The wireless autonomous spanning tree protocol for multishop wireless body area networks

Reference:
Braem Bart, Latré B., Moerman I., Blondia Christian, Demeester P..- *The wireless autonomous spanning tree protocol for multishop wireless body area networks*
Mobile and Ubiquitous Systems: Third Annual International Conference on Networking & Services 2006, San José, California, USA - s.l., 2006, p. 1-8
Handle: http://hdl.handle.net/10067/658820151162165141

The Wireless Autonomous Spanning tree Protocol for Multihop Wireless Body Area Networks

Bart Braem[1], Benoît Latré[2], Ingrid Moerman[2], Chris Blondia[1], Piet Demeester[2]
[1] University of Antwerp – IBBT vzw – Department of Mathematics and Computer Science – PATS
Middelheimlaan 1, B-2020 Antwerpen, Belgium, {bart.braem, chris.blondia}@ua.ac.be
[2]Ghent University – IBBT vzw – IMEC vzw – Department of Information Technology (INTEC) – IBCN
Gaston Crommenlaan 8, B-9050 Ghent, Belgium, {benoit.latre, ingrid.moerman, piet.demeester}@intec.ugent.be

***Abstract*— Wireless body area networks (WBANs) have gained a lot of interest in the world of medical monitoring. Current implementations generally use a large single hop network to connect all sensors to a personal server. However recent research pointed out that multihop networks are more energy-efficient and even necessary when applied near the human body with inherent severe propagation loss. In this paper we present a slotted multihop approach to medium access control and routing in wireless body area networks, the Wireless Autonomous Spanning tree Protocol or WASP. It uses crosslayer techniques to achieve efficient distributed coordination of the separated wireless links. Traffic in the network is controlled by setting up a spanning tree and by broadcasting scheme messages over it that are used both by the parent and the children of each node in the tree. We analyze the performance of WASP and show the simulation results.**

I. Introduction

In the last few years, wearable health monitoring systems have gained the attention of various researchers in order to cope with the rising costs of the health care system. An important task of such a system is to collect physiological parameters like the heartbeat, body temperature, SpO_2-levels,. . . To this purpose, several sensors are placed in clothes, directly on the body or under the skin of a person. If these sensors are equipped with a wireless interface, a new type of network spanning the entire body takes form: a wireless on-body network or a Wireless Body Area Network (WBAN) [1]. A WBAN provides continuous health monitoring and real-time feedback to the user and the medical personnel. Furthermore, the measurements can be recorded over a longer period of time which improves the quality of the measured data [2]. Next to sensor devices, actuators or actors are used. These take some specific actions according to the data received from the sensors or through interaction with the user. For example, an actuator can be fitted out with a built-in reservoir and pump for administering the correct dose of insulin to diabetics based on the measurements of glucose level received from the sensors. Next to these sensors and actors, each WBAN has a sink or personal server such as a PDA. It is attached to the hip or wrist and is responsible for the interaction with the user.

Current research efforts on WBAN largely focus on the development of new sensors and new radio interfaces. One of the major concerns is to lower the energy usage of the different devices in order to extend the life time and therefore the user-friendliness of the system. Several methods exist to achieve this purpose: the development of low-power radio interfaces and low-power sensors and the use of optimized network protocols which lower the energy consumption even further. Apart from medium access control another major point of focus within WBAN research are propagation issues. It was found that radio signals experience severe path loss close to the body [3]. Consequently, direct communication between all nodes will come at high transmission cost or will prove to be impossible.

In this paper, we introduce a new approach for communication in multihop wireless body area networks: the Wireless Autonomous Spanning tree Protocol or WASP. This protocol incorporates slotted medium access and the automatic setup of the different routes. To that end, a spanning tree is set up automatically. This spanning tree is subsequently used to route the data toward the sink and to assign the different slots in a distributed manner. Thus, WASP uses a crosslayer approach where medium access control and routing are handled by the same spanning tree and therefore a higher throughput and lower energy consumption can be achieved.

The remainder of this paper is organized as follows. An overview of related work is given in section II. In section III we will give a general overview of WASP by using a small example network. An analysis of the performance is given in section IV. In section V the implementation and experimental results of the protocol are stated. Finally, section VI gives some future directions and section VII concludes the paper.

II. Related Work

Protocols for WBANs can be divided in *intra-body communication* and *extra-body communication* ones. The first control the information handling between the sensors or actuators and the sink, the latter ensure communication between the sink and an external network. Doing so, the medical data of the patient at home can be remotely consulted by a doctor. The WASP-protocol addresses the issue of intra-body communication, therefore we will focus this section to this type of communication.

