Traffic Aware Medium Access Control Protocol for Wireless Sensor Networks

Xi Zhang
xzh@mobnets.rwth-aachen.de

Junaid Ansari
jan@mobnets.rwth-aachen.de

Petri Mähönen
pma@mobnets.rwth-aachen.de

RWTH Aachen University, Department of Wireless Networks
Kackertstrasse 9, D-52072 Aachen, Germany

ABSTRACT

Wireless sensor networks are characterized by stringent battery resource. Owing to a wide range of applications, sensor networks are expected to support variable amounts of traffic loads. The resulting peak loads may drive the network into congestion, leading to high latency whereas low traffic loads cause energy wastage in idle-listening. This paper presents a preamble sampling based Medium Access Control (MAC) protocol for wireless sensor networks, which is highly optimized to traffic types and traffic loads in the network. Our MAC protocol exercises various optimization techniques on the preamble length based on traffic requirements. We present an analytical modeling of the protocol and derive the optimum performance parameters, which are validated using real implementation on a COTS sensor node platform. We also present in-depth simulation results for performance metrics such as power consumption, latency and delivery rates. These results adhere to our real hardware implementation results. A comparative analysis of our protocol against other state-of-the-art sensor network MAC protocols is presented in the paper to show the gains of our approach.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

MAC, Traffic awareness, Wireless sensor networks, Low power

1. INTRODUCTION

Wireless sensor networks (WSNs) in general have a limited lifetime due to the strict energy budget at sensor nodes. Energy consuming activities at nodes such as sensing, computation and communication dictate the lifetime of a network. Generally, communication has a significant share of power consumption cost which makes the operating lifetime of a network dependent directly on the media access procedures. WSNs have a wide range of applications with different characteristics and their traffic requirements may vary a lot from one application to another. Most of the monitoring applications require support for constant data communication with a small data size to be reported periodically while applications such as emergency or event detection have long idle durations but demand bursty data sending capability upon occurrence of a phenomenon of interest. Over the course of operation, different types of addressing (unicast, multicast, broadcast) for data packets lead to different amounts of control overhead. Appropriate mechanisms are needed for MAC protocols to efficiently meet the requirements of these varying traffic loads and types. WSN MAC protocols exercise duty cycling schemes, i.e. turning the radio on and off periodically, to save energy and to minimize idle listening duration. In order to coordinate the nodes exercising duty cycling for data communication, different schemes have been designed inflicting different amounts of coordination overhead. One of the challenges for WSN MAC protocol design is that there exists no universal MAC solution that is optimal for all diversified applications. It is usually the case that a particular solution works well with a specific application but not so efficiently with others.

There have been three main popular streams of MAC designs in sensor networks: contention free slot assignment based schemes, contention based common scheduling protocols and contention based preamble sampling type of methodologies [18]. Slot assignment schemes have poor scalability and are not suitable for networks with variable sizes and mobility. Many schemes try to re-use the wasted slots to achieve better channel utilization and to provide support for variable amount of traffic loads. However, due to the dynamic nature of the network (with new nodes joining the network and existing nodes disappearing because of the depleting battery and/or mobility), the overhead for the slot assignment/maintenance becomes significant. Protocols based on common schedule share part of the problem as the slot assignment based protocols, and the schedule maintenance overhead degrades network performance especially in cases of low traffic conditions. These two categories of MAC protocols spend a significant amount of overhead in maintaining the synchronicity in the network, e.g. activating the transmitter and the receiver at the same time. In preamble sampling based protocols, control overhead is directly related to the amount of traffic load in the network. Preamble sampling protocols allow nodes to be fully asynchronous. The coordination of nodes for data exchange is implicitly carried out by transmitting a preamble sequence before the data packet. Nodes polling the wireless channel are aware of upcoming data upon detecting a preamble sequence. The preamble sequence should be long enough so that all

MobiWac'09, October 26–27, 2009, Tenerife, Canary Islands, Spain.

the nodes in the vicinity of the transmitter, asynchronously polling the channel, are able to detect it. Although there is no control overhead when there is no traffic or the overhead is very little in case of low traffic conditions in the network, frequent transmissions lead to high costs of preamble transmission and reception. A long preamble not only leads to high power consumption, it also reduces the effective channel utilization since transmission of one packet requires a long channel occupancy duration. This in turn results in worse performance in terms of latency and packet delivery ratio. There are many protocols designed on the preamble sampling premise trying to shorten the length of the preamble to reduce power consumption and to improve latencies and packet delivery ratios. Many state of the art protocols perform well for some particular traffic conditions but are handicapped for others. By combining the existing preamble optimization techniques, one can have a highly adaptable preamble sampling MAC protocol. In this work, we present a MAC protocol which fully supports asynchronous networks, combines different preamble shortening techniques and includes additional features for efficient handling of a varying degree of traffic loads and traffic types.

