therefore, interest has grown in multi-technology cooperation for efficient use of the wireless resources [6, 7]. Coordinated management of these heterogeneous, mostly wireless, environments is highly challenging, as each technology has specific capabilities and serves specific use cases [8, 9].

To address this problem, several solutions have been proposed on different layers of the network stack. Arguably, the most popular solution nowadays is Multipath Transmission Control Protocol (MPTCP), can load balance Transmission Control Protocol (TCP) flows across multiple network interfaces [10, 11, 12]. However, MPTCP lacks intelligence and works only between endpoints. IEEE 1905.1 is a data link layer solution, which allows dynamic flow redirection through a virtual Medium Access Control (MAC) layer [13]. However, the downsides of IEEE 1905.1 are twofold. First, it only provides flow level control, while packet level control is needed for, among others, fine-grained load balancing that fully utilizes all wireless capacity. Second, it was designed for specific network technologies (e.g., Ethernet and Wi-Fi) and is therefor not technology-agnostic and does not support mobile networks. In

contrast, a solution for mobile networks does exist in the form of LTE-Wireless Local Area Network Aggregation (LWA) [14, 15, 16]. This allows the simultaneous use of LTE and Wi-Fi by using Wi-Fi as another medium of transmission for LTE. To be compliant with the LTE standard, tunnels are used over Wi-Fi to transport data to the gateway. There is, however, a major downsides to this approach, as it is technology specific, supporting only Wi-Fi and LTE. Extending it to more technologies would require a revision of the 3GPP standard. In conclusion, none of these approaches can tackle the problem in its entirety, and a more generalized and technology-independent solution is needed.

To cope with the above mentioned heterogeneity in wireless networks we introduce an inter-technology management framework, called ORCHESTRA. The framework consists of two components: a Virtual MAC (VMAC) layer and a centralized controller. The VMAC offers a single connection point to the upper layers, while transparently bonding over the underlying network technologies and supports various operations at packet-level granularity. The three major features are: (i) seamless vertical handovers between technologies, (ii) load balancing across several technologies, and (iii) duplication across multiple technologies. Following the Software-Defined Networking (SDN) paradigm, the centralized controller maintains a global network overview by receiving detailed monitoring information from each VMAC and can in return enforce instructions (i.e., propagate rule changes). Furthermore, we support a gradual network-wide roll-out through the transparency towards standard (i.e., legacy) devices and the possibility to interact with existing network controllers.

In contrast to our previous work, we do not focus on the intelligence or algorithms that can be deployed on top of management frameworks to perform the actual optimizations [17, 18]. Furthermore, we extend our work in the following ways: First, we split the framework in different building blocks that in detail discuss the overall workflow and operations, and the interactions with different actors. Second, we discuss in detail how the framework can operator with wireless technologies, in particular LTE. Third, we provide a comparison with MPTCP, the standard industry solution, in a real-life prototype.

The contributions of this paper are threefold. First, we propose the VMAC design, the ORCHESTRA SDN meta-controller principles, and the underlying building blocks. Second, we introduce the packet-level handover, load balancing, and duplication features of ORCHESTRA. Third, we show the integration of IEEE 802.11 technologies with LTE in a real-life prototype. Fourth, we provide a comparison between the proposed ORCHESTRA solution and MPTCP Moreover, the remainder of this paper is structured as follows. We start by presenting related work in Section 2. Section 3 gives an overview of the ORCHESTRA framework, including the VMAC and centralized controller. Section 4 describes how the framework can be used with different underlying communication technologies, such as Wi-Fi and LTE. In, respectively, Sections 5 and 6, we present a number of relevant use cases and the detailed implementation of our prototype. We follow up with the results and discussion in Section 7 and, finally, conclude in Section 8.

2 Related work

2.1 IEEE 1905.1 standard

With the rise in the availability and popularity of both wired and wireless communication technologies, interest grew to concurrently use multiple technologies. This was particularly the case in home environments. In 2013, these efforts led to the definition of the IEEE 1905.1 standard [13, 19]. This standard introduces a novel architecture where an abstract MAC layer was positioned on top of the existing data link layer (i.e., OSI layer 2). As such, it was possible to transparently combine all the different MAC interfaces into one [20]. Each device connected to the network is represented by a unique virtual MAC address. Moreover, the standard allows to transparently hand over traffic stream and to perform load balancing across the different available interfaces by means of packet header matching rules. This functionality introduces a flow-level control over the network. The IEEE 1905.1 standard was designed with Local Area Networks (LANs) in mind and supports the following communication technologies: Ethernet, Wi-Fi, (Power-line) HomePlug, and Multimedia over Coax (MoCA). Regardless of the potential of IEEE 1905, the standard lacks industry adoption and only a limited number of products exist that support it (e.g., Qualcomm Hy-Fi). No follow-up releases or developments have been proposed since its release in 2013.

2.2 Software-Defined Networking-based approaches

The popular SDN principle can be an alternative for using a hybrid MAC layer. This is especially true as the SDN principle is also being proposed in LANs and wireless networks [21, 22, 23]. One of the first wireless SDN controllers, called ODIN, aims to increase the Wi-Fi experience of users. Furthermore, it also attempts to make dense wireless networks more manageable, while introducing QoS for a wide range of applications and use cases. The essential components of the ODIN architecture are the ODIN master and the ODIN agent. The ODIN master acts as the centralized network controller, while the ODIN agent is installed, using OpenWRT, on the different access points (APs) in the network. The key innovation of the framework is the introduction of the Light Virtual AP (LVAP) abstraction, as an addition to the default virtualization of APs (i.e., Virtual APs (VAPs)), to enable seamless mobility of stations. This is achieved by the virtualization of the association states and by separating these states from the physical APs. As such, when a station moves away, only the corresponding LVAPs is transferred to other physical APs, and the station remains associated [22, 24].

Building further on top of the innovations of the ODIN framework, the 5G-EmPOWER network framework is a more recent solution for the orchestration of wireless networks [25]. It extends the network programmability, compared to the ODIN solution, by offering both a REST API and a series of Python inter-

faces [22, 25]. Additionally, also more Virtualized Network Functions (VNFs) are provided. These novel features allow for more control and insight in the available resources in the network, such as the free bandwidth across the network or the load distribution across the infrastructure. However, the key novelty of the 5G-EmPOWER framework is the support for cellular networks, in addition to the offered control over the Wi-Fi networks [26]. The most notable offered control aspects are: resource allocation, network monitoring, wireless clients state management, and network reconfiguration [22, 25].

2.3 3GPP and Tunneling approaches

Related research can also be found in the area of 4G/5G networks, as in order to cope with the ever-growing bandwidth and traffic speed demands, especially towards the highly hyped 5G networks, the 3GPP community began exploring the wireless spectrum outside of the traditional licensed bands. Two different approaches have been proposed to offload traffic from the cellular networks: first, the use of unlicensed spectrum (i.e., LTE-Unlicensed (LTE-U) and LTE License Assisted Access (LTE-LAA)) and, second, the addition and use of Wi-Fi technologies (i.e., LWA) [14, 27, 28]. In the first case, LTE is directly used in the unlicensed spectrum (specifically the 5 GHz band). However, this can potentially cause severe performance degradations in coexisting Wi-Fi systems [29, 30]. Furthermore, note that different LTE-U deployments of different operators can also interfere with each other [31]. In contrast to LTE-U, LTE-LAA contains a Listen-Before-Talk (LBT) protocol and employs a so-called freeze period, where LTE leaves free airtime for other technologies. This allows LTE-LAA to be used on a larger scale and provides better coexistence with, for instance, Wi-Fi technologies. It has been shown that the throughput per Wi-Fi AP, under coexistence with LTE-LAA, is comparable to cases where the AP shares its spectrum with other IEEE 802.11 devices.

On the other hand, LTE-LWA provides the seamless usage of both LTE and Wi-Fi networks by combining an LTE Evolved Node B (eNB) with one or more Wi-Fi APs [28, 32]. This can be done through a physical integration or by using a (or multiple) external network interface(s) and can substantially increase performance. Similar to the previously listed 3GPP solutions, LTE-LWA introduces additional utilization of the 5 GHz band. However, no hardware changes are required on the infrastructure level, while also fewer coexistence issues are introduced [30]. The seamless usage of both communication technologies is achieved through the tunneling of mobile traffic flows over the Wi-Fi connection. Additionally, also handovers between both technologies are made possible. Research in the area of LWA focuses mainly on achieving high performance and low latency handovers. This is, for instance, achieved by decreasing the overhead of handovers and scheduling them properly, which leads to a reduced handover duration [33, 34]. Currently, only two LWA deployments are planned worldwide (in Singapore and Taiwan), while

already over 30 trials and deployments (both planned or launched) exist for LTE-U and LTE-LAA [35].

Furthermore, note that some other commercially available products exist that use a similar tunneling approach to hide away the underlying communication technologies. These products typically target LANs, and in particular office environments. A tunnel is configured between a so-called pro-active router or modem and an instance in the cloud, while the different technologies under the hood (e.g., Digital Subscriber Line (DSL), fiber, satellite or LTE) are concealed. The router decides which underlying technology to use per traffic flow, based on QoS parameters. This technique is also known as Software-Defined Wide Area Networks (SD-WAN) bonding. It is, among others, offered by the companies Mushroom Networks and Peplink [36, 37].

2.4 Multipath Transmission Control Protocol

Arguably the most used solution, especially by industry, is MPTCP. This extension of regular TCP originates from 2013 and allows the concurrent usage of multiple network interfaces for the transmission and reception of data [10, 38]. As such, it enables increased resource usage and redundancy in networks that contain multiple technologies. At the transport layer, the MPTCP enables the use of parallel regular TCP connections (called subflows) that each uses a different interface and can follow different routes through the network. However, MPTCP is transparent to the application layer as the different subflows are delivered as one to the higher layers.

The MPTCP scheduler is the most critical component as it decided how incoming data streams (i.e., TCP segments) should be handled, ideally taking into account the ever-changing network characteristics (e.g., increased Round Trip Time (RTT)) [39, 40]. Different policies can be executed by the scheduler: first, it is possible to divide or duplicate data streams across the available subflows. As such, respectively, higher throughput or increased reliability is achieved. Second, a back-up policy can be applied where one (or multiple) subflow(s) are kept idle and will only be used when the main subflow would break. Additionally, a back-up stream can be used as a channel to send retransmissions. While the fallback subflow is already in place and a handover can be executed very quickly and transparently, research has shown that these handovers are not seamless [41, 42]. Moreover, depending on the exact circumstances, it can take up to 2-3 s before the switch between subflows is completed. The Lowest Round Trip Time First (LowRTT) scheduler, that sends a series of segments over the subflow with the lowest RTT until its congestion window is filled, is the most popular one [39]. While MPTCP operates on a packet-level (i.e., subflow), the scheduling is only performed between two hosts and no coordination exists across the entire network. Another disadvantage is that it has been proven that MPTCP is very aggressive towards other non-controlled TCP connections in the network [43]. This behavior does not lead to guaranteed advantages for the MPTCP users.

Features	IEEE 1905.1	SDN-based	LTE-LWA	MPTCP
Network domains	LAN	LAN	LAN-Radio Access Network (RAN)	Any (end-to-end)
Technologies	Ethernet, HomePlug, Wi-Fi, MoCA	Wi-Fi, 3GPP	Wi-Fi, LTE	All
Coordination	Global	Global	Local (within cell)	Between end-points
Control-level	Flow-based	Flow-based	Flow-based	Packet-based (sub-flows)
Transport protocols	Any	Any	Any	Only TCP
Backward compatibility	No	No	Yes	Yes
Vertical Handovers	Yes	Yes	Yes (within cell)	Yes (between sub-flows)
Needs client changes	Yes	No	Yes	Yes
Products available	Qualcomm Hy-fi	Odin, 5G EmPOWER, ...	Two planned deployments	Android, iOS, Tessares, ...

Table 1: Comparison of existing multi-technology control and management solutions.

Originally, MPTCP was designed with multi-homed devices such as smartphones (enabled with both Wi-Fi and mobile interfaces) or servers (that are set up with multiple Ethernet interfaces) in mind [10]. Currently, indeed, MPTCP is actively being used on a large scale in Android and iOS devices (e.g., by Siri) [44]. However, multiple telecommunication operators also employ MPTCP to split traffic across both wired and wireless backbone networks (called hybrid access networks). This is, in particular, the case for DSL and LTE solutions, to circumvent the limited capacity of DSL wires (also known as DSL-LTE bonding or hybrid-DSL). Such a solution is, among others, offered by the company Tessares [45].

2.5 Comparison and summary

Table 1 positions the different discussed multi-technology management solution next to each other and compares different features such as network domains, supported technologies, level of control, and supported transport protocols. In general, we can say that nearly all listed approaches are technology dependent and/or target a specific network domain or use case. LANs are targeted by nearly all discussed technologies, except for the cellular ones. MPTCP and SDN solutions were originally designed for wired networks (e.g., in data centers, or in the core network) but are now also being applied to home and office networks. In terms of communication technologies, all listed solutions support multiple communication technologies, with Wi-Fi being the most

popular one. Furthermore, the above listed approaches operate, in general, on a flow-level, being able to reroute or hand over traffic flows across different network paths or connections. The major exception to all of this is MPTCP. MPTCP has already been applied in multiple domains and use cases, while being fully independent of the underlying communication technologies. It is also the most fine-grained solution that exists, as it allows to set up multiple sub-flows that can be used to transport the individual packets of a single traffic flow. However, MPTCP comes with two major drawbacks as it only supports TCP traffic and coordination is only possible between two endpoints and not network-wide.