In the last year, several implementations of WBANs have been proposed. In [4] a system architecture of a wireless body area sensor network is presented. This system architecture both handles the intra- and extra-body communication. The communication between the sensors and the sink is slotted. The slots are synchronized using beacons periodically sent by the sink. A similar system is proposed in [1] where the authors use IEEE 802.15.4/ZigBee compliant radios. This type of radio is also used in the CodeBlue-project [5] and in [6]. The BASUMA-project (Body Area System for Ubiquitous Multimedia Applications) [7] aims at developing a full platform for WBANs. As communication technique, a UWB-frontend is used and a MAC-protocol based on IEEE 802.15.3. This protocol also uses time frames divided into contention free periods (with time slots) and contention access periods (CSMA/CA). In [8] a collision free real-time MAC-protocol is implemented. This protocol divides time into frames in which only one node is allowed to transmit. The scheduling order is derived by applying the Earliest Deadline First algorithm.

Most of the projects above only consider direct communication between the sensors and the sink. However, the propagation loss around the human body is very high. Generally, in wireless networks, it is known that the transmitted power drops off with d^n where d represents the distance between the sender and the receiver and n the path loss coefficient. In free space, n has a value of 2. Several researchers have been investigating the path loss around the human body [3], [9]–[12]. All of them come to the conclusion that the radio signals experience large losses as n reaches a value between 4 and 7. Hence, direct communication between the sensors and sink will not always be possible, especially when one wants to lower the transmission power of the radio in order to save the precious energy available in the nodes. The projects above do not consider multihop communication.

A lot of research has been done in the area of sensor networks. As body area networks share some characteristics with this type of network, we will consider some techniques where multihop communication is used. Radio energy consumption in sensor networks is reduced by controlling the power and duty cycle of the radio. Scheduled protocols, such as S-MAC [13], uses scheduling to coordinate sleeping among neighboring nodes to avoid idle listening. A preamble sampling technique is used in WiseMAC [14] where a node regularly samples or polls the medium for a very brief time to check whether a packet needs to be received. An ultra-low duty cycle MAC that combines scheduling and channel polling is presented in [15]. Another technique is TDMA where contention-introduced overhead and collisions are avoided.

III. Protocol Description

A. General overview

Sensors in a WBAN can be considered as generating CBR traffic where traffic requirements do not change quickly over time [16]. Therefore WASP uses coordination of the medium access.

WASP is a slotted protocol that uses a spanning tree for medium access coordination and traffic routing. Each node will tell its children in which slot they can send their data by using a special message: a WASP-scheme. This WASP-scheme is unique for every node and constructed in the node sending the scheme. A distinguishing property of WASP is the dual usage of the WASP-schemes by exploiting the broadcast nature of wireless links. A node uses the schemes to control the traffic of its children and simultaneously to request more resources from its parent for these children. This minimizes the coordination overhead because each scheme is used by the parent and the children of the sending node. Everything the node has to know to generate this scheme can be obtained by listening to the WASP-schemes coming from its parent node (i.e. one level up in the tree) and from its children (i.e. one level lower in the tree). Consequently, the division of the time slots is done in a distributed manner.

Hereafter a short overview is given of the operation of the protocol in a steady-state situation. In this section we assume a spanning tree has already been constructed and is used to propagate the data to the sink. The time-axis is divided in different cycles (further referred to as WASP-cycles) that consist of a fixed number of time slots. In a WASP-cycle, each node is allowed to send its data and/or to forward data received in the previous cycle to the next node. At the beginning of each cycle, the sink sends its WASP-scheme to its children. This WASP-scheme informs the sink's children when they can send their information. These children respond to the scheme by sending out their own WASP-scheme in their designated time slots. These WASP-schemes are based on the WASP-scheme of the sink and on the requirements of their children. Thus each node right below the sink calculates its own WASP-scheme and sends it to its own children which form the second level. On their turn, these nodes send out the WASP-scheme and so on. Doing so, each node will know when it can send its data without the need for a device that centrally calculates the distribution of the time slots. A more elaborate example will be given in section V-B.

B. WASP-schemes

Each node will send a WASP-scheme to its children to inform them when they are allowed to use the link. In this section, we will describe what elements can be found in such a WASP-scheme. In order to focus our thoughts, the simple example network of figure 1 will be used. The lines between the nodes indicate the tree structure of the network. All communication is wireless and the tree is constructed in such a way that the nodes only can hear their parent, their siblings (i.e. children of the same parent) and their children. This assumption simplifies the protocol explanation. We assume that in this example each node only has one data packet to send to the sink per cycle. Of course, in general, nodes will be allowed to send multiple data packets per cycle.