2. RELATED WORK

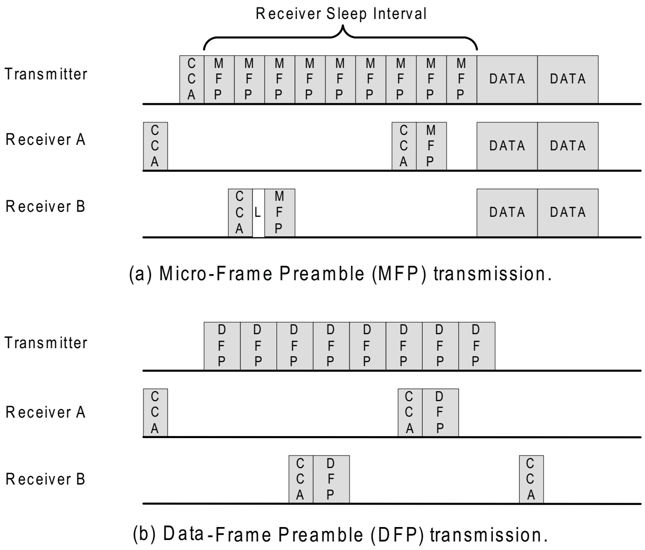
Since energy conservation in sensor network communication is one of the prior considerations, power efficient MAC protocol designs have been widely researched. Preamble sampling protocols allow nodes to operate asynchronously without adding extra control overhead for maintaining the sleep schedules. These properties are cogent to typical characteristics of sensor networks with low traffic and dynamic nature [9]. As a result, a number of preamble-sampling MAC solutions have been proposed. B-MAC [14] is the first preamble sampling protocol to be introduced. A transmitting node sends a preamble sequence of a length equal to the periodic channel sensing time of the receiving nodes. Receiving nodes poll the channel randomly and upon detecting the preamble, keep on listening to the preamble and receive the data packet following the preamble. Since the preamble sequence is typically long, a significant amount of energy is spent by both the addressed nodes and non-addressed nodes in receiving/overhearing the preamble. MFP-MAC [1] divides the long monolithic preamble into tiny frames containing the destination address and data payload information. A randomly waking up node is able to decide if the data is intended for it or not after receiving one preamble frame, and avoid listening to the rest of the uninterested preamble sequence. If it is the addressed node, it wakes up again for data packet transmission which follows the preamble. X-MAC [3] divides the monolithic preamble into frames, each containing the destination address. For unicast transmission, preamble strobing technique is used where, after transmitting a frame, the transmitter waits for an acknowledgement from the destination node. Subsequent preamble frames are sent if the preamble frame is not acknowledged within a certain waiting period. After receiving the acknowledgement, the transmitter immediately sends out the data. Preamble strobing saves energy for transmitters in the case of unicast transmission by avoiding transmission of extra preamble frames once a preamble frame has been acknowledged. B-MAC+ [2] sends the actual data packets repetitively to form the preamble sequence. The receiving nodes need to receive only the data packets without any preamble frames. B-MAC+ exercises preamble strobing for unicast transmission and thus both the transmitter and receiver save energy by avoiding sending and receiving unwanted preamble sequence. Having no information about the sleep schedules of the destination node(s), a significant amount of energy is spent for preamble transmission/reception despite the usage of the described preamble shortening techniques. WiseMAC [7] makes use of neighbourhood sleep schedule to optimize the preamble length for unicast transmission. Each node explicitly announces its wake-up schedule in the acknowledgement packet. Having the knowledge of wake-up information, a transmitter delays the data transmission till the time destination node is scheduled to wake up. WiseMAC also adjusts the length of the preamble based on the jitter offset developed over time between the transmitter and the receiver clocks. In a follow-up article [6], the authors of WiseMAC devised a scheme of repeating the data-packet for the broadcast case where the preamble lengths are long. Relying heavily on the neighbourhood sleep schedule, WiseMAC has shortcomings in a dynamic network where sleep schedules can be easily outdated or incomplete. Furthermore, repeating data packets in preamble introduces energy saving only when data size is small [11]. SCP-MAC [17] protocol uses synchronized channel polling and combines the features of scheduled based protocols with preamble sampling. This approach is suitable for networks operating in low duty cycles with static characteristics. After analyzing the different preamble sampling protocols mitigating the length of the preamble, we devised a protocol called traffic aware MAC protocol (TrawMAC) [10]. TrawMAC combines the features of transmitting the preamble as a train of tiny frames, strobing the preamble frames and optimizing the preamble based on sleep schedules of destination nodes. Additional features such as replacing a broadcast transmission with multiple unicast transmissions, data aggregation and multiple packets transmission with single channel reservation make it highly energy efficient and versatile to different traffic conditions.

3. PROTOCOL DESCRIPTION

TrawMAC is designed with the primary goal to optimize energy consumption by exploiting the shared traffic information across routing and MAC protocol layers with minimum compromises in latency and packet delivery ratio. It is based on the preamble sampling technique, where nodes sense the medium periodically according to the duty cycle specified. As described in Section 2, a significant amount of energy is spent in transmitting long preambles. Shortening of preamble length is thus a natural approach to achieve energy conservation in preamble sampling based MAC protocols. Depending on the traffic type, i.e. unicast, multicast and broadcast, TrawMAC exercises different mechanisms to shorten the preamble length. The design of TrawMAC protocol is much influenced by MFP-MAC, WiseMAC and X-MAC protocol which have been described in Section 2. It combines the advantages of these protocols especially when handling different traffic patterns and also includes other enhanced features as explained later in this section.