This important network-wide coordination is currently only offered by the IEEE 1905.1 standard and SDN approaches (e.g., ODIN or 5G-EmPOWER) as they introduce a centralized controller to the network. Other approaches, such as LTE-LWA, have a more local and distributed form of coordination by exchanging messages between the different involved devices. Furthermore, although all listed approaches introduce intra- and inter-technology handovers, the seamlessness and performance of these operations can differ significantly. Typically, no guarantees can be provided on the duration of the handover. Finally, MPTCP is the only approach that, in addition to the handovers, also introduces features like duplication or packet-based load balancing. It should also be noted that the IEEE 1905.1 standard have never really been adopted by industry, in stark contrast to, for instance, MPTCP.

Summarized, it is clear that the existing approaches fail to address the multi-technology problem in a fundamental manner, without targeting specific communication technologies or application domains. Furthermore, in order to boost network-wide performance there is a need for centralized coordination, accompanied by more fine-grained control, to also support packet-level operations and not only flow-level operations. Note that at the end of this manuscript, we revisit Table 1 and provide a comparison of the proposed ORCHESTRA framework with the existing solutions discussed above (cf. Table 2).

3 Framework description

The goal of the proposed framework is to offer a single solution to manage all different technologies within a network, regardless of the technologies and the type and scope of the network. The ORCHESTRA framework consists of two main parts: first, the transparent VMAC layer that manages physical interfaces on a device without modifying the underlying layers. It can be deployed on any device, both end-user or being part of the network (e.g., an edge or core node), and introduces seamless interactions between the different technologies. Second, following the SDN principle, we introduce the ORCHESTRA controller that has a global view of the network. The main responsibility of the controller is to manage the different VMACs across the network, based on real-time monitoring information. Both components are extensively discussed

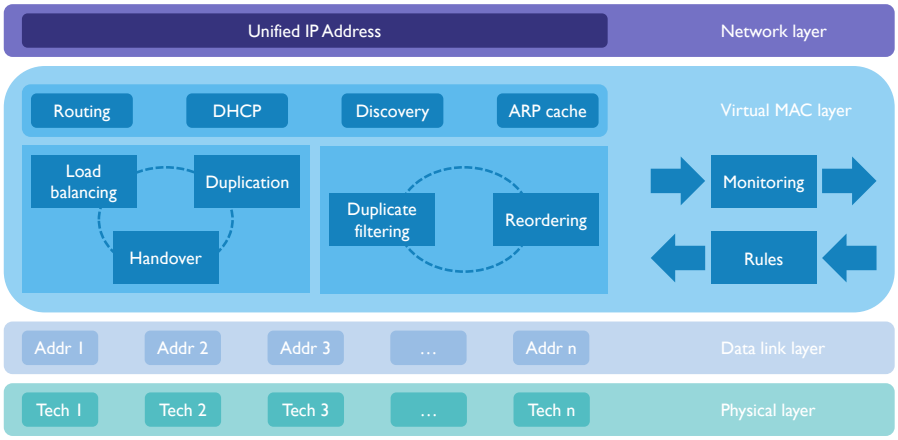

Fig. 1: Overview of the VMAC layer with its position in the OSI model, its buildings blocks, and its offered functionality.

in the next subsections, where we highlight, among others, the different building blocks, features, and interactions with legacy (i.e., non-ORCHESTRA) devices.

3.1 Virtual MAC layer

In order to provide continuous and reliable connectivity, a key feature is the enabling of inter-technology handovers and roaming. As identified in Section 1 the current structure of the OSI network model obstructs this behavior. All communication technologies or standards operate completely independent from each other, as they each define their own lower layers of the network stack (in particular the MAC and physical layers). This means that connectivity is currently handled on an interface basis as each interface has its uniquely assigned network address and applications tend to bind on a single, specific interface. Consequently, changing between interfaces results in connection loss. We solve this issue by introducing a virtual MAC layer (VMAC) and abstracting connectivity from the user and applications. It also enables the implementation of functionality (e.g., load balancing) that works across multiple technologies. The general architecture and capabilities of the VMAC are shown in Figure 1.

The novel layer is placed above the existing data link layer and below the network layer, appearing transparent to both of them. The main responsibility of the VMAC is to forward incoming packets from the network layer to one (or multiple) of the underlying interfaces (i.e., technologies) under its control, or vice versa, forward packets received from the data link layer to the above network layer. Existing layers are thus not modified, do not require knowledge of the presence of the VMAC, and packets are still regularly passing through

them. As such, abstraction and encapsulation, key principles of the OSI stack and the Internet, are maintained.

One of the main advantages is that there is only one interface (i.e., the VMAC) visible to the network and upper layers, while the underlying technologies, and their respective layers, are hidden away. This also means that also only a single Internet Protocol (IP) address per device is needed, without requiring any additional overhead. In contrast to the IEEE 1905.1 standard, no unique virtual MAC address is required. As such, the interaction with existing standards and protocols does not need to be altered. Furthermore, as the VMAC is capable of managing all the different interfaces, and these interfaces can be connected to different networks, it also needs to be able to route packets between different networks. For instance, between an edge network, that might be wireless, and a core network, that might be wired. Therefore, the VMAC incorporates the required bridging functionality as well.

Because of the single interface to the upper layers, and the abstraction of and control over all the underlying technologies, the VMAC captures all traffic and can therefore seamlessly handover between technologies. In particular, the VMAC introduces the following advanced functionalities:

1. Seamless handovers within a single technology between wireless endpoints or between different technologies to support mobility and leverage the best QoS available for a node across different technologies.
2. Packet-based load-balancing (and reordering at the other end) between two or more technologies to maximize network utilization.
3. Duplicating (and deduplicating at the other end) individual packets across several technologies to support high reliability.

The features stated above are enabled by the introduction of packet matching rules, to which incoming packets are matched. Additionally, statistics are gathered and forwarded to the central controller, while in turn, rules and commands are (ideally) received from the controller. This interaction is in more detail discussed in Section 3.2.1. Based on these rules (e.g., send all traffic to a specific node over a single interface) and commands (e.g., to perform a handover), the VMAC decides which interface handles the received packet and let the lower MAC layer take care of the actual transmission. With this granularity, the virtual layer can support packet-level control instead of flow-based handling, which allows for more versatility and control. This is in strong contrast to existing solutions such as IEEE 1905.1 and LTE-LWA.

Considering the general packet-flow, the VMAC introduces only minimal differences: on a sending node, when a packet arrives from the upper network layer, it enters the VMAC instead of directly going to the MAC of one of the underlying interfaces. Depending on the rule, that matches the header information of the packet, the VMAC decides to hand the packet over to the correct underlying interface, or in the case of duplication, multiple interfaces. On the receiving side, the incoming traffic is pushed from the data link layer interfaces to the VMAC, instead of directly being passed to the upper layers. In the specific cases of duplication and load balancing, some additional processing

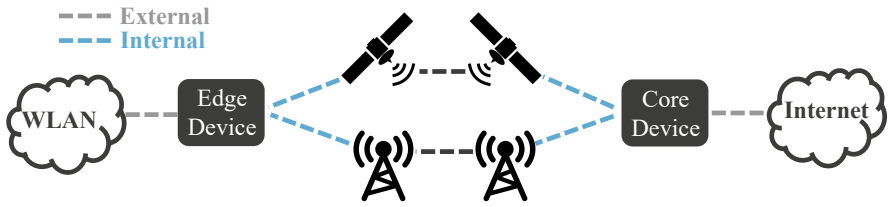

Fig. 2: Illustration of internal and external interfaces in a backhaul scenario.

has to be done on the VMAC, before passing the packet upwards. We discuss these intermediate steps in Section 3.1.2.

The VMAC can be installed on any device, both consumer (i.e., endpoint) or infrastructure side, and applied in all kinds of networks (e.g., LANs, RANs, backhaul networks). However, we mentioned earlier that the VMAC requires bridging functionalities in the sense that it is capable of forwarding packets from, for instance, a wireless backhaul or edge network to a (wired) core network and vice versa. Note that this bridging functionality is typically required in ISP or backhauling use cases. In this case, we need to make a distinction between the interfaces, controlled by the VMAC, according to their functionality. We discriminate two types of interfaces: internal and external interfaces. The internal interfaces are part of the wireless backhaul network, meaning that they handle the traffic to and from edge nodes. In many cases, this might appear as its own subnet without direct access to an outside network. On the other hand, there is at least one external interface, which is part of the core network or an external network. This interface has outside connectivity and needs to handle packets from a different subnet and translate the IP addresses to the internal interface and vice versa. The VMAC is also responsible for handling the routing between the different subnets to ensure connectivity. The difference between internal and external interfaces is illustrated in Figure 2. Note that this functionality is not required on all nodes, for instance, not for endpoints at the edge of a network. A modified implementation (more lightweight) can be provided for such devices, in particular in the context of resource-constrained devices.

Subsequently, we first describe the basic building blocks that are necessary for the VMAC architecture. Afterwards, we explain in detail the features that the virtual layer has to offer.

3.1.1 Building blocks

Unified IP: In order to have a stable connection on the transport layer, it is vital that the IP addresses of the endpoints do not change. For this purpose, the VMAC only uses a single IP address for all interfaces. This IP address is (arbitrary) requested by the VMAC through one of the interfaces under its control. In the case of a handover, the IP address remains the same, while only

the physical interface changes. As a consequence, the VMAC has to take care of Dynamic Host Configuration Protocol (DHCP) and relieve higher layers of it, otherwise the operating system gets into conflict with the network configuration as the same IP cannot be present on multiple interfaces. Furthermore, this also gives the controller an important role in actually managing all the different technologies and the VMAC in informing the controller correctly about what technologies it controls.

Dealing with multiple interfaces: When there are multiple interfaces active at the same time, for example when load balancing a traffic flow across multiple technologies, the normal Address Resolution Protocol (ARP) table of the operating system is not sufficient anymore. An operating system only matches an IP address to one of its interfaces, but if multiple interfaces are active (under the same IP address), the operating system would continuously overwrite the entry. Therefore, to cope with multiple simultaneously active interfaces and legacy devices, the VMAC needs to take care of the ARP handling. As such, keeping track of which IP address is reachable over which interface and if (potentially) an IP address is reachable over multiple interfaces. For this purpose, the VMAC maintains its own ARP cache and, upon receiving an ARP, stores the MAC address of the interface on which it received the ARP reply. The VMAC signals on all interfaces that it is available and remembers which IP to MAC tuple is available on which interface. To cope with the fact that an IP address can be reachable over more than one interface, a separate ARP cache is maintained per interface. This allows having multiple active connections at the same time, without experiencing problems regarding routing and discovering the other endpoint. When the VMAC receives an ARP request on a specific interface, the VMAC issues the transmission of an ARP reply only on that particular interface. This way a VMAC-enabled device can still communicate with legacy devices, both on the client or infrastructure side. Note that in an ideal case where only devices are present that are equipped with a VMAC, this functionality is deprecated, and the controller will take care of settings the rules.

Monitoring: There is a continuous stream of configuration and monitoring information from the VMAC to the controller, allowing the controller to have a detailed and global view over the network. The configuration information includes which interfaces are available on the device and their properties and capabilities. An example of such an interface might be an LTE connection with a bit rate of 150 Mbps. Furthermore, also the state of a specific interface, if it is up or not, can be shared. In addition, the monitoring information includes statistics about these interfaces and the traffic going through them. This includes, for instance, the received packets or bytes per second, QoS information, recorded signal strength values for wireless links, and latency information. This monitoring information is sent to the controller using simple User Datagram Protocol (UDP) packets.

Rules: The behavior of the VMAC is defined in the form of rules, typically set by the controller. On one hand, there are configuration instructions that specify the frequency of the transmission of monitoring reports to the controller. On the other hand, there are the rules that define how incoming packets (from both data link and network layers) should be handled. The latter includes the use of advanced functionalities such as load balancing or duplication. These rules consist of two parts and can be, conceptually, compared to OpenFlow rules. The first part states which packets should match the rule. This can, for instance, be done using source and destination IP addresses, port numbers, transport protocol types, and/or sequence numbers. The second part of the rule defines how the matching packets should be processed. This includes, for instance, simple forwarding over a single interface, load-balancing, or duplicating over multiple interfaces. For example, it is possible to directly forward all packets arriving within a specific IP range to one interface, while we balance a video stream across two or more interfaces to increase its throughput.

Furthermore, rules are sent to the VMAC by UDP packets. While the ORCHESTRA controller can handle this by a standard (UDP) socket, the VMAC checks the packet headers of incoming packets from the data link layer for the IP address of the controller and then extracts the required data from these packets. Note that the VMAC can also work without a controller present in the network and decide on its own transmission rules if necessary. It is also possible that the VMAC (or a local application running on top) decides to update the packet matching rules itself, for instance, in case of a disruptive network change (e.g., an interface going down). This is in order to minimize the impact on network traffic. Afterwards, the controller can, if needed, update these rules again, based on the received monitoring information and the global view, to have an optimal configuration across the entire network.