The WASP-scheme of the sink (node S) is as follows:

$$\overbrace{S}^{1}\underbrace{AB}_{2}\overbrace{.^{3}}^{3}\underbrace{ABB}_{4}\overbrace{X}^{5}\underbrace{11101}_{6} \qquad (1)$$

Slot	Level 0	Level 1	Level 2
0	S: WASP-scheme: SAB.^{3}ABBX-10011		
1		A: send data + WASP-scheme: A.^{1}CX-1	
2		B: send data + WASP-scheme: BDEX-11	
3			C: send data + WASP-scheme: C.^{1}X-
			D: send data + WASP-scheme: D.^{2}X-
4		Contention slot A	E: send data + WASP-scheme: E.^{1}X-
5		Contention slot B	Contention slot C
6		A: forward data	Contention slot D
			Contention slot E
7		B: forward data	
8		B: forward data	
9	Contention slot S		

Fig. 2. Steady-state WASP cycle for the example network

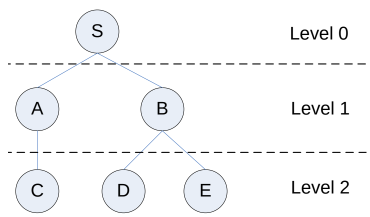


Fig. 1. Example network. The lines indicate the tree structure.

where

1 = The address of the sending node;
2 = Every child of the sink is designated one slot. In that slot, the node sends its WASP-scheme followed by data;
3 = A silent period of 3 slots. In this period, the child nodes receive data from their children that they need to forward to the sink;
4 = The children forward the received data to the sink;
5 = Contention slot;
6 = ACK-sequence.

First, each of the child nodes of the sink is awarded 1 slot, part 2 of equation (1). In this slot, the child nodes will send their WASP scheme and data. This WASP-scheme will be heard by the sink and by the children of the node. Consequently, the sink is informed of its child node's traffic requirements and is capable of calculating the duration of the silent period for the next cycle.

During the silent period, part 3 of equation (1), the sink will not be sending or receiving any messages as this period is used by the sink's children to receive data or messages from their children. Doing so, interference between different levels will be limited.

Following the silent period, a sequence is given in which the children can forward the received data to the sink, part 4 in (1). After that, a contention slot is inserted in order to allow new nodes to join the network, part 5 in (1). The scheme is completed by the piggybacked acknowledgments, represented by simple bits, part 6 in (1). This will be explained in section III-H.

A slightly different WASP-scheme is used by the other nodes, for example node A:

$$\overbrace{A}^{1}\overbrace{.^{1}}^{3}\underbrace{C}_{4}\overbrace{X}^{5}\underbrace{1}_{6} \tag{2}$$

A first difference is the duration of the silent period. Whereas with the sink the silent period was used to allow the nodes of level 2 to send their data to level 1, the silent period is now used to indicate the nodes of level 2 when they should remain silent in order not interfere with the ongoing communication on the higher level. In other words, in the silent period of the sink, the sink will be silent, and in the silent period of the other nodes, their child nodes will be silent. This reasoning brings us to the second difference with (1): the omission of part 2. The nodes below level 1 are not allowed to send their data immediately as all siblings of their parent node need to send their WASP-scheme first in order to reduce delay.

In the other levels (level three and lower), the nodes can start sending when the transmission of the parent node has ended, but the same principle holds: the silent period indicates when the child nodes should remain silent (and consequently can turn off their radio). A more elaborate example with a larger number of levels can be found in section V-B.

An overview of the entire WASP-cycle of this example can be found in figure 2. In each timeslot, it is shown what action is taken. In timeslot 0, the sink S, at level zero, sends its WASP-scheme. In the next timeslot, sensor A sends its WASP-scheme and its data. Further, in timeslot 3, we can see that node C and D can send simultaneously.

C. Duration of the silent period

The duration of the silent period for the sink SP_S equals the number of slots needed to send the data from level two to level one. Thus, it can be calculated as follows:

$$SP_S = \max_{\forall i \in Ch_S} \left(\sum_{j \in V_i} TS_j \right) + 1 \tag{3}$$

where Ch_i are the children of node i, TS_j is the number of slots needed by node j to send its own data and V_i denotes the nodes in the tree below node i, node i not included. One extra slot is inserted in order to allow the presence of a contention slot.

The length of the silent period of the other nodes is calculated based on the parent's WASP-scheme and thus is different below level 1:

$$SP_{level_1} = \text{slot \# start silent period - slot \# node's first occurrence - 1} \quad (4)$$

$$SP_{level_{i>1}} = \text{slot \# contention slot - slot \# node's first occurrence} \quad (5)$$

For example, the length of the silent period of node A equals the slot number of the start of the silent period (= 3) minus the slot number of the first occurrence of node A in the WASP-scheme of the sink (= 1) which gives a silent period of 1 slot. The length of the silent period of node C is the slot number of the contention slot (= 4) minus the slot number of the first occurrence of node C (= 3) which equals 1 slot. Indeed, a silent period of one slot is necessary because node A still has a contention slot.

D. Contention slots

A contention slot allows nodes to join the network by sending a JOIN-REQUEST. It is possible to omit this slot for a few consecutive WASP-cycles to increase the maximum throughput. However, in order to keep the connection time reasonably low, at least one contention period per n milliseconds is required. Moreover, to support a notion of mobility a high frequency of contention slots is required. As it is possible that more than one node can send during these contention slots, a random delay is inserted before each node's transmission. Doing so, the probability of collisions during these slots is decreased. There is no special mechanism that detects collisions. If a node that wants to join the network does not hear its address in the next WASP-scheme of its parent, it assumes that a collision has occurred and sends a new JOIN-REQUEST in the next contention slot.