3.1 Preamble Structure

TrawMAC divides a monolithic preamble sequence into small preamble-frames, each containing a destination node address, a source node address, a message type and optional fields depending on the message type. The details of frame structure are described in Section 3.5. The construction of preambles varies based on the number and size of data packets to be transmitted and the destination address for transmission. There are mainly two types of preamble-frames: data-frame and micro-frame. When the size of the data packet is small, data-frame is used; otherwise, micro-frame is transmitted. The packet size threshold for switching between data-frame and micro-frame is selected to optimize energy consumption of the network. The actual threshold selection depends on the radio chip specifics such as data rate and power consumption in active states. The data-frame, as the name suggests, contains the data payload inside the preamble-frames. A train of data-frames



(a) Micro-Frame Preamble (MFP) transmission.

(b) Data-Frame Preamble (DFP) transmission.

Figure 1: Operational cycle in the case of broadcast transmission.

forms Data-Frame Preamble (DFP) which serves the purpose of transmitting both the preamble and the data. Micro-Frame Preamble (MFP), on the other hand, consists of micro-frames which contain only the control information. The size of the micro-frame is desired to be minimal in order to allow low control overhead. By using a preamble which consists of a number of small preamble-frames instead of one single long frame, energy consumption can be reduced at the receiver as well as at the transmitter. In the case of broadcasting data-frame preambles, a receiver goes to sleep immediately after receiving a single data-frame. In the case of broadcasting micro-frame preambles, all receivers go to sleep asynchronously after receiving a micro-frame, and switch back to receive mode together at the start of the data packet transmission as illustrated in Figure 1. This is achieved by including the start time of data transmission in the micro-frame structure. A preamble transmission timer is used to control the length of the preamble sequence in the case of broadcasting, which is equal to the sum of check interval (time between successive wake-ups of the receiver) and channel sensing time. It is assumed that all nodes in the network operate at the same duty cycle. One extra preamble-frame is transmitted after the expiry of the preamble transmission timer since there is a possibility that some receivers wake up in the middle of the transmission of the last frame. Adding one extra frame ensures that those nodes receive a preamble-frame as well. In the case of unicast transmission, additional optimizations on the preamble length are applied as discussed in Section 3.2 and Section 3.3.

3.2 Neighbourhood Information

TrawMAC maintains a sleep schedule of the neighbours at each node in a similar fashion as described in [6] in order to reduce the preamble length. With the knowledge of the schedule, nodes are implicitly synchronized to each other. The sleep schedule of the transmitting nodes is announced in preamble-frames. Unlike the approach in [6], it is a more effective way of disseminating the schedule information since the schedule is announced in both unicast and broadcast transmissions. Furthermore, the non-addressed nodes can update their neighbour schedule information when they overhear preamble-frames. Optimization on the preamble length based on the gathered sleep schedules of the neighbouring nodes is applied in unicast transmissions. The transmitting node looks up the sleep schedule information table of its neighbours and delays the transmission of the preamble frame till the wake-up schedule of the destination. By delaying the transmission of a packet, the preamble length is shortened by the delayed duration. If the receiver's schedule is not found in the table, the transmitter starts transmission immediately and sets the maximum length of preamble transmission to be the same as that for the case of broadcast transmission. It is important to note that nodes do not skip channel sensing at their scheduled wake-up time even if they are in the process of delaying a preamble transmission. It is necessary since there are other nodes in the neighbourhood potentially transmitting to this node base on its wake-up schedule.

The addressed nodes receive schedule information when receiving preamble-frames while the non-addressed nodes can receive this useful information from overhearing of frames as well. Although some radio transceivers such as Texas Instruments' CC2420 chip have an address recognition feature implemented in silicon and can flush the non-addressed packet out of the receiver buffer before being processed by the MAC layer, energy has already been spent at the radio in receiving the entire packet. Since the overheard information can be useful and no extra energy consumption is needed, all the received packets are to be sent up to the MAC layer. The non-addressed receiver, after receiving frame packets, stores the wake-up schedule of the transmitter and goes to sleep.

3.3 Preamble Strobing

Preamble strobing technique [3] is used in the case of unicasting data. After transmitting a preamble-frame (DFP/MFP), the transmitting node waits for an acknowledgement of the frame from the potential receiver. When an acknowledgement is received, the transmitter immediately stops sending subsequent frames, and in the case of MFP, sends the data packet(s) immediately afterwards. With the combination of neighbourhood wake-up schedule information, in the best case only one preamble-frame needs to be transmitted. However, due to the possible mobility of the nodes in the network and clock drifts over time, the estimation of the neighbour's scheduled wake-up time can be less reliable, and more than one preamble-frame might need to be transmitted as shown in Figure 2. Without the reception of an acknowledgement packet, the transmitter keeps on sending subsequent preamble-frames and waiting for the acknowledgements. In the worst case, the length of the preamble transmission becomes the same as that for the broadcast case. After sending the micro-frame acknowledgement, the receiving node immediately expects a data packet. Each data packet is acknowledged as well. A retransmission of data packets can be ensued upon the failure to receive acknowledgements after a certain timeout interval. Non-addressed nodes might also wake up during data packet transmission. In order to avoid further overhearing, the receiving node is only allowed to listen to the medium for a maximum of two frame duration in case channel activity is detected. If the node does not receive any complete packet within the time duration, it is forced to sleep.