Discovery: When a device equipped with a VMAC joins a certain network, it needs to discover available controllers. Therefore, it broadcasts a (UDP) discovery message, to which the controller (or the most suited in case of multiple distributed controllers) responds. While the VMAC is not associated with a controller, it does not yet know the IP address of the controller. Consequently, the above-described procedure of receiving the controller's instructions based on the IP address is not yet possible. This can be solved by using a unique identifier in the UDP discovery message. Afterwards, the VMAC parses every incoming UDP packet, until a packet is received that is marked with the identifier at the beginning of the payload (first 64 bytes), and the IP address of the controller is learned.

3.1.2 Features

Handovers: A handover is an act of moving from one connection endpoint, like an AP or base station, to another connection endpoint. This can be done both within a single technology (referred to as intra-technology or vertical handover) or across different technologies (referred to as inter-technology or

horizontal handover). Furthermore, different devices can perform a handover, for instance, a client device in a LAN or cellular network and an edge node in a (wireless) backhaul network. The decision for a handover is typically made centrally by the controller and it informs the respective VMAC about it. In the case of an inter-technology handover, only a single interface is currently used for the transmission of data and this interface is considered to be the active interface. The goal of the handover is to transfer the status of the active interface to another interface. Consequently, the packet matching rules are updated to reflect the change and to, instead, use the new active interface. In the case of an intra-technology handover, the active interface stays the same, but its endpoint is changing.

The following procedure is executed: first, the VMAC buffers outgoing packets on the active interface (for a brief moment of time), before either switching the endpoint of the active interface or changing the active interface to the new one. In case the active interface is changed, it sends out a gratuitous ARP to announce that the IP address is now associated with the MAC of the new interface and all relevant devices can update their routing tables. If the interface is not connected yet, the VMAC takes care of connecting it (e.g., performing an association procedure) and then switching to it. As a fail-safe, if it is not possible to set up a connection, the VMAC switches back to the old connection. When the handover is successful, the VMAC releases the buffer and starts transmitting again. Additionally, without instructed by the controller, in case of a link failure, the VMAC can decide to switch interfaces or connections by itself to maintain a connection on a certain technology.

To efficiently hand over a device, make sure all traffic streams are correctly delivered, and minimize the overhead, a protocol and synchronization steps need to be in place. This is done by exchanging several messages between all involved parties (e.g., the client device, the APs with the corresponding technologies, and, if necessary, switches) and agreeing on a time to perform the actual handover between technologies. In the described process we assume that a handover takes place between a client and the two different APs, but the procedure is identical for a handover between any other types of devices. As mentioned above, the process starts when the controller informs the different VMACs that a handover is imminent. All VMACs acknowledge this and one of them (typically the station or endpoint) starts a synchronization timer, which is also communicated to the other devices involved, as such agreeing on the current time. Next, each of the three devices announces the time window δ, needed to perform the actual handover. First, all parties agree (by exchanging acknowledgements (ACKs)) on the largest time δ among all actors. While, afterwards, in a similar fashion they also agree on a time t to initiate the handover. Finally, the handover is executed and afterwards the connection is tested. In case, the handover fails, both nodes fall back to the previous configurations. During the time δ of the handover, the VMACs on both the client and the new AP buffer the packets and transmit them after the handover has been completed and acknowledged.

In case the AP, in the previously described setup, is not equipped with a VMAC, a handover is still possible if the AP is managed by a SDN/Network Function Virtualization (NFV) controller, as such offering a form of legacy support. Otherwise, if the AP is operating fully independently, a VMAC-enabled device can still perform a handover and buffer its packets to lower the overall packet loss. However, no guarantees can be given on the overall duration of the handover and the overall performance, as this completely depends on the configuration of the APs in question.

Load-balancing: In contrast to a handover, while performing load-balancing, multiple interfaces are active and the network traffic is distributed according to a certain scheme to each interface. Opposite to most existing approaches, the VMAC introduces load-balancing at a packet level. For this, a simple weighted Round Robin load-balancing is used where a fixed number of packets are assigned to a specific interface before moving on to another interface. This can be done without introducing overhead as the VMAC only needs to forward the packet to another interface that is active. However, if packets that are part of a continuous data flow are being sent across different interfaces, no guarantees can be made on the order of arrival at the other endpoint, especially in a wireless context (e.g., due to different latencies across different technologies or external interference). Therefore, at the other end, packets need to be reordered in the VMAC before being passed to the upper layer. Otherwise, transport layer protocols, especially TCP, will react in an unpredictable way. This reordering is done on a flow basis, per TCP session, and according to the transport layer protocol header, more specifically, the port number and the sequence number. Additionally, the source and destination IP addresses are used to assign different packets to a flow. The VMAC keeps track of the sequence numbers and buffers packets when a sequence number is missing. If the out-of-order packet arrives, it will be forwarded immediately, while the previously buffered packets are forwarded afterwards in the correct order, until a next out-of-order packet is identified. As some packets may never arrive at the receiving end, a timeout for missing packets is provided. This timeout depends on the rate of the traffic flow as this determines the turnaround of sequence numbers, used to identify out-of-order packets. Furthermore, the timeout is dynamically adjusted by monitoring the throughput and kept as low as possible to minimize negative effects on the transport layer protocol. When the timeout is reached, all packets that are available are forwarded in an ordered fashion to the network layer.

Duplication: Duplication is a useful method to achieve high reliability as it strongly increases the probability of a packet to arrive at the other endpoint. At the sending side, the VMAC enables this by copying an incoming packet and transmitting it across different interfaces (depending on the specified rule). However, at the receiving side, the VMAC cannot simply push the receiving packets to the network layers, as the same packet can potentially be forwarded multiple times. In turn, this can trigger unwanted behavior on the application

layer or a reaction of the transport protocol (especially in the case of TCP). As such, the receiving VMAC is responsible for filtering out these duplicates (i.e., performing deduplication). This is done in a similar manner to the reordering of packets upon performing the packet-level load-balancing, as described above. The VMAC maintains a hash map of packets that it already received. The packets are identified by source and destination IP address, transport layer protocol, IP identifier and IP fragmentation offset. A timeout is in place to prevent memory over usage. This timeout is, similar to the one used for missing packets, depending on the actual flow rate and therefore monitoring is necessary to adjust it appropriately. As the maximum number of duplicates that can arrive is known, the entry can be deleted as soon as this number is reached. Otherwise, the entry is deleted after the timeout. Note that none of the existing methods discussed in Section 2 allows for such a fine-grained form of duplication.

3.2 The controller

While the VMAC allows for fine-grained MAC-level control inside individual devices, the ORCHESTRA controller is the heart of the proposed framework and enables multi-technology management and orchestration across the entire network. Following the SDN principle, the controller takes control from the individual devices and their respective VMACs. The controller keeps track of all connected VMACs and issues instructions and updates to all of them to optimally configure the entire network. Furthermore, the controller is also capable of communicating with (other) SDN controllers, as well as individual infrastructure devices such as APs and switches. This is done by operating at a new and higher hierarchical layer, above of existing SDN controllers. An overview of the architecture can be seen in Figure 3. The architecture allows for more network control and a single central point to implement management logic. This contrasts with existing solutions like MPTCP and LTE-LWA. The communication with existing SDN controllers or infrastructure entities, allows for legacy support and an easier roll-out of the ORCHESTRA solution. This in opposition to, for instance, the IEEE 1905.1 standard, which requires more disruptive network changes and was never widely adopted. Additionally, the controller can also be distributed to increase scalability and reliability. In the next subsections, we elaborate more on the details of the controller, in particular on the communication aspect and the offered management possibilities.

3.2.1 Communication and interfacing

Here, we discuss all the interactions that are possible between the ORCHESTRA controller and the different entities in the network.

VMAC: The communication between the controller and the VMAC is lightweight and was already introduced before. In particular, the discovery of the

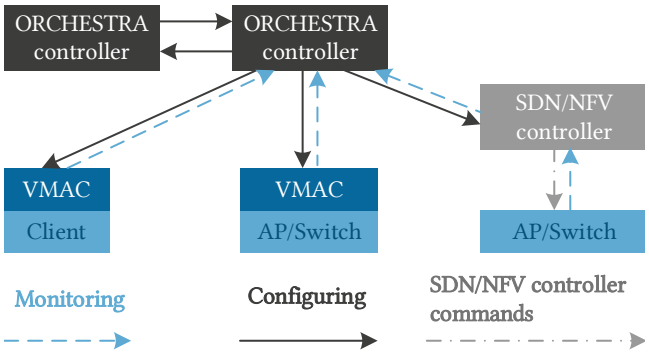

Fig. 3: The controller architecture and its communication.

VMAC and the two-way communication between the VMAC and the controller to, respectively, transmit monitoring information and rules, are discussed in Section 3.1.1.

Other SDN controllers: As it is unlikely that all devices within the managed network are immediately equipped with the novel VMAC, it is important to support legacy devices. In many networks, SDN controllers are already in place that offer certain management functions that can be exploited for devices not using the virtual layer. Examples of such frameworks, like ODIN and 5G-EmPOWER, are discussed in Section 2.2 [24, 25]. Interfacing with these SDN controllers requires more effort than the lightweight communication with the VMAC, as they usually do not support a built-in so-called northbound interface that is accessible through external communication. However, most controllers (e.g., the Ryu OpenFlow and the 5G-EmPOWER controller) offer application support insofar as you can write an application on top of the controller that interfaces with the controller and implements some higher-level functionality. This can be exploited by creating a northbound interface running, as such an application on top of the controller. The application handles the communication and translation of information from the SDN controller to the ORCHESTRA controller as well as commands from the ORCHESTRA controller to the SDN controller. The ORCHESTRA controller typically enforces station handovers (identified by MAC addresses) towards wireless SDN controllers (e.g., 5G-EmPOWER). While towards wired SDN controller (i.e., OpenFlow controller) the focus lays on traffic flow management and routing (e.g., adding flows, deleting flows, changing output ports). Vice versa, all SDN controllers provide the ORCHESTRA controller with the information that is available within the framework. For instance, traffic information (e.g., source and destination addresses, port numbers, or throughput), device information (e.g., MAC addresses or capabilities), and network conditions (e.g., link capacities or signal strengths). As such, we allow the ORCHESTRA controller to have an overview of, and to optimize, the various networks or segments

managed by different controllers. Note that the exact communication can be realized through different kinds of communication frameworks or protocols, for instance, using the lightweight and performant ZeroMQ framework [46].

Infrastructure devices: However, not all devices in the networks of today are managed by SDN controllers. While client devices cannot be managed at all without the presence of a VMAC or a SDN controller, infrastructure devices, such as APs, can in most cases be controlled through a variety of standardized protocols. This typically depends on the specific type of device. For instance, continuing the popularity of the SDN paradigm, OpenFlow is prevalent as the communication protocol towards switches. This means that switches can be controlled directly by utilizing the OpenFlow protocol and send flow-based rules. For APs this is less straightforward, but they often support configuration through the Network Configuration Protocol (NETCONF) with Yang as the modeling language. As such, a Yang model can be developed for every type of device, according to the exact capabilities of that device. Finally, in theory, it is also possible to extend the supported protocols, but this would require updates to the devices as well. If the option exists to update the (endpoint) devices, it might be better to move directly to either an SDN solution or the installation of the proposed VMAC.

Distributed ORCHESTRA controllers: To ensure scalability and reliability, the ORCHESTRA controller can be distributed. The communication between these different distributed ORCHESTRA controllers is handled in a similar fashion as the communication with other SDN controllers. The controller maintains an eastbound interface that includes the discovery of other controllers through broadcasting, as well as the transmission of heartbeats to maintain the connection. Only relevant information to other controllers is exchanged to reduce network traffic. This information includes common devices that are in the range of multiple controllers, especially if a device might be moved from one controller to another. Information is exchanged either by request or by informing another controller that one of the nodes is leaving the control of the current controller. For instance, consider a device that is in the range of two APs and is connected to one of them. One of the APs is in the region of the first controller, while the other AP is managed by a second controller. As both controllers have information on the device, the state is shared among both controllers. If a handover is needed, because of a newly computed assignment would place it in the region of another controller, the controller currently responsible for the device, informs the other controller to take over the device. The new controller, in turn, updates its flow rules and AP configuration and acknowledges the handover. Afterwards, the old controller deletes the remaining flow rules and the AP configuration and only further monitors the device. Note that this entire exchange happens fully transparent to the moved device and its VMAC. Finally, as the communication between different ORCHESTRA controllers is similar to the interactions with other SDN controllers, the same underlying communication frameworks and protocols can be used.

3.2.2 A global view in one location

The ORCHESTRA controller has two other components besides the communication interfaces. The first part consists of a data store where all received information is aggregated and combined into one state model, representing the whole network under consideration. This includes information about the VMACs (e.g., throughput, Received Signal Strength Indicator (RSSI), latency), about infrastructure devices (e.g., how many clients are connected, the capabilities, and performance), and about the SDN controllers (e.g., the local view of that controller). All of this information is stored in a single format in a large store or database that can be shared among the potential several controllers (e.g., through a distributed database).