E. Heterogeneous data rates

In the example above, each node was only assigned one data packet or slot per cycle. In more general networks, some nodes will require a larger data rate then others. In such cases, these nodes will be assigned multiple slots per cycle. The desired number of slots can be requested when joining the network. Sometimes the wanted data rate of a node can change over time, e.g. when more accurate measurements are desired over a certain period of time. The node will ask for more slots in its WASP-scheme. This will be seen by the parent node that will assign more slots in its next WASP-scheme.

F. Building and maintaining the spanning tree

The following steps are taken to let nodes join the network:

1) The new node scans the wireless medium for a certain time and picks up the WASP-scheme of the nodes in range.
2) Based on these received WASP-schemes, the new node determines which node will be the best parent, based upon the node's requirements (low delay, reliability, a weighted average, ...).
3) During the contention slot of the chosen parent, the new node sends a JOIN-REQUEST.
4) The parent receives this request and adds the new node in the next WASP-scheme.

If for a certain time a child node does not receive a WASP-scheme, a new parent is chosen using the same procedure. Each node should periodically check which nodes are in its neighborhood and choose a new parent if a better one is around. The node does not have to notify its parent. If a parent doesn't receive the WASP-scheme of a child for n consecutive times, it assumes that the child is no longer part of the network and doesn't include it in the WASP-scheme anymore.

These basic tree maintenance mechanisms support low mobility.

G. Routing and addressing

In order to facilitate the forwarding to the different nodes, the tree structure of the network can be used for addressing. However this poses problems as nodes might change attachment points in the tree. A more general solution is manually asking the sink for a unique address. The sink receives traffic from all sensors so it knows the addresses in use. A WASP address is composed of 6 bits. The sink gets address 000000, the first node will get 000001 and so on. Using 6 bits the number of nodes is limited to 64, which is reasonable due to the nature of a BAN (see section IV-D).

In a WASP scheme 1 byte describes each slot. The first bit denotes the slot type. 0 stands for a regular slot where data is sent and 1 for a special slot. In case of a data slot the second bit denotes the traffic direction: 0 for regular traffic to the sink and 1 for traffic in the other direction or more general traffic that requires routing. The other 6 bits define the actual address of the node that is permitted to send in the slot. In case of a special slot, the 7 remaining bits are used to define the length of the silent period. E.g. 1.0000011 denotes a silent period of length 3. If all the bits are set to 1, the slot is a contention slot instead of a silent period.

Most traffic in a WBAN flows from the nodes to the sink. Traffic in the other direction can be supported by setting the second bit in the WASP-scheme. A node will add the source address with that bit set to the WASP scheme. Each child then decides whether it is on the path to the destination. If so it turns on its radio to receive the packet. Routing can be done by using a technique similar to learning bridges. Nodes record the addresses in traffic passing by and route packets from the sink to the nodes using that information.

H. Acknowledgments

At the end of each WASP-scheme, an ACK-sequence for the previous WASP-cycle is sent. It contains a bit for each slot in which data was sent in the previous cycle. A 0 denotes that the packet was not received correctly, a 1 denotes success.

The node that receives the WASP-scheme, say node N, will check the ACK-sequence. If the position of a 0 corresponds to one of the slots where N sent data, the node will resend that data in this cycle. Its parent will already include an extra slot for node N.

This acknowledgment scheme is quite weak but it suffices largely. Due to the absence of contention data loss will be limited to interference problems.

I. Synchronization

The nodes in the tree need to be synchronized in order to avoid shifting of the start of slots between nodes. However the recurring cycles allow for resynchronization at the beginning of each cycle. Each node should wake up some milliseconds before the start of slot to avoid these issues.

IV. Performance Analysis

In this section, we present the performance analysis of our proposed protocol. We will address performance issues such as minimum delay, maximum throughput and sleeping time of the nodes.

A. Maximum overall throughput

The maximum overall throughput of the protocol can be seen as the percentage of the useful traffic (or goodput) that can be sent over the network. As we are working in a multi hop environment, the maximum throughput will mostly depend on the number of nodes in the network and the number of levels in the tree. Indeed, the maximum amount of data that can be sent to the sink per WASP-cycle depends on the number of nodes in the first level and the length of the silent period. As mentioned in section III-B, the silent period of the sink allows the nodes of level one to receive their data. This means that if we can lower the duration of the silent period, the maximum throughput will rise. Thus by minimizing the number of children of the nodes of level one, the maximum throughput can be improved. Generally, the throughput in terms of percentage can be found as

$$TP = \frac{\text{\# slots where data sent to sink}}{\text{length of a WASP-cycle}} \tag{6}$$

where the length of the WASP-cycle is expressed in slots and depends on the length of the silent period.