3.4 Switching from Broadcast/Multicast to Unicast

The energy consumed by a receiver in unicast transmission is approximately the same as that in broadcast. However, the transmitter usually consumes a lot more energy when broadcasting data since the long preamble cannot be shortened using the methods for unicast traffic. Therefore, frequent broadcasts can deplete the energy on the transmitters. In a static network neighbourhood,

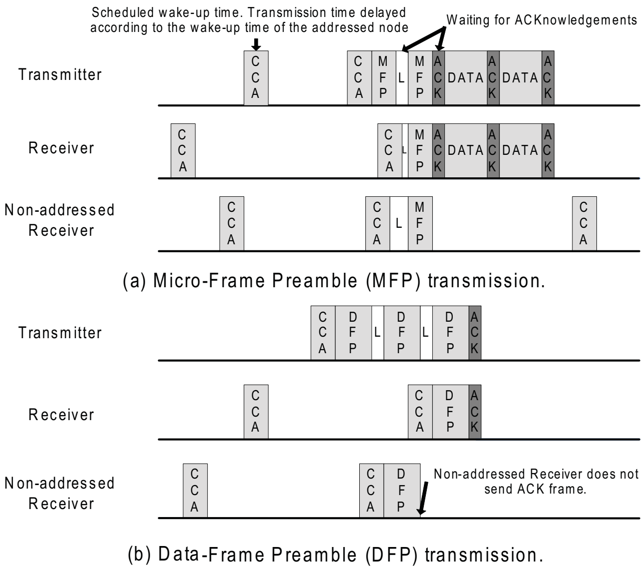


Figure 2: Operational cycle in the case of unicast transmission.

it is sometimes more energy efficient to do multiple unicasts instead of a single broadcast. If the node density is small and the network operates in low duty cycles, a long preamble is required for the broadcast transmission. If the neighbours are fixed (when no mobility is present in the network) and their wake-up schedules are known to the transmitter, a single broadcast may be replaced by multiple (equal to the number of neighbours) unicast transmissions. Although these transmissions are targeting specific receivers, the destination address field remains to be broadcast address and no acknowledgements are expected at the transmitter. Therefore, these transmissions can be more accurately termed as pseudo-unicast transmissions. The transmitter sends individual packet for each of its neighbouring nodes according to their wake-up schedules. If nodes wake up closely to each other, the scheme can be simplified by broadcasting with a shorter preamble. In the case of MFP transmission, instead of transmitting data packets immediately after the preamble-frame transmission for each destination node, the data packets are only transmitted once after the preamble-frames for all the nodes are sent. The transmission time of data packets is pre-calculated before the transmission of the first preamble-frame so that the timing information can be included in all of the preamble-frames. In this case, the operation can be appropriately described as a selective transmission of micro-frame preamble. Similar preamble shortening principles are applied to multicast transmissions as that to broadcast transmissions, e.g. a multicast transmission can be disintegrated into multiple unicasts if the schedule information of the destined nodes is known.

3.5 Frame Structure

There are three types of frames which are used in this MAC protocol, DFP, MFP and data packet. Each DFP contains the information of the destination address, the source address, the wake-up schedule of the source node, the message type (data-frame) and the data payload itself. When the receiver receives a data-frame, as indicated by the message type field, it can stop listening since there is no other relevant data following. A MFP contains the information of the destination address, the source address, the wake-up schedule of the source node, the message type (micro-frame) and in a broadcast transmission, the start time of the transmission of data packet(s). The start time is used by all the receivers to wake up together for the data reception. In a data packet, apart from the destination address, the source address, the message type (data-packet) and the data payload itself, a special field which indicates the number of data packets to be followed is included. Since TrawMAC is able to send multiple data packets accumulated at the sender with a single reservation as described in Section 3.6, the receiver needs to know how many data packets to expect so that it does not sleep early unnecessarily or listens for extra period of time. Specifying the exact number of packets gives the overhearing non-addressed nodes an idea of the duration of channel occupancy. Including the number in the data packets instead of the micro-frames and counting down on the run time reduces the chance of idle listening. Since packet loss is not uncommon in a wireless channel, the receiver would be less likely to wait for some packets which will never arrive by having an updated number of data packets to be expected.

3.6 Data Aggregation

When messages from upper layer arrive at the MAC layer, they are sometimes not processed immediately when the channel is unavailable, the transmitter is in a forced-to-sleep state due to prior knowledge of channel occupancy or in the process of receiving packets. Therefore, data packets might accumulate during the waiting period. Before the actual transmission, after the channel is determined to be free, TrawMAC looks through the data packets queued for transmission and sends the ones with the same destination address together in a frame train fashion. Here, the transmission of data packets is based on a first-come-first-served basis, i.e. if the destination address of the packets in the queue is `a,a,b,c,a`, only the first two with address `a` will be transmitted together.

4. ANALYTICAL EXPRESSION FOR OPTIMUM ENERGY CONSUMPTION

In this section, we analytically model TrawMAC protocol and derive an expression for the optimum sampling period giving minimum energy consumption for a given traffic load. Sampling period refers to the time interval where the node polls the channel once. It is equal to the sum of the check interval and one channel polling duration. The optimum sampling period can also be re-expressed as the optimum duty cycle. It is directly related to traffic load due to the trade-off between the large preamble overhead at low duty cycle and frequent channel polling activity at high duty cycle. For a given traffic load, when the duty cycle is lower than the optimum value, energy is wasted in preamble transmission activity; when the duty cycle is higher than the optimum value, energy is wasted in channel polling activity. Our model is simplified with several assumptions and serves as a proof of concept for the relationship between traffic load and duty cycle and their effect on node energy consumption. The channel is assumed to be ideal, i.e. all transmitted packets are successfully received by the receivers. Packet collisions, retransmissions and data aggregation is not modeled. The channel is not saturated, meaning all the packets generated at nodes are transmitted. Since the sensor network radios have very fast mode switching durations, the energy consumed in radio mode switchings, e.g. transmit to receive, sleep to receive, etc. is assumed to be negligible.