Second, the controller also offers a northbound interface which applications, running on top of the controller, can use. This allows for implementing decision-making logic and algorithms on a single location in the network, managing different devices and network technologies in a ubiquitous manner. As network technologies are abstracted and the controller takes care of the abstraction layer, this greatly simplifies the implementation of such management logic applications.

In our previous works [17] and [18], we introduce such algorithms that optimize the network-wide throughput. Other algorithms can, for instance, also focus on Time-Division Multiple Access (TDMA)-based scheduling or even on energy efficiency. The intelligence schemes use the aggregated information of the storage as input to provide a certain configuration for the network. For instance, in the case of the algorithms presented in our previous work, a device to connection point to technology mapping is created, as well as routes for all traffic flows [18]. Based on this configuration, the necessary commands are issued to the corresponding devices across the network to actually roll-out the particular configuration.

4 Applicability to different wireless technologies

In this section, we discuss how underlying communication technologies can be used in conjunction with the ORCHESTRA framework. We focus mainly on IEEE 802 and 3GPP technologies, and highlight, in the case of LTE, potential challenges that can be encountered when integrating the technologies.

4.1 IEEE 802

The IEEE 802 standards define a physical layer and a MAC layer for different technologies, such as Ethernet (IEEE 802.2), Wi-Fi (IEEE 802.11) or wireless Personal Area Networks (PANs) such as ZigBee (IEEE 802.15). These two layers define the physical transmission over the medium and how the medium should be accessed. For instance, in the case of Wi-Fi the MAC defines a

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme. The IEEE 802 standards do not define any layer higher than the MAC layer, which means any network layer communication can be used. By default, this is IP, as it is the prevalent network protocol. Integrating IEEE 802 technologies into the VMAC is therefore straightforward and can be seen as plug and play. As the VMAC is positioned on top of the existing MAC layers of the technologies, it simply receives the incoming packets from the MAC layer and similarly, injects the outgoing packets into the appropriate MAC layer(s). No modifications to the underlying technologies are necessary, as intended.

4.2 LTE

Similarly, to the IEEE technologies, 3GPP technologies are defined by specifying a physical layer and a MAC layer. The transmission over the medium is defined by the physical layer, while the access to the medium is defined by the MAC layer, offering the possibility to use Frequency Division Duplex (FDD) or Time Division Duplex (TDD) modes. Contrary to IEEE 802 technologies, 3GPP technologies split the control and management plane from the data plane, comparable to the key principle of the SDN paradigm. For this reason, 3GPP specifies a set of entities for providing authentication and connectivity to the User Equipments (UEs). The following procedure is followed: when a UE tries to connect to a network, it first talks with the eNB, which notifies the Evolved Packet Core (EPC) to authenticate the user subscription. If a valid subscription is found, the eNB establishes a General Packet Radio Service (GPRS) Tunneling Protocol (GTP) tunnel to the gateway to grant the UE access to the network of the operator. If there is no valid subscription, the eNB cannot simply create a tunnel to the gateway and therefore, the client does not get connectivity with the network. A series of different interfaces are defined between the management entities and the access to external networks still utilizes GTP tunnels as a means of transportation.

As such, we can say that LTE by default carries legacy functionality in the form of these GTP tunnels. While GTP tunnels might have an advantage in managing clients in a traditional sense and in providing a secure channel across another technology than LTE (e.g., in LTE-LWA), it has the downside that all (data) traffic flows through the gateway, the endpoint of the tunnel. When considering use cases that provide services that are close to the edge of the network (i.e., close to the user device), this is a major disadvantage as you are producing additional traffic in the core network. Furthermore, in the scope of the proposed VMAC this also introduces limitations as, among others, packet-based load balancing and duplication becomes infeasible. This is due to the fact that the GTP tunnels do not allow to detect individual traffic flows, and it becomes infeasible to aggregate data flows that originate from another technology and network. As such, the standard LTE core architecture is not compatible with the VMAC. This can be addressed by including an additional

header in the packet with flow information, but this would create additional overhead. However, alternative solutions are available.

4.2.1 LTE-LWA

As introduced in Section 2.3, the 3GPP community introduced the cooperation of LTE and Wi-Fi technologies in order to offload traffic from the cellular networks [14, 27]. In particular, LTE-LWA was introduced in 3GPP Release 13 [28, 32]. LTE-LWA allows for both a co-located and a non-co-located deployment of the two technologies. In the first case, the Wi-Fi AP and LTE eNB are connected through an external interface, denoted as Xw. On the other hand, the physical integration of the AP in the eNB is also possible. In both cases, the aggregation of user plane data flows, transmitted over the two different technologies, occurs in the Packet Data Convergence Protocol (PDCP) layer. In turn, the LTE-WLAN Aggregation Adaptation Protocol (LWAAP) is responsible for encapsulating the data packets to tunnel them over the Wi-Fi connection.

This LWA architecture provides an aggregation point for the LTE and Wi-Fi technologies, before traffic flows disappear in the GTP tunnels. As the VMAC is intended to bind over different interfaces, offering a single upwards connection, the aforementioned architecture can also be used for the installation of the VMAC. The VMAC can replace (or be merged with) the PDCP layer, offering additional features like packet-level load balancing and duplication. Furthermore, the deployment of the VMAC architecture removes the need for the tunnel over the Wi-Fi connection, as the VMAC is fully IP-based. As this adapted architecture requires changes to the current standards, it counteracts our initial idea of transparency to upper and lower layers, while potentially limiting the adoptability of the presented approach. Note that also only a select number of LTE-LWA deployments is currently planned worldwide, as mentioned in Section 2.3

4.2.2 MEC architecture and Local Breakout

To allow for more flexibility and control over network resources, shorter routes, and the introduction of an IP interface, it was proposed to break open the above-mentioned GTP tunnels for data traffic [47]. This idea originates from the desire of telecommunication operators to have more insight in, and control over, the data traffic [48]. Furthermore, it is also proposed to enable edge computing, more efficient access to resources and services for clients (e.g., for gaming), and for 5G connectivity in Vehicular AdHoc Networks (VANETs) [49, 50, 51]. Within the context of edge computing, the Multi-access Edge Computing (MEC) architecture has been developed, as shown in Figure 4. The essential part is located in the base station, where user IP data packets can be intercepted by decapsulating GTP packets. Those IP packets are rerouted to the edge network (i.e., MEC server/gateway) by the introduction of a breakout rule that changes the path. This mechanism, known as Local

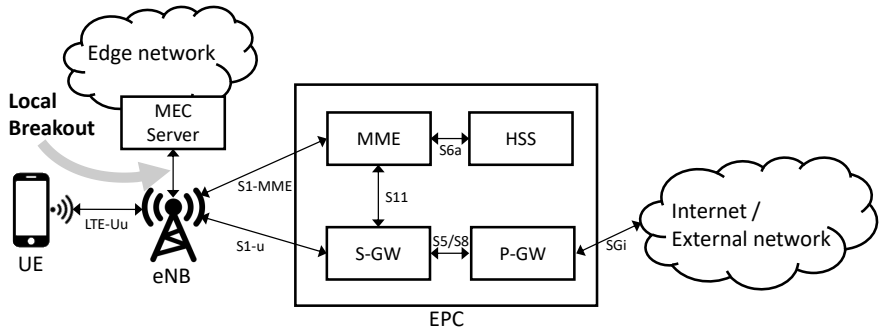

Fig. 4: Basic MEC architecture

Breakout (LBO), was standardized in 3GPP Release 15 [52, 53]. Note that it does not affect the management part of LTE that still uses the standard GTP tunnels. The main advantage of this architecture is that the local traffic can be offloaded from the RAN to reduce the end-to-end latency of edge services and save core network load. The MEC system was developed independently from the already existing LTE networks. However, it is currently being considered in the further development of the 5G technology, since edge computing has been marked as one of the key elements required to enable future Internet of Things (IoT) services [54].

The MEC architecture, and in particular the LBO, opens opportunities for the use of the VMAC layer in an 3GPP context. First of all, it is possible to integrate the presented VMAC layer with the MEC architecture by installment on the MEC server. This enables the use of the ORCHESTRA features, like seamless handovers or load balancing, over the LTE connection and the other present communication technologies (e.g., Wi-Fi) in the edge network. Furthermore, in a non-edge computing context, we can still use the LBO to intercept the IP packets from the connected UEs that are ORCHESTRA-enabled. These IP packets are then routed to whichever VMAC layer is installed at the infrastructure side. Ideally, the VMAC is positioned close to the edge of the network. As such, flows that are split across different routes can be merged as early as possible, limiting the differences in, for instance, latency and arrival time, for the split flows due to the different link conditions. The VMAC can, for instance, be installed on an additional device connected to the eNB, similar to the MEC server. From this device, the merged flows can be routed again to the core network (if required) in order to reach the Internet or external networks. The VMAC layer can also be installed in the core network itself or on intermediate nodes between the eNB and the EPC. This all depends on the network architecture of, for instance, the telecommunication operator. Essential is that the VMAC is positioned in such a manner that split flows going over different routers (i.e., technologies) can be routed to it. Note that using the LBO technique and routing the IP packets directly, removes the need and

overhead of the GTP tunnels (for data traffic). In Section 6, we present such a prototype implementation that utilizes the LBO mechanism to allow for an ORCHESTRA setup with Wi-Fi and LTE technologies. Furthermore, future work should further study the optimal placement of the VMAC layer in the RAN in more detail.

5 Use cases

In this section we discuss various use cases to demonstrate the versatility and applicability of the ORCHESTRA framework. For each use case we clarify the advantages for end users (e.g., better services) and the gains for the network operators (e.g., additional chargeable services or easier network management).

5.1 Enhanced satellite networking solutions

As still two-thirds of humankind has no access to wired or wireless Internet, interest has grown in satellite networks with global coverage capabilities [55, 56]. Furthermore, satellite technologies are also being used to provide Internet access (i.e., Wi-Fi) on board of ships. Initially, Geosynchronous (GEO) satellites were used to provide connectivity to a large area, at the cost of a very low data rate and high latency because of the long distance to the satellites. Therefore, a hierarchical spot-beam architecture has been proposed where a GEO satellite controls a group of Low Earth Orbit (LEO) satellites that each offer connectivity to a smaller area on the ground [56]. However, because of the use of the mobility of LEO satellites, a much more dynamic environment is created, which requires advanced SDN solutions to manage the frequent horizontal handovers between satellites [56]. This is where ORCHESTRA comes into the picture as it can manage the handovers in a more transparent way, thereby reducing the management burden for the satellite network operator. Because of the fact that recalibrating and positioning the satellite receiver to a new satellite (i.e., a handover) takes time, dual-receiver solutions have been proposed, where a second receiver is directed to another satellite, while the first one remains connected with the old satellite. In this case, ORCHESTRA can provide a smooth handover and manage both interfaces to the receivers.

As satellite networks, by nature, introduce a relatively large delay and connectivity issues can occur, the cooperation with other technologies brings clear advantages. For instance, a ship that travels near a coastline can be in range of land-based LTE networks, which often offer better QoS than a satellite link. The implementation of ORCHESTRA in the ship's receiver (i.e., the edge node) allows for the simultaneous use of both LTE and satellite networks. This results, among others, in a more stable and performant Wi-Fi network on board of the ship for the crew and passengers.

5.2 Enabling autonomous driving

The vehicles on our roads are becoming more intelligent and will, eventually, become fully autonomous. An essential aspect of this evolution is the communication between these vehicles and (road-side) infrastructure and between vehicles mutually. This communication is required to support features like platooning, provide updates on the condition of the road and traffic ahead, or even optimal lane usage. Currently, two main concurrent technologies have been developed: IEEE 802.11p (the base for the IEEE 1609 and European ITS-G5 standard) and LTE-Vehicular (LTE-V) [57, 58]. As both technologies will be deployed, for instance, alongside our roads and in our cars, load balancing can be used to off-load traffic and devices across the two technologies. This can help, among others, to keep latency low and allow for high-speed communication. Furthermore, the duplication of critical data can be used to offer more reliable communications. Finally, note that ORCHESTRA can also be considered for the in-car network, as these autonomous vehicles will also typically provide Internet connectivity in the car for there passengers.

5.3 Edge computing for large IoT deployments

Edge computing is the paradigm where intelligence and computational resources are (partially) moved away from the traditional cloud environment to the edge of network [59]. As such, it allows addressing concerns like response time, battery life constraints, bandwidth efficiency, and data safety or privacy [59]. Edge computing has been identified as one of the key enablers of the large-scale adoption of the IoT paradigm [54, 59]. For this reason, it is also a critical aspect of the 5G technology roadmap and research [48, 54]. At the edge, large numbers of interconnected devices (e.g., sensors, cameras, intelligent displays, end-user devices, ...) will be present, while different communication technologies will be used. In this heterogeneous environment, ORCHESTRA can aid by offering inter-technology network management to, among others, enable more efficient communication to reduce energy consumption and offload traffic streams to support large volumes of data and users. An interesting direction for future work is the application of ORCHESTRA in the MEC architecture, as discussed in Section 4.2.2.