In the example of figure 1, we see that per WASP-cycle 5 packets can be sent to the sink and the total length is 10 timeslots. Thus, we have a maximum throughput of $\frac{5}{10}$ or 50%. This seems to be a low number, but we have to keep in mind that we are working in a multi hop environment.

The following formula determines the length of a WASP-cycle T_{WC}:

$$\begin{aligned} T_{WC} \quad = \quad & \text{\# data children } L_1 + SP_S + \\ & \text{forwarding data of } L_2 \text{ from } L_1 \text{ to } L_0 + 2 \end{aligned} \tag{7}$$

where L_i represents level i. The two extra time slots added at the end are used for the transmissions of the sink's WASP-scheme and the contention slot at the end.

The duration of the forwarding period equals the number of timeslots needed to send the data of each node. Using (3), we can rewrite this formula as

$$T_{WC} = \sum_{i \in V} TS_i + \max_{\forall i \in Ch_S} \left(\sum_{j \in V_i} TS_j \right) + 3 \tag{8}$$

Thus, (6) can be reformulated as

$$TP = \frac{\sum_{i \in V} TS_i}{\sum_{i \in V} TS_i + \max_{\forall i \in Ch_S} \left(\sum_{j \in V_i} TS_j \right) + 3} \tag{9}$$

and highly depends on the number of nodes and the structure of the tree. In order to evaluate this formula, we will distinguish two extreme cases: 1) all nodes can communicate directly with the sink and 2) all nodes are in different levels. We further assume that each node only has 1 packet to send per WASP-cycle. This means that in both cases, the nominator of (9) equals the number of nodes in the network (x). In the first case, the second term in the denominator of (9) equals zero and in the second case to $x - 1$. The boundaries for 50 nodes are 94% in the first case and 49% in the second case. The latter has the largest silent period. Thus, in order to increase the throughput, the silent period should not be too long. This can be achieved by setting up a balanced tree.

B. Delay limits

The experienced delay depends on the number of levels present in the network. Indeed, a node can send his data only up one level during each WASP-cycle. The only exception is to be found at level 1, where the sink's children can first receive the data from their children and then forward the data.

We can define an upper and lower bound for a node i:

$$\text{Lower bound} = \max\left((level\ node_i - 2) \cdot T_{WC}, 1\right) \tag{10}$$

$$\text{Upper bound} = \max\left((level\ node_i - 1) \cdot T_{WC}, 1\right) \tag{11}$$

The maximum function is needed as the delay can not be lower than 1 slot.

The maximum delay over the whole network can be expressed as follows, assuming that the network has at least 2 levels:

$$\text{maximum delay} = \left(\max_{\forall i \in V} (level\ node_i) - 1 \right) \cdot T_{WC}. \tag{12}$$

Summarizing, if we want to have a high throughput, we should minimize the length of the silent period and for a low delay minimize the number of levels. These two conditions do not contradict, therefore a high throughput can be achieved while preserving the low delay.

C. Sleeping nodes

When the nodes have heard the WASP-scheme of their parent, they can turn their radio off in the slots where they are not involved in the communication. This allows for energy saving. In the example scheme of figure 2, the sink can turn its radio off in the silent period as it will not receive data from its child nodes. Thus, the radio can be turned off 3 slots. Node A can also turn its radio off in its silent period, when it knows that its siblings are sending and when none of its children is allowed to send data. So, node A can sleep 5 slots. The following formula can be used to calculate the number of time slots in which node i can sleep or has to wait for the WASP-scheme of its parent $T_{sleep+wait,i}$:

$$T_{sleep+wait,i} = T_{WC} - (\sum_{j \in V_i} TS_j + 1) - TS_i - \sum_{j \in V_i} TS_j - 1. \tag{13}$$

The second term refers to receiving the data from its lower layers (including the contention slot), the third and fourth term bring the sending of the data into account and in the last term, the node is listening to the scheme of its parent. This formula gives the upper bound of the number of slots a node can sleep. Indeed, if a node would perfectly know when a slot starts it could turn on its radio at the beginning of each slot for a very short time.

D. Scalability

The number of nodes in a WBAN is limited by nature of the network. It is expected that the number of nodes will be in the range of 20–50 [4], [9]. Our address structure, see section III-G, supports up to 64 addresses which is therefore sufficient for WBANs. If more addresses need to be supported, the proposed address structure can be altered. Instead of 6 bits, 14 bits can be used. This will however negatively affect the amount of overhead generated by WASP.

Further, the more nodes in the network, the more data will be sent. This will negatively affect the maximum throughput per node.

E. Interference

Although WASP is slotted interference can arise from nearby subtrees. The resulting interference can be minimized by randomizing WASP schemes. This randomization is not unlimited, e.g. the position of the contention slot is currently fixed, but it can reduce the interference probability.