4.1 Models and Parameters

We consider a certain network size of n nodes. All nodes are within the vicinity of each other. Each node transmits r_{data} data packets per second. One data packet takes t_{data} seconds to be transmitted. If DFP is used, data packets are included in the preamble and $t_{\text{data}} = 0$. Each node spends power in the operations: carrier sensing before transmission, transmit, receive, channel polling at periodic wake-up and sleep state denoted by P_{cs}, P_{tx}, P_{rx}, P_{poll} and P_{sleep}, respectively. In the following, we list the terms used in our modeling:

$t_{\text{poll\_once}}$: single channel polling duration (s),
$t_{\text{samp\_period}}$: channel sampling period (s),
l_{frame}: length of one preamble-frame (bit),
$t_{\text{cs\_once}}$: single channel carrier sensing duration (s),
t_{b}: bit duration corresponding to radio data rate (s/bit),
t_{ack}: duration of one acknowledgement packet transmission (s).

4.2 Broadcast

The overall energy consumption of a node is the sum of energy spent in each operation and is given by,

$$E = E_{\text{poll}} + E_{\text{rx}} + E_{\text{cs}} + E_{\text{tx}} + E_{\text{sleep}}. \quad (1)$$

Since energy can be expressed as the product of power and time, the equation can be re-expressed as :

$$E = P_{\text{poll}}t_{\text{poll}} + P_{\text{rx}}t_{\text{rx}} + P_{\text{cs}}t_{\text{cs}} + P_{\text{tx}}t_{\text{tx}} + P_{\text{sleep}}t_{\text{sleep}}. \quad (2)$$

The values of the power terms are determined by the hardware specifications while the expressions of timings are given as follows:

$$t_{\text{poll}} = \frac{t_{\text{poll\_once}}}{t_{\text{samp\_period}}},$$

$$t_{\text{rx}} = (n-1)r_{\text{data}}(1.5l_{\text{frame}}t_{\text{b}}) + (n-1)r_{\text{data}}t_{\text{data}},$$

$$t_{\text{cs}} = r_{\text{data}}t_{\text{cs\_once}},$$

$$t_{\text{tx}} = r_{\text{data}}t_{\text{samp\_period}} + r_{\text{data}}t_{\text{data}},$$

$$t_{\text{sleep}} = 1 - t_{\text{poll}} - t_{\text{rx}} - t_{\text{cs}} - t_{\text{tx}}.$$

Receivers need to listen for 1.5 micro-frame duration to receive one complete micro-frame for the destination information and timings of the data packet on average. The entire preamble length to be transmitted before data packets equals to the sampling period. Since our target is to find the sampling period/duty cycle value which leads to the minimum energy consumption, we plug the above defined terms into Equation (2) and take the derivative w.r.t. $t_{\text{samp\_period}}$:

$$\frac{dE}{dt_{\text{samp\_period}}} = -\frac{P_{\text{poll\_once}}t_{\text{poll}}}{t_{\text{samp\_period}}^2} + r_{\text{data}}P_{\text{tx}} + \frac{P_{\text{sleep}}t_{\text{poll\_once}}}{t_{\text{samp\_period}}^2} - r_{\text{data}}P_{\text{sleep}}. \quad (3)$$

Putting $\frac{dE}{dt_{\text{samp\_period}}} = 0$ and simplifying the terms gives the optimum sampling period:

$$t_{\text{samp\_period}} = \sqrt{\frac{t_{\text{poll\_once}}(P_{\text{poll}} - P_{\text{sleep}})}{r_{\text{data}}(P_{\text{tx}} - P_{\text{sleep}})}}. \quad (4)$$

It may be noted that the $t_{\text{samp\_period}}$ expression is independent of n, the number of nodes contending for the same channel. It is due to the fact that in TrawMAC, the energy dissipated in receiving mode (which is dependent on n) is independent of the sampling period as we can see from the expression for t_{rx}. However, since our model is only applicable for non-congested network, n has an upper bound which is imposed by the equation:

$$nr_{\text{data}}(t_{\text{samp\_period}} + t_{\text{data}}) <= 1.$$

If n is greater than this boundary, the network is over-saturated and the optimum duty cycle derived from our model does not anymore offer the optimum performance.