5.4 Extended coverage in rural areas

While a majority of people live in hyper-connected cities, there is still a significant amount of people that live in more rural areas, for instance, in the southern part of Belgium. These houses often have an old DSL line, originally for telephone communications, that is used for Internet access. However, the limited capacity of these lines is not sufficient to meet the growing demands of end-users. As houses are sparsely distributed with large distances between them,

it is also too expensive for telecommunication companies to deploy high-speed broadband solutions. Therefore, recently, hybrid-DSL with LTE solutions have been proposed where the home gateway is capable of receiving both [45]. This is also known as DSL-LTE bonding. Traffic is divided among both links, thereby increasing the available capacity. Often, MPTCP is deployed at both endpoints to utilize both interfaces. However, each MPTCP connection needs to be created at one end and split again at the other end, which raises the management burden. Furthermore, the network operator needs to manage all the different MPTCP connections going to all end-users. The deployment of the ORCHESTRA framework heavily reduces the complexity of the management as it transparently handles all interfaces and traffic flows without the need for merging and splitting flows. Moreover, it also supports other traffic types than TCP. Note that we compare the performance of the ORCHESTRA solution to MPTCP in Section 7.

5.5 Wireless community networks

Because of the high availability, low-cost, and ease-of-deployment of wireless LAN equipment, wireless community networks have emerged [60]. In these wireless communities, broadband connectivity and a number of free services (e.g., free community-wide Voice over IP (VoIP)) is offered by a dense deployment of APs connected in a wireless mesh with fixed wireless access. These wireless community networks are traditionally connected to the Internet through a mobile network (e.g., 3G or4G) and/or one or more Wi-Fi point-to-point links, possibly over a long distance. Note that a nearly identical use case can also be found on large events (e.g., festivals), where a wireless mesh is deployed to provide connectivity for visitors or services, while a wireless backhaul network is installed. In both cases, the ORCHESTRA framework can be introduced to manage the wireless backhaul network. This enables features such as transparent handovers and load balancing between the different paths and technologies while reducing the deployment and management effort. Moreover, ORCHESTRA can also be used to manage the wireless community network itself.

6 Prototype Implementation

The current implementation of the prototype uses the Click modular router on a Ubuntu 16.04 machine [61]. We opted for Click as it allows for fast and high-level prototyping, which is handy for ongoing research. This is in contrast to a kernel-level implementation for a more finalized framework. While we use existing Click elements for basic packet handling, we implemented the VMAC logic in new elements to support the proposed functionality. The basic packet flow is shown in Figure 5. Multiple interfaces are connected to the *SuperFromDevice* and *SuperToDevice* block which take, respectively, care of

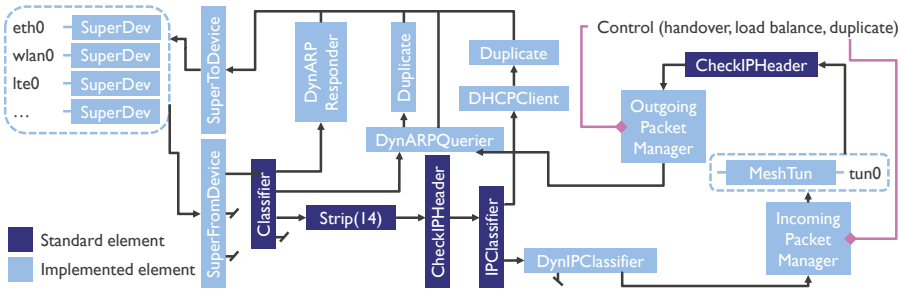

Fig. 5: The implementation graph for Click showing the different elements used.

forwarding packets from and to interfaces. Below we discuss the packet flows for both incoming and outgoing traffic.

6.1 Incoming traffic

For incoming packets, the header is stripped and the class of the packet is detected. This is done in the elements *Classifier*, *Strip (14)*, *CheckIPHeader*, *IPClassifier*, and *DynIPClassifier* in Figure 5 (denoted in grey). As the VMAC takes care of the generation of ARP requests and replies, it needs to filter out ARP at this point. A received ARP reply indicates that the virtual layer did send out a request because a packet in the buffer is waiting to be transmitted. If the arrived packet is an ARP request however, the virtual layer immediately replies (*DynARP responder*). Furthermore, as the VMAC does not provide a DHCP server or similar, it has to forward DHCP requests to the interface that is connected to the corresponding network. However, the VMAC has a DHCP client of its own that takes care of requesting IP addresses (*DHCPClient*). This is necessary as the VMAC uses only one IP address for all interfaces. Note that we do not implement in this prototype the different internal and external interfaces, as discussed in Section 3.1.

If the incoming packet is determined as data traffic, the next step is the *IncomingPacketsManager* which implements the logic of the proposed features (e.g., deduplication or reordering). As it is incoming traffic, packets need to be reordered or deduplicated if load balancing or duplication is used. The VMAC also checks for controller traffic at this point and consumes the packet if this is the case, in order to change its configuration or rules. Afterwards, the data packet is forwarded to the *Tun* interface and made available to higher layers.

In this prototype implementation, for the sake of easiness, the control packets (i.e., commands), indicated in purple, are directly sent over a socket to the *IncomingPacketsManager* where they are processed and the packet matching rules (e.g., for reordering or deduplication) are updated. In a real implementation, the control packets would follow the same route as the data traffic until they reach the *IncomingPacketsManager*.

6.2 Outgoing traffic

In the other way around, outgoing traffic is handled in a similar manner. After identifying the packet, there are two main components. First, the *OutgoingPacketsManager* which implements load balancing, duplication, and the handover logic. This component decides to which interface a packet is forwarded. Furthermore, similar as for the case of incoming traffic, the commands to update the packet matching rules are sent directly to the *OutgoingPacketsManager* in order to update the rules (e.g., update the weights for load balancing). Second, there is the *DynARPQuerier*. Here, ARP requests are generated and transmitted across multiple interfaces, while an outgoing packet is buffered if no ARP entry exists for the requested IP. As soon as a reply arrives or if the entry exists, the packet is forwarded to the underlying interface, which takes care of the actual transmission.

7 Evaluation and discussion

In this section, we compare the capabilities of the ORCHESTRA framework to MPTCP for the three described key functionalities. First, we start with a description of the evaluation setup. Next, we show the results for handovers between two interfaces, followed by load balancing across two interfaces, and ending with duplication across two interfaces. Each interface is using a different technology, namely Wi-Fi and LTE.

7.1 Experimental setup

The prototype setup consists of several components, displayed in Figure 6. The prototype represents the setup for the deployment of the VMAC on the devices in a wireless backhaul scenario. The core components that are equipped with the VMAC are the following: (i) the edge node, which is close to the end-user and consists of a device that acts as an LTE UE and a Wi-Fi client. (ii) the core node, which is connected to the wired core network and connected over Ethernet to an LTE eNB and a Wi-Fi AP. Additionally, there is an EPC that manages the LTE network (e.g., authentication), and an external DHCP server. Both are connected via a switch to the core node. Note that the setup would be the same for deployment in LAN, except for the fact that the edge node would be replaced by a client device, and the core node would be called an infrastructure device.

The AP consists of an APU2c4 board using the LEDE operating system with an IEEE 802.11n Wi-Fi card using a 20 MHz channel [62]. Furthermore, it is configured through OpenWrt as a bridge between the wireless and wired network. The base station is installed on a computer with an Intel core i7 8700k processor and 16 GB of RAM with a USRP B210 Software Defined Radio (SDR) using a 15 MHz channel. It uses a modified srsLTE implementation to

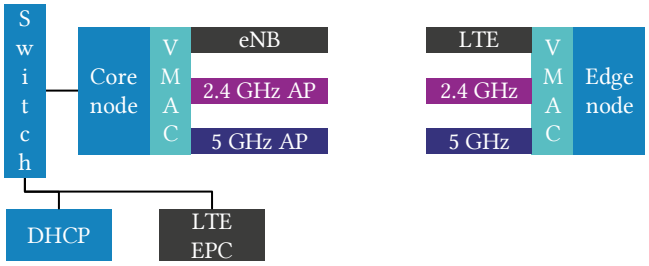

Fig. 6: The setup of the prototype including all devices.

create an eNB that allows to remove GTP tunnels. The eNB is managed by the OpenAirInterface EPC [63]. The edge device, the core device, and the EPC device are all Intel NUCs with a core i5 4250U processor and 16 GB of RAM. For the UE, a Huawei E3372 LTE USB stick is used, while the DHCP router is an arbitrary home router. Note that previous versions of the prototype also contained an Ethernet connection.

In the following scenarios, except stated otherwise, LTE and Wi-Fi (on the 5 GHz frequency band) are employed for the two interfaces. We evaluate our solution with both TCP and UDP streams, generated through *iperf*, while comparing it to MPTCP version 0.94 [64]. All tests are conducted in an office environment where there is a distance of 2 m between the edge node and both networks. Each scenario is repeated 10 times, and average results are reported.

During the different experiments, we demonstrate and evaluate the different features of the framework (i.e., handovers, packet-level load balancing, and duplication). In a full-blown deployment, intelligence (such as the network-wide load balancing algorithms presented in our previous work [18, 65]) would be installed on top of the ORCHESTRA controller. As such, optimizing the overall network performance and QoS by using the different features of ORCHESTRA. In our prototype, instead of using an algorithm, we pre-programmed a series of actions that should be sent to the different devices and their respective VMACs. An example, of such a series of commands, is to perform a handover between two technologies every 30 seconds or to dynamically change the weights when performing load balancing.

7.2 Seamless and transparent multi-technology handovers

In this scenario, we consider that a handover between Wi-Fi and LTE (or vice versa) is initialized every 30 seconds, which is indicated by the vertical lines in the figures. Furthermore, we consider a 1 Mbps flow of traffic for 120 seconds. Figures 7 and 8 show, respectively, the results in terms of throughput and latency. MPTCP can handle handovers with a 1 Mbps stream to a limited extent in terms of throughput (it loses connection, but can reestablish it), as can be seen in Figure 7. However, latency increases heavily as MPTCP loses

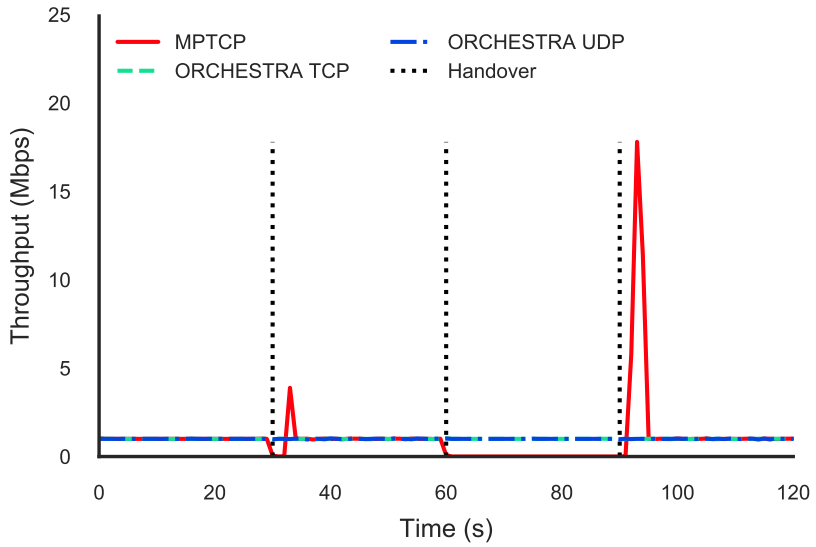

Fig. 7: Handover performance of MPTCP and ORCHESTRA in terms of throughput.

connection and first needs to establish a new sub-flow. In contrast, ORCHESTRA can seamlessly switch between technologies and maintains a constant throughput, while keeping the latency low (cf. Figure 8). In particular, we see that there is a downtime of 21 % for MPTCP, while this is 0 % for ORCHESTRA. The two handovers from Wi-Fi to LTE (at 30 s and 90 s) result in a connections loss of respectively 3 and 2 s. This corresponds to values reported in literature [41, 42]. However, switching from LTE to Wi-Fi at 60 s, results in a connection loss of 30 s, as the connection is lost until switching back to Wi-Fi. The reason for this significantly higher connection loss is unclear, and is potentially caused by some misconfiguration. Finally, note that the seamless connectivity provided by ORCHESTRA is the case for both TCP and UDP streams. This is also in contrast to MPTCP which, by its nature, only supports the TCP protocol.