V. Implementation and validation

A. Implementation

The protocol was implemented in nsclick, a simulator that allows Click Modular Router instances to run in the ns-2 network simulator [17].

Figure 3 gives an overview of the different Click elements and interconnections in our implementation. The packets coming from the network are dumped to ns-2 traces and then classified according to their type. Data is processed or forwarded, joins are handled during contention slots and the WASP-schemes are analyzed. If a node does not have a parent it will react on a scheme with a join, otherwise a new scheme is prepared.

Two extra elements, without any incoming or outgoing ports, are used: WASPTiming, responsible for the slot timing issues, and WASPInfobase which stores useful information needed by all elements in a node.

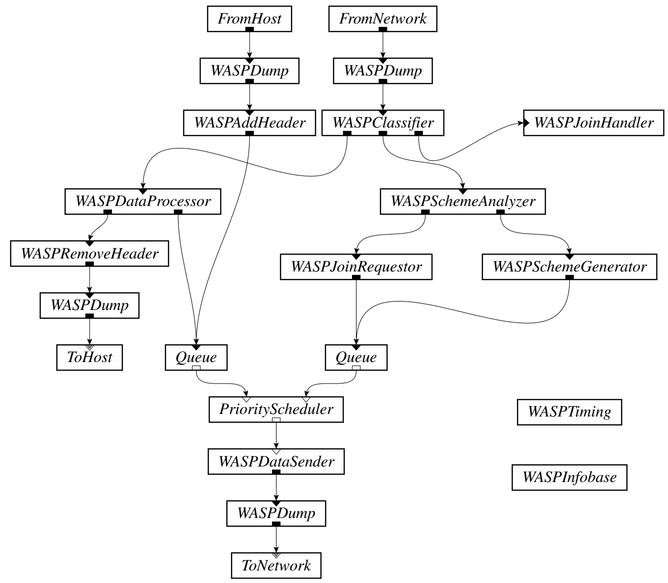


Fig. 3. Overview of different click elements and interconnections

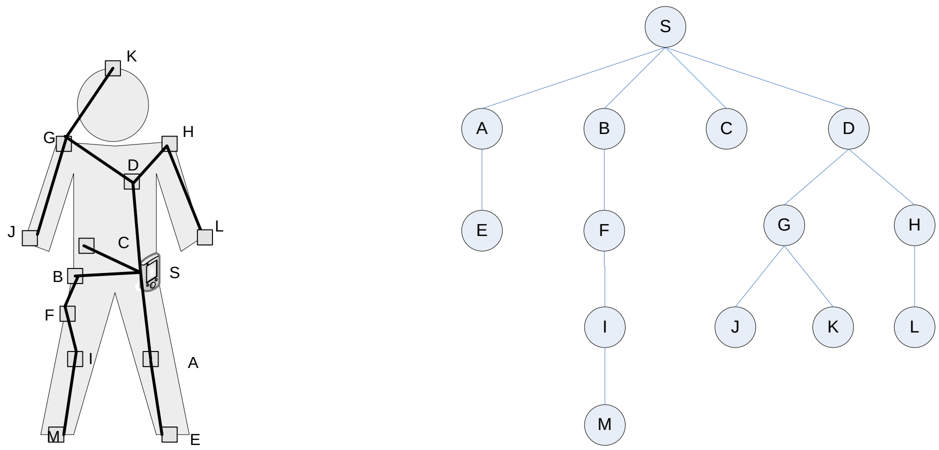
As mentioned in section II, the path loss between two nodes on the body is highly different from the path loss in free space. Hence, we have adapted the propagation model of ns-2 to the path loss model of [3] which accurately models the path loss near a flat phantom for muscle tissue at 2.4 GHz. In our simulations we have used a narrowband radio working at 2.4 GHz with a data rate of 1 Mbps.

B. Example scenario

In this section, a more elaborate example is given and discussed. The network used is that of figure 4, where 4(a) shows the network on a body and 4(b) a more generic view of the network in the form of a spanning tree. We assume that each node has data to send, which is reasonable as sensor traffic is normally constant bit rate traffic. The sequence of the nodes is not randomized in this example and all sensor data is sent to a sink. As an example, (14) shows the WASP-scheme of the sink. The numbers correspond to the ones used in (1).