4.3 Unicast

The total energy consumption at a node takes the same form as Equation (2) for the unicast case. Instead of receiving all packets transmitted in the neighbourhood, a particular node is the destination for kr_{data} packets out of the total $(n-1)r_{\text{data}}$ packets transmitted by its neighbours. Our model optimizes the sampling period w.r.t. traffic load. However, for a unicast transmission in TrawMAC, with the perfect knowledge of the neighbourhood timing schedules, only one preamble-frame is to be transmitted. This is the minimum energy consumption that one node can achieve which is independent of the sampling period. In the absence of any timing information, the preamble length depends on the offset between the sleep schedules of the transmitting and receiving nodes. In the worst case, the minimum preamble length is the same as the case of broadcast transmission as given by Equation (4). However, the expressions of the timing is different to the broadcast case and they are described as below:

$$t_{\text{poll}} = \frac{t_{\text{poll\_once}}}{t_{\text{samp\_period}}},$$

$$t_{\text{rx}} = (n-1)r_{\text{data}}(1.5l_{\text{frame}}t_{\text{b}}) + kr_{\text{data}}t_{\text{data}} + 2r_{\text{data}}t_{\text{ack}},$$

$$t_{\text{cs}} = r_{\text{data}}t_{\text{cs\_once}},$$

$$t_{\text{tx}} = r_{\text{data}}t_{\text{samp\_period}} + r_{\text{data}}t_{\text{data}} + 2kr_{\text{data}}t_{\text{ack}},$$

$$t_{\text{sleep}} = 1 - t_{\text{poll}} - t_{\text{rx}} - t_{\text{cs}} - t_{\text{tx}}.$$

A coefficient 2 is used for acknowledgement timing terms because a node needs to transmit/receive acknowledgement for both the preamble-frame and the data packet in the case of MFP transmission. If DFP is used, the coefficient equals to one. Similarly, the upper bound of n for MFP transmission is limited by the equation:

$$nr_{\text{data}}(t_{\text{samp\_period}} + t_{\text{data}} + 2t_{\text{ack}}) <= 1.$$

4.4 Comparison of Analytical Results to the Simulation Results

Simulations are conducted to verify the optimum sampling period deduced from our mathematical model. The set-up of TrawMAC simulation model can be found in Section 5. From Equation (4), we can see that data transmission rate r_{data} is a variable while other terms are fixed either due to radio properties or protocol designs. Different curves are plotted with various data transmission rates. The lowest power consumption per node should be achieved by using the optimum sampling period. We plotted the power consumption per node at different sampling periods and observed that the graphs are in concave up shape; thus, minimum points exist. The minimum points are shown to be coherent to the optimal value obtained from analytical formulation. The parameters that we used for the simulations are based on the CC2420 radio transceiver characteristics and are listed in Table 1.

We considered a network size of 3 nodes to present a non-congested network environment. The length of the preamble-frame, l_{frame}

Parameter	*Value in the CC2420 model*
Power in receive mode (P_{rx})	0.06204 W
Power in channel polling (P_{poll})	0.06204 W
Power in transmit mode (P_{tx})	0.05742 W
Power in sleep mode (P_{sleep})	0.0000000693 W
Time for one channel polling ($t_{\text{poll\_once}}$)	0.000128×8 s
Time to transmit/receive a bit (t_{b})	$1/250000$ s

Table 1: Parameter values used in the simulations.

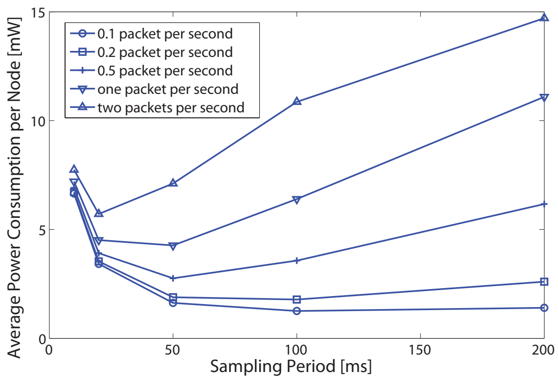


Figure 3: Power consumption at a sensor node with different sampling rates for different data traffic rates.

is 176 bits. Each simulation setup with different pairs of parameters is run for 20 minutes and each run is repeated three times. Figure 3 shows the average power consumption of a node at different sampling times for different data packet rates for broadcast transmissions. Using Equation (4), we can obtain the optimal sampling time ($t_{\text{samp\_period}}$) for different data packet rates. As a comparison for instance, it may be observed from Figure 3 that the minimum average power consumptions obtained for 2 packets per second and 0.5 packets per second are at around 20 ms and 50 ms, respectively. These values correspond to the results listed in Table 2 which we have obtained from the analytical expression using Equation (4). Thus our simulation results perfectly adhere to the analytical results.

Data packet rate [s^{-1}]	*Optimal sampling period [ms]*
0.1	105.2
0.2	74.4
0.5	47.0
1.0	33.3
2.0	23.5

Table 2: Analytical values of the optimal sampling periods at different data packet rates.

5. EVALUATION

We evaluate TrawMAC both in the OMNeT++ v4.0 [13] simulator and in an implementation in TinyOS 2.x [15] on TmoteSky [12] motes. We use simulations to explore the behaviour of TrawMAC in large scale networks with different routing protocols and compare the results to WiseMAC, which is a state-of-the-art preamble sampling protocol in terms of energy efficient performance. WiseMAC has the ability to adapt to traffic load and outperforms regular preamble sampling MAC metrics such as latency [5]. Although the simulator allows us to experiment with complicated networks, the performance of simulated protocol usually differs from real node implementations since the radio models are not calibrated exactly according to the hardware, and the real wireless medium can only be modeled with approximations. In order to validate the trend/behaviour exhibited by the simulated protocol, we carry out experiments on a small scale network of TmoteSky motes and compare the results to the most commonly used preamble sampling protocol B-MAC+.