In order to demonstrate the backward compatibility of ORCHESTRA, an additional small experiment was performed where we tested three variations in the configuration of an AP and station. Note that we do not use the wireless backhaul scenario here, as a difference in configuration is more likely to occur in a LAN scenario. In the first configuration both the AP and station are ORCHESTRA-enabled devices with a VMAC. The second configuration consists of an ORCHESTRA-enabled AP as before, while the station is a legacy device (without a VMAC). The third considered configuration contains two legacy devices. For the legacy devices, we used two Intel NUCs with Ubuntu 16.04 installed. As Ubuntu machines are used, we had to simulate the hand-

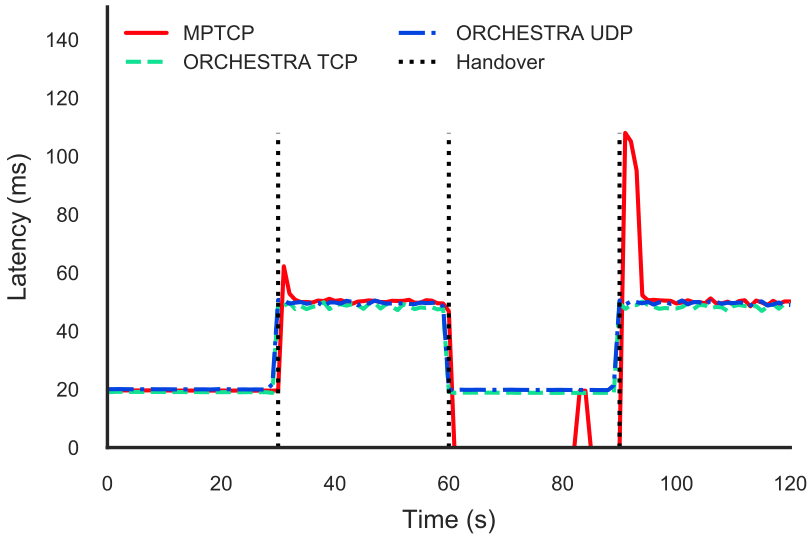

Fig. 8: Handover performance of MPTCP and ORCHESTRA in terms of latency.

over of devices without a VMAC layer. By default, Ubuntu reacts very slow, if at all, when a connection is breaking down and multiple technologies are available. For a wireless connection, this can easily take 15 s or more, in which case the connection completely drops. The value of 15 s was experimentally determined. Note that this value for more end-user oriented operating system, like Windows or macOS, will typically be lower. Therefore, to have *iperf* not break down, a script monitored the link continuously and if no traffic was detected for four seconds, it switched the route to the correct interface. This approach can, to some extent, be compared to band steering where an AP forces a station to another frequency, except for the monitoring script. We show throughput results for both a TCP and UDP traffic flow of 6 Mbps.

From Figures 9 and 10 it is clear that all three variations result in different throughput patterns and that connection drops occur under the presence of legacy devices. For the scenario with two legacy devices, we see that after performing a handover the traffic drops completely because the underlying connection was lost as there is no coordination between the two devices. On the other hand, *iperf* with TCP tries to overcompensate by increasing the amount of traffic until on average a throughput of 6 Mbps is reached, as soon as the connection is reestablished. This can clearly be seen in Figure 9, as the throughput heavily increases and reaches up to 23 Mbps for Wi-Fi and 8 Mbps for LTE. This means that the application has to handle the connection loss and as soon as it detects it, it needs to reestablish the connection, causing a significant downtime. Note that not all applications can cope with this behav-

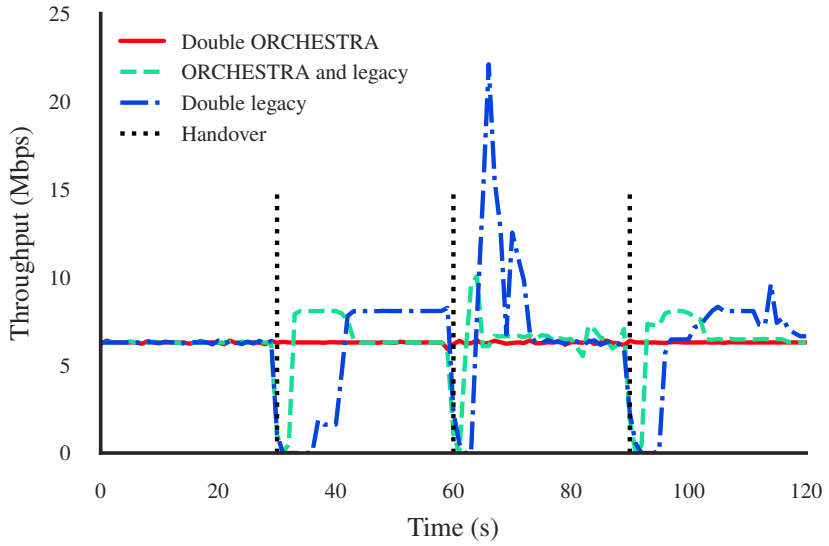

Fig. 9: Comparison of handover performance for ORCHESTRA and legacy devices for TCP traffic.

ior. UDP traffic exhibits a similar behavior, as shown in Figure 10. However, it is more resilient to sudden link failure as it does not require ACKs and packets are sent regardless if a connection exists or not.

In the second case, consisting of one VMAC-enabled and one legacy device, different behavior is experienced as the VMAC, can detect much faster than the operato if a connection is dropping. Upon a connection loss, it can easily switch to another technology and send packets over the new connection. This can be seen in Figures 9 and 10 as the drop in throughput is not as long as with the two legacy devices. While improving the downtime, the drop itself is not completely avoidable as the handover is done without informing the other device. This is still in stark contrast to the third scenario with two ORCHESTRA devices where drops are completely mitigated and seamless handovers are performed. Figure 9 shows that TCP itself is not reacting at all to the handovers and that the throughput remains constant throughout the run of the experiment. This heavily improves performance and the traffic flow (i.e., the underlying TCP protocol) does not need to overcompensate for the time the connection is down. The responsibility to take care of the network connectivity is removed from the application and is completely in control of the network intelligence or the network operator, who have a better overview of the network.

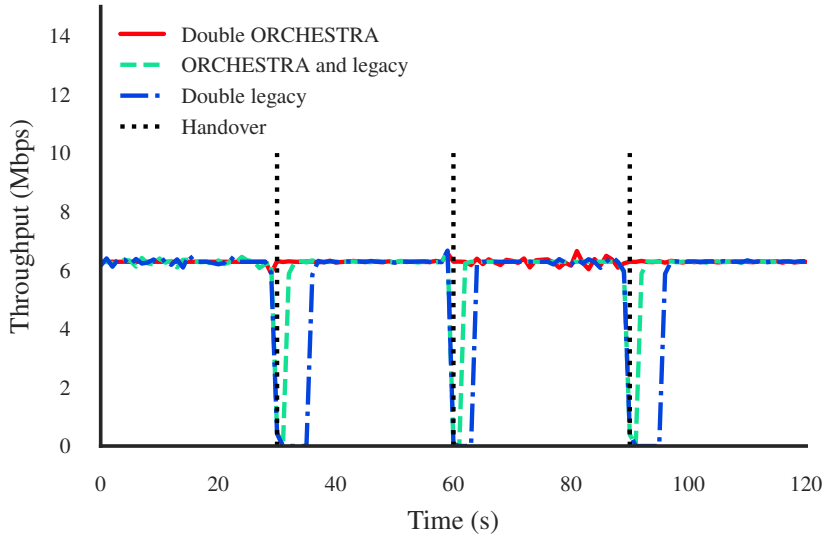

Fig. 10: Comparison of handover performance for ORCHESTRA and legacy devices for UDP traffic.

7.3 Fine-grained packet-level load balancing

In order to demonstrate the packet-level load balancing capabilities of ORCHESTRA using our prototype, the two interfaces are actively used at the same time. Both for a TCP and UDP stream we configure the VMACs on the two devices to balance the traffic evenly (i.e., according to a 50/50 distribution) across both the Wi-Fi and LTE interface. We compare this to MPTCP that is configured to use the default (round-robin) RTT scheduler. This scheduler sends a fixed number of packets over a specific interface, before rotating to the next interface. For this experiment, we again use a traffic flow of 6 Mbps.

The results in terms of throughput and latency are shown in, respectively, Figures 11 and 12. On average the desired rate of 6 Mbps is achieved by both MPTCP and ORCHESTRA. The latter does this for both TCP and UDP traffic flows. Figure 12 indicates that there is a significant increase in latency (of 40.6 %) for the TCP stream when using ORCHESTRA. We clearly see that latency builds up for the first 20 to 25 s before stabilizing. While, in contrast, the UDP flow and the TCP flow with MPTCP experience lower amounts of latency throughout the experiment.

The explanation for the behavior experienced with TCP when using the ORCHESTRA framework is twofold: First of all, the challenge in load balancing the individual packets of a traffic flow across two different wireless technologies lays in the different latency properties of those technologies. This

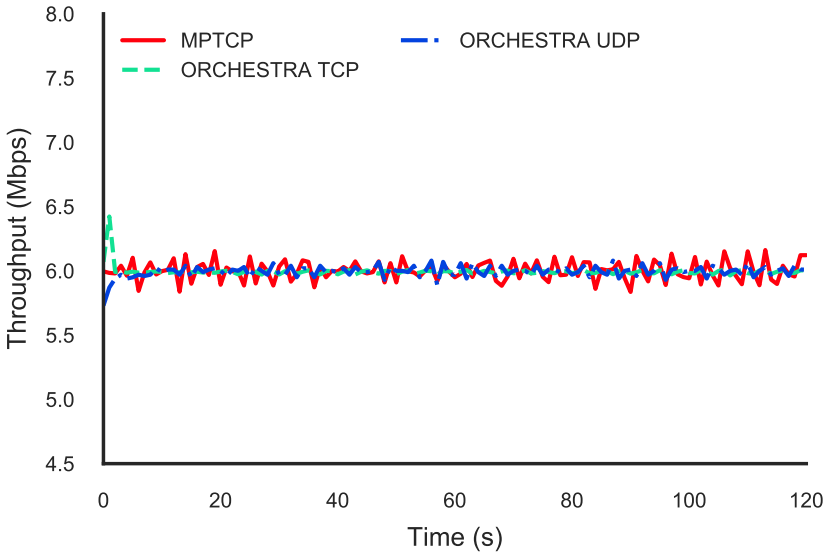

Fig. 11: Load balancing performance of MPTCP and ORCHESTRA in terms of throughput.

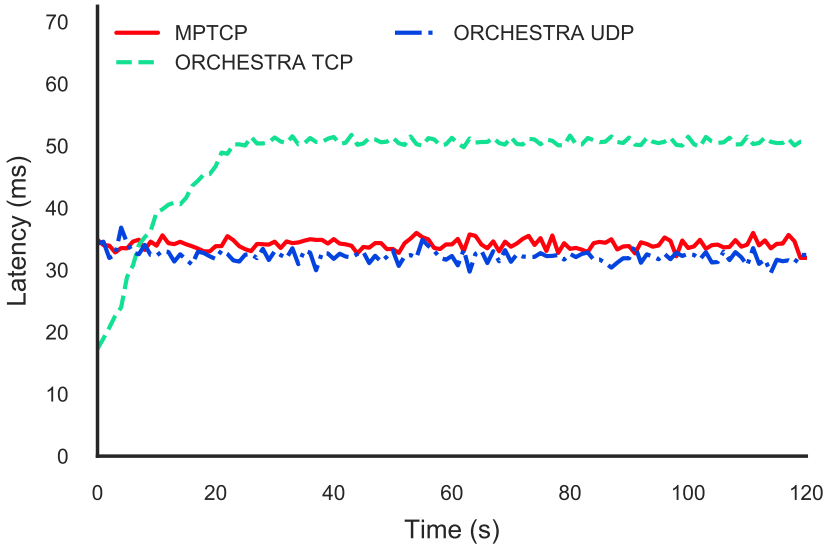

Fig. 12: Load balancing performance of MPTCP and ORCHESTRA in terms of latency.

potentially results in out-of-order packet arrivals. MPTCP circumvents this problem as its scheduler sends out a fixed number of packets after each other on the same interface before using the other one. This results in a 67 %/33 % distribution of packets across both interfaces in favor of Wi-Fi. As the Wi-Fi connection has lower latency, this partially explains the difference in latency with ORCHESTRA that really balances all packets evenly in a 50 %/50 % distribution. Furthermore, with the MPTCP protocol, large amounts of packets are sequentially sent on the same interface. This series of packets will thus always arrive in order at the receiver side (under circumstances with no packet-loss, as is the case here). However, note that the throughput of the TCP flow also slightly fluctuates, due to differences in latency when the scheduler switches between the interfaces.

Second, ORCHESTRA uses a reordering mechanism at the VMAC to cope with the reception of potentially out-of-order packet arrivals (as explained in Section 3.1.2). For TCP, this reordering is necessary because no assumptions can be made about the capabilities of the upper layer. In this case, out-of-order packets are placed in a hash map, until missing packets have been received (or a timeout is triggered). The packets are reordered according to TCP sequence numbers before being delivered to the upper layers. Since packets are distributed across both technologies according to a perfect 50/50 scheme, frequent reordering is needed, causing the increase of latency before stabilizing. The smaller fluctuations throughout the remainder of the experiment are caused by the TCP rate control mechanisms that react on the slightly varying inter-packet times. Note that the UDP flow does not experience this behavior, due to the lack of rate control algorithms.

In order to further demonstrate the functionality of the packet-based load balancing, and to investigate the impact of the reordering, the following experiment is conducted: similarly to before, we transmit both a 6 Mbps TCP and UDP flow that is split across two interfaces of the prototype, namely the 5 Ghz Wi-Fi interface, and the LTE connection. However, in contrast to the previous experiment we do not only balance the load evenly but vary the percentages: at the start the traffic stream of 6 Mbps is balanced 50/50 % across both available interfaces. After 30 s this is altered into 30 % of traffic over Wi-Fi and 70 % over LTE. At the 60 s mark we return to the initial 50/50 % configuration, before ending up with a 70/30 % for, respectively, Wi-Fi and LTE.