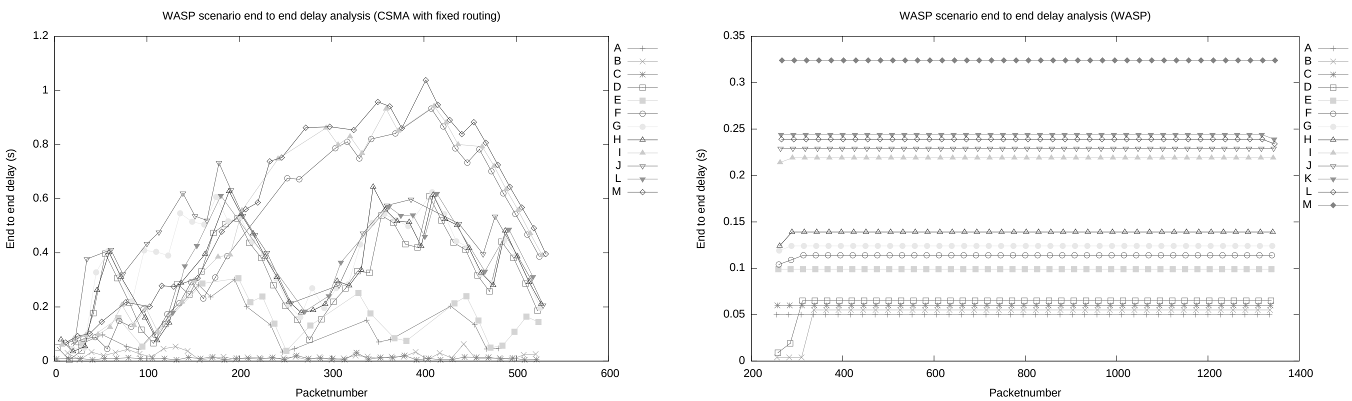
$$\overbrace{S}^{1}\ \underbrace{ABCD}_{2}\ \overbrace{.^{6}}^{3}\ \underbrace{ABBBDDDDD}_{4}\ \overbrace{X}^{5}\ \underbrace{1111111111111}_{6} \tag{14}$$

Figure 5 shows the end-to-end delay for this reference scenario when running 7.5 seconds. Figure 5(a) shows the results when CSMA is used combined with fixed (optimal) routing. The delay shows high variation and regularly exceeds 0.35 seconds. About 30% of the packets are dropped and node K does not even succeed at transmitting a single packet. Figure 5(b) shows the results when WASP handles medium access and routing and the slotsize is 5 ms. The delay is fixed and the levels are clearly visible. The maximum delay is 0.324072 seconds and no packets are dropped. The smaller delays at the left of this graph can be explained by the absense of traffic forwarding in the beginning of the simulation. The



(a) Network on the body. (b) Abstract view of the network.

Fig. 4. Simulated network.



(a) End to end delay with CSMA and fixed routing. (b) End to end delay with WASP.

Fig. 5. End to end delay comparison.

larger number of packets is due to the WASP-schemes that are broadcast each cycle.

In this example, node M has the highest delay. This node is situated on level 4 and thus roughly needs 3 WASP-cycles, according to (12). Calculating the delay explicitly, we see that the delay only amounts to 43 time slots. This can be verified by writing down the whole WASP-cycle, which is omitted due to space constrictions. Node M sends its data to node I at time slot 12 in the first WASP-cycle. In the second cycle, node I forwards it to node F. And in the third cycle, node F forwards it to node B who forwards it to the sink in the same cycle. The packet arrives at the sink in time slot 14. Adding this up, we get 45 time slots or a delay of 225 ms. The extra delay of 100 ms comes down to the time difference between the generation of the packet and the availability of a slot for node M. More generally, this extra delay is a value between 0 and the length of the WASP-cycle, hence in this example at most 105 ms.

As said in section IV-B, the nodes can turn their radio off in slots where they are not involved in the communication. The

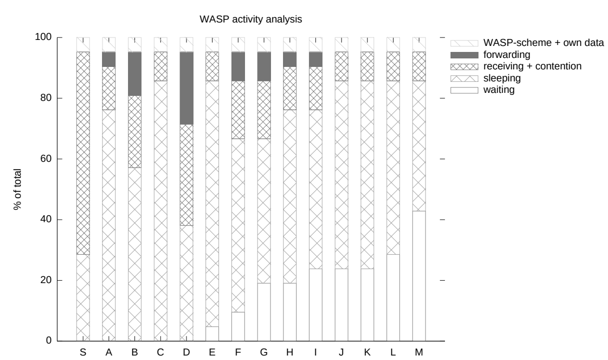


Fig. 6. Time usage in a node

time usage of the nodes is depicted in figure 6. In the example scheme the sink can turn its radio off in the silent period as it will not receive any data from its child nodes. Thus, the radio can be turned of 6 slots. Node A can also turn its radio off in its silent period, when it knows that its siblings are sending and when none of its children is due to send data. So, node A can sleep 16 slots.

VI. Future work

The WASP-protocol should be extended in order to even further ameliorate its performance. Different steps can be taken. For example, additional information can be advertised in the WASP-scheme, such as load, the quality of the link, etc. This can be used to choose a better parent and accordingly a better tree structure. Another method is data concatenation for lowering the number of transmissions. Doing this a node will put the data of several packets into one packet thus lowering the overhead per data bit sent.

The most important performance gain can be obtained by determing the optimal length of a time slot. This can be done based on the delay requirements for a specific application (12). We have not yet done this, but this should be feasible based on further analysis.