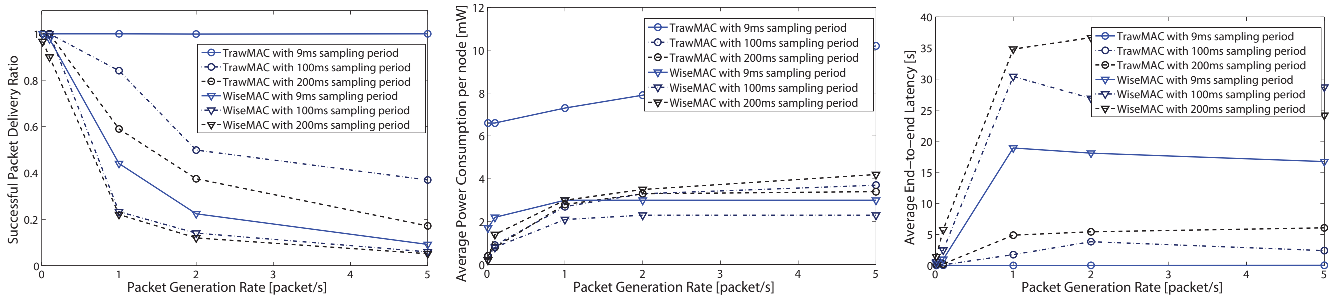
5.1 Simulation based Evaluation

In our simulation evaluation of TrawMAC, we focus on three main performance metrics namely packet delivery ratio, energy consumption and end-to-end latency. Both the packet delivery ratio and the end-to-end latency in this section refers to the values measured at the application layer. OMNeT++ v4.0 simulator is used and a radio model for CC2420 chip is readily available for the MAC protocols to be built upon. Three scenarios are set up with different network sizes and mobility to give an overview of the behaviour of TrawMAC in comparison with WiseMAC.

Comparisons between TrawMAC and WiseMAC are made when they have the same sampling period. Keeping the sampling period equivalent enables us to have a fair comparison in terms of latency and packet delivery ratio since the packet generation rate is the same for simulation setups of both protocols. Due to different algorithm used for Clear Channel Assessment, TrawMAC polls the channel for a longer period of time at each wake-up instance as compared to WiseMAC. This is because of the preamble strobing technique used by TrawMAC. Nodes need to wait for an acknowledgement between preamble-frame transmissions and the gap needs to be covered by the channel polling period to prevent false negative channel detection. A longer channel polling duration results in a higher effective duty cycle and thus leading to a higher power consumption of TrawMAC when both protocols are performing low-power-listening with the same sampling period.

In the first scenario, 10 nodes are placed within one hop to each other and transmitting to one receiving node (sink) with various packet generation rates. From Figure 4(b) it is shown that TrawMAC has a larger average power consumption than WiseMAC using the same sampling period. It is partially due to the longer channel polling time as described in the previous paragraph. Furthermore, the traffic in the network in this scenario is unidirectional. In a network where there is only one way traffic, i.e. from nodes to the sink in this case or vice versa, the nodes running TrawMAC do not benefit from the neighbourhood timing information they have gathered since they do not need to transmit to those nodes. Nodes running WiseMAC, on the other hand, get to transmit to the nodes which they have received acknowledgments from. Therefore, TrawMAC which spreads timing information in data packets suffers in this situation while WiseMAC which transmits the timing information in acknowledgment gains advantages in terms of power consumption. Although TrawMAC shows a poorer energy saving ability in this scenario, from Figure 4 we can see that it has clearly better performance in packet delivery ratio and latency. Increasing sampling period results in declination in both packet delivery ratio and latency performance of TrawMAC but power consumption is greatly reduced. The combinational effect gives TrawMAC with 200 ms sampling period a better performance in all three metrics than WiseMAC with 9 ms sampling period.

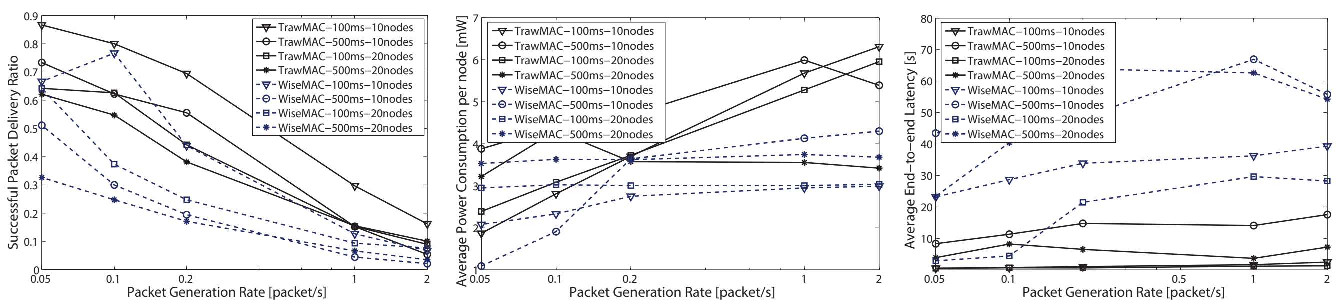
In the second scenario, simple routing protocol based on flooding algorithm [8] is placed on top of the MAC layer and both 10-node and 20-node static networks are simulated to form multi-hop networks. Nodes are randomly distributed but care has been taken to ensure that there exists at least one link from each node to the sink. The network density for 20-node network is higher than 10-node network. In the third scenario, routing protocol based on Dijkstra's algorithm [16] is used in a 20-node network with the same deployment as in the second scenario. After the performance results in the static network are gathered, mobility is introduced to the nodes. Nodes are moving at a speed with uniform distribution between 0 m/s and 0.8 m/s. The results from these two scenarios are compared to analyze how the two MAC protocols behave when used with different routing protocols.