From the results for both TCP and UDP, shown in Figures 13 and 15, it is clear that the packet-based load balancing works as intended, as the traffic is in both cased distributed across both interfaces according to the set weights. Both the TCP and UDP flows achieve a stable throughput of 6 Mbps. The need for packet reordering is clearly shown in Figure 13 where the TCP flow only achieves a throughput of 3.2 Mbps when reordering is disabled. Furthermore, Figure 14 shows the observed latency across the entire length of the experiment with TCP traffic. Similar to the previous experiment, latency increases for the first 25 s of the experiment. Afterwards, latency varies depending on how the packets of the flow are scheduled across the two technologies.

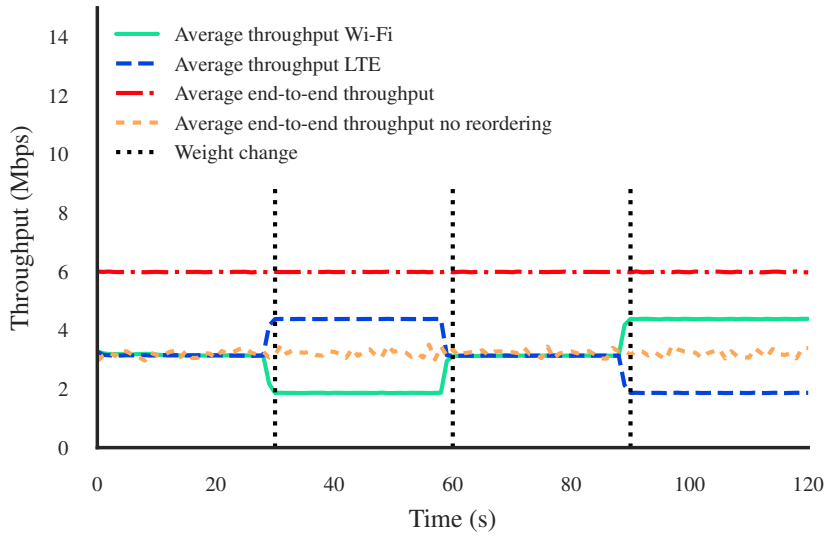

Fig. 13: 6 Mbps TCP flow load-balanced over two technologies with a weight change from 50/50 (Wi-Fi/LTE) to 30/70 to 50/50 to 70/30.

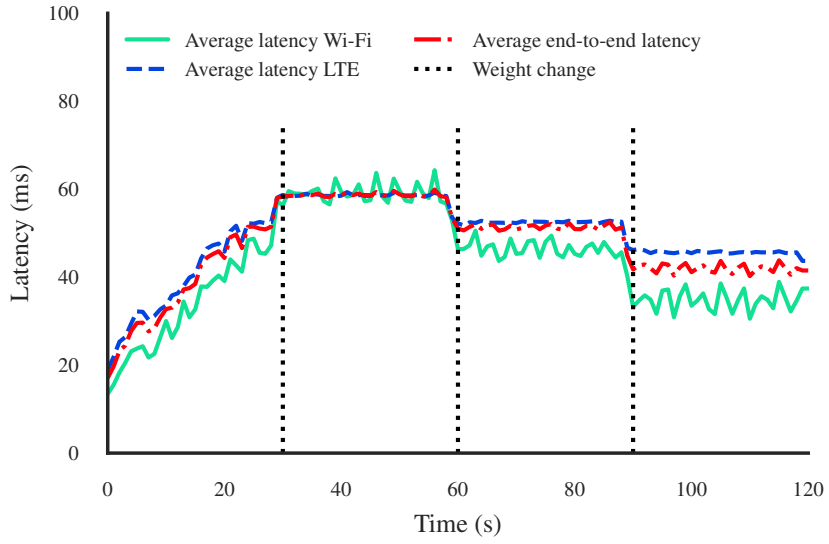

Fig. 14: 6 Mbps TCP flow load-balanced over two technologies with a weight change from 50/50 (Wi-Fi/LTE) to 30/70 to 50/50 to 70/30.

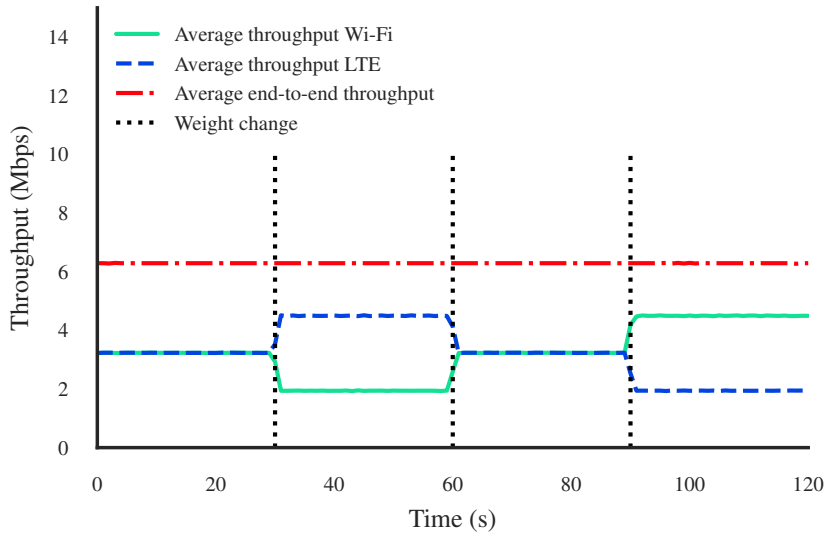

Fig. 15: 6 Mbps UDP flow load-balanced over two technologies with a weight change from 50/50 (2.4 GHz/5 GHz) to 30/70 to 50/50 to 70/30.

Overall, we can say that the increased throughput and flexibility of the packet-based load balancing with reordering, comes at the cost of a slightly increased latency (in comparison to MPTCP). This should be addressed in future work to try to close the gap in latency between ORCHESTRA and MPTCP in this scenario. However, we clearly demonstrate that the technology abstraction of the VMAC layer is working as intended and that, even with TCP, we can precisely (on a packet-level) load balance a flow (both TCP and UDP) among multiple technologies with different characteristics. This MAC-level scheduling can be extended in future work to include packet scheduling across different competing technologies to minimize interference.

7.4 Duplication of critical data in unreliable environments

For the duplication scenario, there are, similar to the load balancing scenario, two continuously active interfaces with LTE and Wi-Fi as their respective technologies. However, instead of balancing the traffic flow, each incoming packet is copied and sent out over both interfaces. We emulate an unreliable environment by dropping packets on each link, with a chance of 25 % per packet per link. A flow of 1 Mbps is used for both TCP and UDP traffic. In both cases, duplicates need to be detected and removed by the deduplication functionality in the VMAC. Below, we will explore the implications and performance of that, compared to MPTCP. For MPTCP the redundant scheduler is used instead

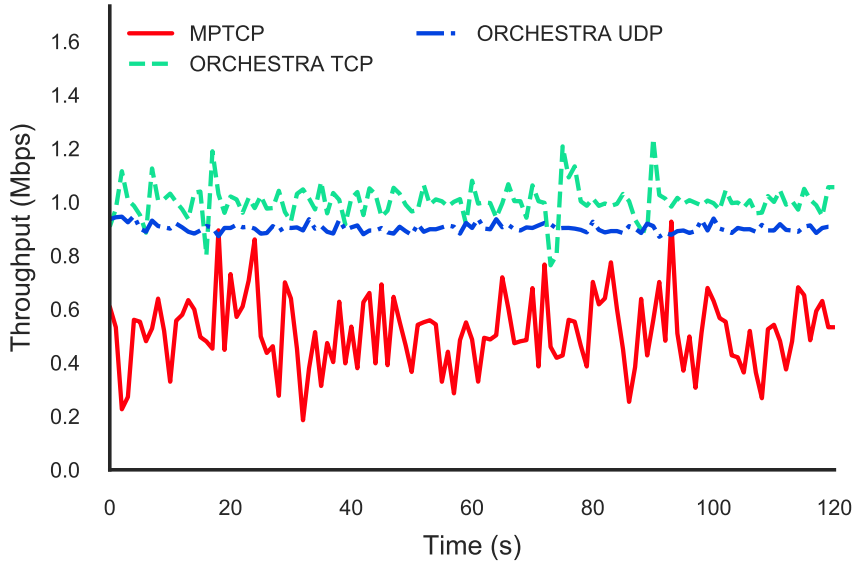

Fig. 16: Duplication performance of MPTCP and ORCHESTRA for throughput.

of the default RTT scheduler [66]. This redundant scheduler sends the data replicated through all of the active subflows available, while back-up subflows are established to send retransmissions.

The results of this scenario are shown in Figures 16 and 17, respectively, for throughput and latency. For MPTCP we notice a large drop in throughput since instead of 1 Mbps, only around 0.5 Mbps is achieved. Furthermore, a corresponding increase in latency can be noted as well. In contrast, ORCHESTRA achieves the full 1 Mbps while the average latency stays below 20 ms as well. This is the case for both TCP and UDP traffic. ORCHESTRA achieves this by duplicating packets transparent to the transport protocol compared to MPTCP, which uses different TCP subflows that each suffers from packet loss. Moreover, the small fluctuations in TCP and UDP throughput (in the ORCHESTRA case) are due to the fact that some packets are still not reaching the receiver as the duplicates can be dropped on both links (with a chance of 12.5 %). However, it is clear that ORCHESTRA significantly increases redundancy, especially in unreliable environments.

8 Conclusions

To cope with the heterogeneity in the networks of today and tomorrow, we propose the ORCHESTRA framework. The framework consists of two key

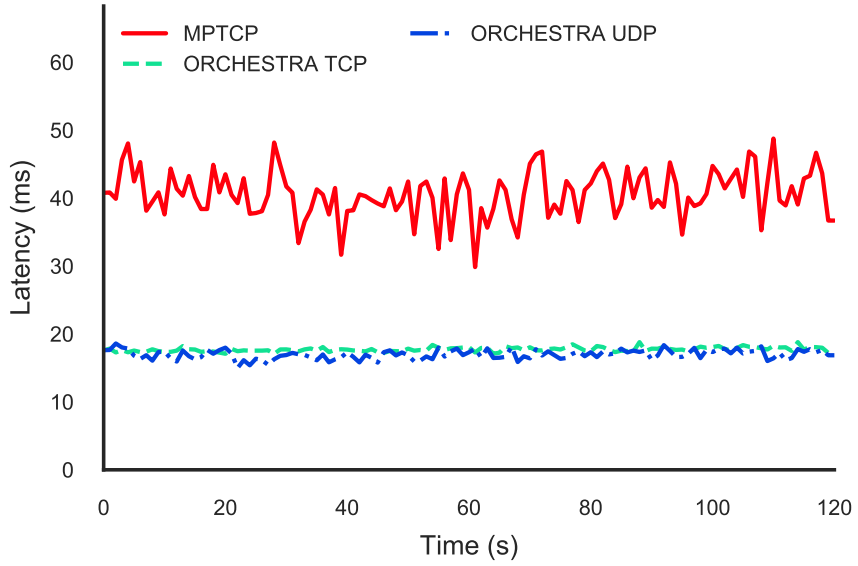

Fig. 17: Duplication performance of MPTCP and ORCHESTRA in terms of latency.

parts: the VMAC layer and a centralized controller. The VMAC offers a single connection point to the upper layers, while transparently bonding over the underlying network technologies. On the other hand, the controller introduces a single point of control and coordination across the entire network. Key features are seamless inter-technology handovers, packet-level load balancing, and duplication. In contrast to many existing approaches, the ORCHESTRA framework is completely independent towards upper (e.g., applications or transport protocols) and lower layers (i.e., technologies), allowing the deployment of the framework in a multitude of applications domains (e.g., LANs, backhauling networks, or satellite networks). An in-depth evaluation, using a real-life prototype setup, demonstrates that the presented features work as intended, and behave similarly or better than the default industry solution MPTCP.

Table 2 revisits the original Table 1 from Section 2.5 and compares the proposed ORCHESTRA solution to the state-of-the-art. It is clear that the main novelty of ORCHESTRA lays in the combination of packet-level control with network-wide coordination. MPTCP is the only solution that offers the same level of control, but does so exclusively between two endpoints and not globally. Due to this fine-grained control, ORCHESTRA can offer more advanced features like the packet-level load balancing, and the duplication of critical data. Furthermore, ORCHESTRA is not limited to certain technologies and application domains, in contrast to approaches like IEEE 1905.1 or LTE-LWA. Future work will include the exploration of novel features (e.g., focusing on the

Features	IEEE 1905.1	SDN-based	LTE-LWA	MPTCP	ORCHESTRA
Network domains	LAN	LAN	LAN-RAN	Any (end-to-end)	**Any**
Technologies	Ethernet, HomePlug, Wi-Fi, MoCA	Wi-Fi, 3GPP	Wi-Fi, LTE	All	**All**
Coordination	Global	Global	Local (within cell)	Between endpoints	**Global**
Control-level	Flow-based	Flow-based	Flow-based	Packet-based (sub-flows)	**Packet-based**
Transport protocols	Any	Any	Any	only TCP	**Any**
Backward compatibility	No	No	Yes	Yes	**Yes**
Vertical Handovers	Yes	Yes	Yes (within cell)	Yes (between subflows)	**Yes**
Needs client changes	Yes	No	Yes	Yes	**Yes**
Products available	Qualcomm Hy-fi	Odin, 5G EmPOWER, ...	Two planned deployments	Android, iOS, Tessares, ...	**Not yet**

Table 2: Comparison of the ORCHESTRA solution to the state-of-the-art.

reduction of energy consumption), the evaluation of ORCHESTRA with other wireless technologies (e.g., LoRa), and the comparison of the deployment of ORCHESTRA in different 3GPP networks or configurations (e.g., considering different MEC setups).