A major improvement would be the extension of WASP to handle non-static networks. Currently almost no mobility is supported which is of course a requirement for a realistic body area network.

VII. Conclusion

In this paper, we have presented WASP, a new cross layer protocol for wireless body area networks that both handles channel medium access and routing. For this purpose, a spanning tree is set up in a distributed manner and timeslots are used. Every node sends out a proprietary WASP-scheme to inform the nodes of the following level when they are allowed to send. These WASP-schemes are generated locally in each node. It is shown that the throughput can reach up to 94%, depending on the number of levels used. The end to end delay is shown to be fixed and related to the number of levels in the tree.

Acknowledgment

This research is partly funded by the Belgian Federal Science Policy Office through the IAP V/11 contract, by the Fund for Scientific Research - Flanders (F.W.O.-V., Belgium) and by The Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) through the contract No. 020152 and a PhD.grant for B. Latré.

References

[1] Emil Jovanov, Aleksandar Milenkovic, Chris Otto, Piet C. de Groen, “A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation,” Journal of NeuroEngineering and Rehabilitation, 2:6, March 1, 2005.

[2] S. Park and S. Jayaraman , “Enhancing the quality of life through wearable technology”, Engineering in Medicine and Biology Magazine, IEEE, Vol. 22, No. 3, May-June 2003, pp. 41–48

[3] L. Roelens, S. Van den Bulcke, W. Joseph, G. Vermeeren and L. Martens, “Path Loss Model for Wireless Narrowband Communication above Flat Phantom”, IEE Electronics Letters, Vol. 42, No. 1, January 2006, pp. 10–11

[4] Chris Otto, Aleksandar Milenkovic, Corey Sanders, Emil Jovanov, “System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring,” Journal of Mobile Multimedia, Vol. 1, No. 4, 2006, pp. 307-326

[5] T. Gao, D. Greenspan and M. Welsh, “Vital Signs Monitoring and Patient Tracking Over a Wireless Network”, Proceedings of the 27th IEEE EMBS Annual International Conference, September, 2005

[6] Benny Lo, Surapa Thiemjarus, Rachel King and Guang Zhong Yang, “Body Sensor Network A Wireless Sensor Platform for Pervasive Healthcare Monitoring”, Adjunct Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005), May 2005, pp.77–80

[7] T. Falck, J. Espina, J.-P. Ebert and D. Dietterle, “BASUMA - The sixth sense for chronically ill patients”, Workshop on Body Sensor Networks, Cambridge, MA USA, April 3-5, 2006

[8] Elisabetta Farella, Augusto Pieracci, Davide Brunelli, Luca Benini, Bruno Ricco, Andrea Acquaviva, “Design and Implementation of WiMoCA Node for a Body Area Wireless Sensor Network,” ICW 2005, 2005, pp. 342–347

[9] Zasowski, T., Althaus, F., Stäger, M., Wittneben, A. and Tröster, G., “UWB for noninvasive wireless body area networks: Channel measurements and results”, IEEE Conference on Ultra Wideband Systems and Technologies, UWBST 2003, Reston, Virginia, USA, Nov. 2003

[10] Fort, A., Desset, C., et al., “Ultra Wide-band Body Area Channel Model”, ICC 2005 - IEEE International Conference on Communications, May 2005, Seoul

[11] Latré, B., Vermeeren, G., Moerman, I, Martens, L. and Demeester, P., “Networking and Propagation Issues in Body Area Networks”, 11th Symposium on Communications and Vehicular Technology in the Benelux, SCVT 2004, Nov. 9, 2004, Ghent, Belgium

[12] Gupta, S. K. S., Lalwani, S., et al. “Towards a propagation model for wireless biomedical applications”, ICC 2003 - IEEE International Conference on Communications, May 2003, pp. 1993–1997

[13] W. Ye, J. Heidemann and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”, Proceedings of the IEEE INFOCOM 2002, Vol.3, New York, June 2002. , pp. 1567-1576

[14] A. El-Hoiydi and J.-D. Decotignie. “WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks”. In Proceedings of the International Symposium on Computers and Communications ISCC, pages 244-251, Alexandria, Egypt, June 2004

[15] W. Ye and John Heidemann, “Ultra-Low Duty Cycle MAC with Scheduled Channel Polling.” Technical Report ISI-TR-2005-604, USC/Information Sciences Institute, July, 2005.

[16] T. Penzel, B. Kemp, G. Klösch, A. Schlögl, J. Hasan, A. Värri, I. Korhonen, “Acquisition of biomedical signals databases”, IEEE Engineering in Medicine and Biology Magazine, Vol. 20, Nr. 3, 2001, pp. 25–32

[17] Michael Neufeld, Ashish Jain and Dirk Grunwald, “Nsclick: bridging network simulation and deployment”, Proc. 5th ACM intl. workshop on Modeling analysis and simulation of wireless and mobile systems, Atlanta, Georgia, USA, 2002, pp. 74–81