Acknowledgement

Patrick Bosch is funded by FWO, a fund for fundamental scientific research, and the Flemish Government, under grant number 1S56616N.

References

1. Ericsson, "Ericsson mobility Report," Tech. Rep. November, 2017. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-november-2017.pdf
2. Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017-2022," 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf
3. M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 130–137, 2016.
4. R. Wang, H. Hu, and X. Yang, "Potentials and challenges of C-RAN supporting multi-RATs toward 5G mobile networks," *IEEE Access*, vol. 2, pp. 1200–1208, 2014.
5. H. Zimmermann, "OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.
6. M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 4–18, 2015.
7. V. Sagar, R. Chandramouli, and K. P. Subbalakshmi, "Software defined access for HetNets," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 84–89, 2016.
8. S. N. Yang, S. W. Ho, Y. B. Lin, and C. H. Gan, "A multi-RAT bandwidth aggregation mechanism with software-defined networking," *Journal of Network and Computer Applications*, vol. 61, pp. 189–198, 2016.
9. M. Chen, A. Li, W. Liu, and J. Hong, "A Multi-Wireless Bandwidth Aggregation Mechanism in SDN Networks," *Open Electrical & Electronic Engineering Journal*, vol. 9, no. 1, pp. 321–327, 2015.
10. A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," Internet Requests for Comments, RFC Editor, RFC 6824, January 2013. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6824.txt
11. R. V. D. Pol, S. Boele, F. Dijkstra, A. Barczyk, G. van Malenstein, J. H. Chen, and J. Mambretti, "Multipathing with MPTCP and OpenFlow," in *High Performance Computing, Networking, Storage and Analysis (SCC) 2012*, 2012, pp. 1617–1624.
12. H. A. Kim, B. H. Oh, and J. Lee, "Improvement of MPTCP performance in heterogeneous network using packet scheduling mechanism," in *APCC*

2012 - 18th Asia-Pacific Conference on Communications: "Green and Smart Communications for IT Innovation", 2012, pp. 842–847.
13. IEEE Std. 1905.1-2013, "IEEE standard for convergent digital home network for heterogeneous technologies," 2013.
14. C. Hoymann, D. Astely, M. Stattin, G. Wikström, J. F. T. Cheng, A. Höglund, M. Frenne, R. Blasco, J. Huschke, and F. Gunnarsson, "LTE release 14 outlook," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 44–49, 2016.
15. H. L. Maattanen, G. Masini, M. Bergstrom, A. Ratilainen, and T. Dudda, "LTE-WLAN aggregation (LWA) in 3GPP Release 13 & Release 14," in *2017 IEEE Conference on Standards for Communications and Networking, CSCN 2017*, 2017, pp. 220–226.
16. P. Nuggehalli, "LTE-WLAN aggregation [Industry Perspectives]," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 4–6, 2016.
17. T. De Schepper, P. Bosch, E. Zeljkovic, J. Haxhibeqiri, J. Hoebeke, J. Famaey, and S. Latré, "ORCHESTRA: Enabling Inter-Technology Network Management in Heterogeneous Wireless Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1733–1746, 2018.
18. T. De Schepper, S. Latré, and J. Famaey, "Scalable Load Balancing and Flow Management in Dynamic Heterogeneous Wireless Networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 693–706, 2019.
19. T. Meyer, P. Langendörfer, M. Bahr, V. Suraci, S. Nowak, and R. Jennen, "An inter-mac architecture for heterogeneous gigabit home networks," in *20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2009.
20. D. Macone, G. Oddi, A. Palo, and V. Suraci, "A dynamic load balancing algorithm for Quality of Service and mobility management in next generation home networks," *Telecommunication Systems*, vol. 53, no. 3, pp. 265–283, 2013.
21. P. Gallo, K. Kosek-Szott, S. Szott, and I. Tinnirello, "Sdn@home: A method for controlling future wireless home networks," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 123–131, 2016.
22. B. Dezfouli, V. Esmaeelzadeh, J. Sheth, and M. Radi, "A review of software-defined wlans: Architectures and central control mechanisms," *IEEE Communications Surveys & Tutorials*, 2018.
23. L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, "Building an sdn enterprise wlan based on virtual aps," *IEEE Communications Letters*, vol. 21, no. 2, pp. 374–377, 2016.
24. L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise wlans with odin," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 115–120.

25. R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming Abstractions for Software-Defined Wireless Networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.
26. E. Coronado, S. N. Khan, and R. Riggio, "5g-empower: A software-defined networking platform for 5g radio access networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 715–728, 2019.
27. A. Mukherjee, J. F. Cheng, S. Falahati, H. Koorapaty, D. H. Kang, R. Karaki, L. Falconetti, and D. Larsson, "Licensed-Assisted Access LTE: Coexistence with IEEE 802.11 and the evolution toward 5G," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 50–57, 2016.
28. D. Laselva, D. Lopez-Perez, M. Rinne, and T. Henttonen, "3GPP LTE-WLAN Aggregation Technologies: Functionalities and Performance Comparison," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 195–203, 2018.
29. F. M. Abinader, E. P. Almeida, F. S. Chaves, A. M. Cavalcante, R. D. Vieira, R. C. Paiva, A. M. Sobrinho, S. Choudhury, E. Tuomaala, K. Doppler, and V. A. Sousa, "Enabling the coexistence of LTE and Wi-Fi in unlicensed bands," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 54–61, 2014.
30. N. Zhang, S. Zhang, S. Wu, J. Ren, J. W. Mark, and X. Shen, "Beyond coexistence: Traffic steering in LTE networks with unlicensed bands," *IEEE Wireless Communications*, vol. 23, no. 6, pp. 40–46, 2016.
31. X. Wang, S. Mao, and M. X. Gong, "A survey of lte wi-fi coexistence in unlicensed bands," *GetMobile: Mobile Computing and Communications*, vol. 20, no. 3, pp. 17–23, 2017.
32. P. Nuggehalli, "LTE-WLAN aggregation [Industry Perspectives]," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 4–6, 2016.
33. P. Sharma, A. Brahmakshatriya, T. V. Pasca S., B. R. Tamma, and A. Franklin, "LWIR: LTE-WLAN Integration at RLC Layer with Virtual WLAN Scheduler for Efficient Aggregation," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.
34. Y.-B. Lin, Y.-J. Shih, and P.-W. Chao, "Design and Implementation of LTE RRM With Switched LWA Policies," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1053–1062, 2018.
35. G. mobile Suppliers Association (GSA), "LTE in Unlicensed Spectrum: Trials, Deployments and Devices," 2018. [Online]. Available: https://www.sata-sec.net/downloads/GSA/180117-GSA-Unlicensed-spectrum-report-Jan-2018.pdf
36. M. Networks, "Truffle - Broadband Bonding Appliance." [Online]. Available: https://www.mushroomnetworks.com/truffle/
37. Peplink, "Multi-WAN Internet Load Balancer." [Online]. Available: https://www.peplink.com/technology/internet-load-balancing/
38. Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "A first analysis of multipath TCP on smartphones," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and*

Lecture Notes in Bioinformatics), vol. 9631, no. September 2015, pp. 57–69, 2016.
39. C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop - CSWS '14*, 2014, pp. 27–32.
40. K. W. Choi, Y. S. Cho, J. W. Lee, S. M. Cho, J. Choi *et al.*, "Optimal load balancing scheduler for mptcp-based bandwidth aggregation in heterogeneous wireless environments," *Computer Communications*, vol. 112, pp. 116–130, 2017.
41. J. Kellokoski, "Real-life multipath tcp based make-before-break vertical handover," in *2013 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2013, pp. 000 252–000 256.
42. C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath tcp," in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*. ACM, 2012, pp. 31–36.
43. R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "Mptcp is not pareto-optimal: performance issues and a possible solution," *IEEE/ACM Transactions on Networking (ToN)*, vol. 21, no. 5, pp. 1651–1665, 2013.
44. F. Rebecchi, M. D. De Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 580–603, 2015.
45. Tessares, "Hybrid Access Networks with MPTCP." [Online]. Available: https://www.tessares.net/
46. P. Hintjens, *ZeroMQ: messaging for many applications*. O'Reilly Media, Inc., 2013.
47. S.-q. Lee and J.-u. Kim, "Local breakout of mobile access network traffic by mobile edge computing," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, 2016, pp. 741–743.
48. F. Giust, G. Verin, K. Antevski, J. Chou, Y. Fang, W. Featherstone, F. Fontes, D. Frydman, A. Li, A. Manzalini *et al.*, "Mec deployments in 4g and evolution towards 5g," *ETSI White Paper*, vol. 24, pp. 1–24, 2018.
49. T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
50. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
51. F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, "Multi-access edge computing: The driver behind the wheel of 5g-connected cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, 2018.

52. G. T. 23.501, "Technical specification group services and system aspects: System architecture for the 5g system, stage 2 (release 15),v1.3.0," 2017.
53. Y. Yu, "Sdn-based local breakout for mobile edge computing in radio access network," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.
54. S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "Mec in 5g networks," 2018. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf
55. F. Khan, "Mobile Internet from the Heavens," *arXiv preprint arXiv:1508.02383*, p. 8, 2015.
56. S. Xu, X.-w. Wang, and M. Huang, "Software-Defined Next-Generation Satellite Networks: Architecture, Challenges, and Solutions," *IEEE Access*, vol. 4, no. c, pp. 1–1, 2018.
57. S. Chen, J. Hu, Y. Shi, and L. Zhao, "Lte-v: A td-lte-based v2x solution for future vehicular network," *IEEE Internet of Things journal*, vol. 3, no. 6, pp. 997–1005, 2016.
58. S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of Things (IoT) communication protocols: Review," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690.
59. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
60. P. A. Frangoudis, G. C. Polyzos, and V. P. Kemerlis, "Wireless community networks: An alternative approach for nomadic broadband network access," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 206–213, 2011.
61. E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.
62. "Lede docs." [Online]. Available: https://lede.readthedocs.io/en/latest/
63. N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "Openairinterface: A flexible platform for 5g research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.
64. "Multipath tcp - linux kernel implementation." [Online]. Available: https://multipath-tcp.org/pmwiki.php
65. T. De Schepper, S. Latré, and J. Famaey, "Flow Management and Load Balancing in Dynamic Heterogeneous LANs," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 693–706, 2018.
66. I. Lopez, M. Aguado, C. Pinedo, and E. Jacob, "Scada systems in the railway domain: enhancing reliability through redundant multipathtcp," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2305–2310.

Biographies

Tom De Schepper is a Senior researcher associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from the University of Antwerp in 2015 and a Ph.D. in Computer Science from the same university titled "Multi-technology Management of Heterogeneous Wireless Networks". His research resulted in 17 articles published in international peer-reviewed journals and conference proceedings, as well as in a submitted patent application. His main interests are in the areas of wireless management and applied artificial intelligence.

Patrick Bosch is a Ph.D. researcher associated with imec and the University of Antwerp, Belgium. He received his Diploma degree in Computer Science from the University of Stuttgart, Germany in 2014. His current research focuses on network orchestration and management of different wireless technologies to improve Quality of Service, interference modeling for wireless networks, and network management. His research resulted in 12 articles published in international peer-reviewed journals and conference proceedings, as well as in two submitted patent applications

Jakob Struye is a Ph.D. researcher associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from the University of Antwerp in 2015. He has experience in MPTCP on Android, and time series prediction using neuroscience-inspired approaches. His current work focuses on ultra-low latency multi-gigabit communication using mmWave networks.

Carlos Donato received his B.Sc in Telecommunications Engineering (2013), M.Sc in Telematics Engineering (2015) and Ph.D, in Telematics Engineering (2018) from Universidad Carlos III de Madrid, Department of Telematics Engineering. Currently, he holds a post-doc position at IDLab - imec in Antwerp. His research interests lie in wireless communications, especially LTE and IEEE 802.11 technologies, resources on demand, energy efficiency, design of network protocols, Software Defined Networking, Cloud-RAN, and AI techniques applied to wireless communciations.

Jeroen Famaey is an assistant professor associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from Ghent University, Belgium in 2007 and a Ph.D. in Computer Science Engineering from the same university in 2012. He is co-author of over 90 articles published in international peer-reviewed journals and conference proceedings, and 10 submitted patent applications. His research focuses on performance modeling and optimization of wireless networks, with a specific interest in low-power, dense and heterogeneous networks.

Steven Latré is an associate professor at the University of Antwerp and director at the research centre imec, Belgium. He is leading the IDLab Antwerp research group (100+ members), which is performing applied and fundamental research in the area of communication networks and distributed intelligence. His personal research interests are in the domain of machine learning and its application to wireless network optimization. He is author or co-author of more than 100 papers published in international journals or in the proceedings of international conferences.