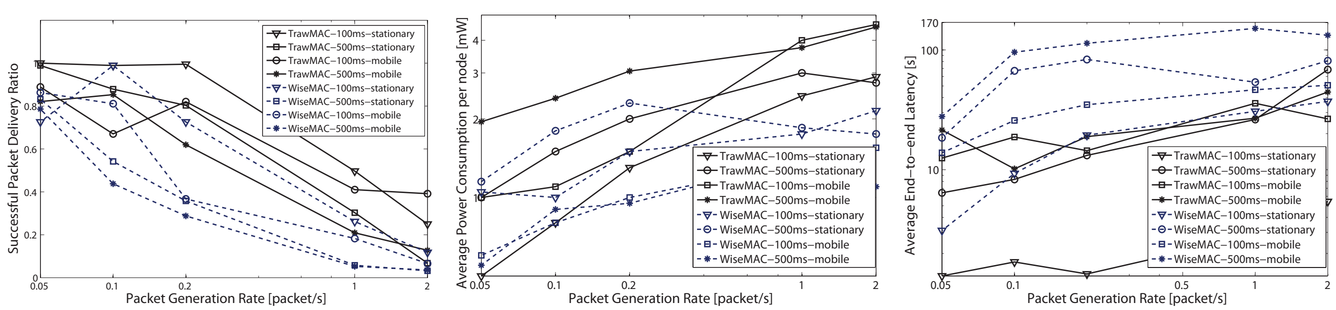
(a) Packet delivery ratio.
(b) Average per node power consumption.
(c) End-to-end network latency.

Figure 4: Performance of TrawMAC and WiseMAC in a one-hop 10-node network.

With a simple flooding routing protocol, nodes broadcast packets all the time. Therefore, preamble shortening techniques can seldom be applied for either of the MAC protocols. With no reduction in preamble length, the channel can be easily congested with traffic and therefore the packet delivery ratio declines sharply with the increase in packet generation rate. Routing protocol based on Dijkstra's algorithm uses unicast once the route has been established. Comparing Figure 5(a) to Figure 5(d), the effect of shortening preamble length is shown since the packet delivery ratio in the case of Dijkstra's is higher as compare to simple flooding. At packet generation rate greater than one packet per second, successful packet delivery ratio is approximately double of the simple flooding case. The packet delivery performance of TrawMAC is further improved with data aggregation. Mobility shows a negative effect on packet delivery due to the nature of the routing protocol. In the chosen protocol, a routing table is established based on the nodes within the neighbourhood before the start of the packet transmission. Although the routing table is updated periodically, it is incapable of handling very dynamic mobile networks where the neighbour list changes very often. To improve the overall MAC and routing performance, TrawMAC is currently being integrated with [4]. Some of the graphs in Figure 5 do not increase or decrease smoothly to be perfectly coherent with the general trend. All the simulation results shown are based on the average of three simulation runs. Due to the randomness of parameters such as mobility pattern, nodes deployment, a sample size of three might not be enough to give a highly precise result. However, we believe that the general trend of the protocol behaviour shown is adequate to prove the effectiveness of our MAC protocol. Figure 5(b) and Figure 5(e) show that packet generation rate has little impact on the power consumption for WiseMAC due to its inability to handle excessive traffic load in the network. TrawMAC consumes considerable more energy as packet generation rate increases due to its duty to deliver more packets while it consumes significantly less energy with lower traffic load and at 100 ms even outperforms WiseMAC. This proves the adaptability of our protocol to various traffic load.

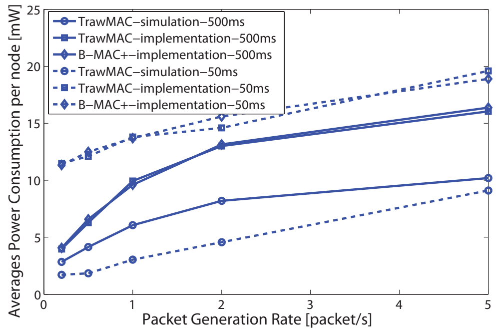


(a) Packet delivery ratio with flooding.
(b) Average power consumption with flooding.
(c) End-to-end latency with flooding.

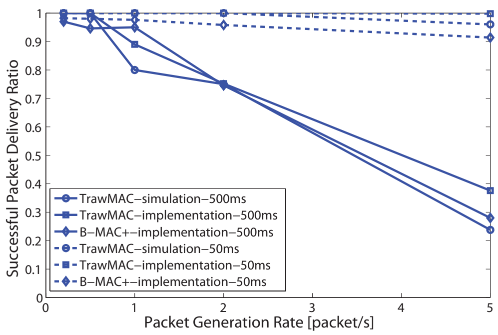


(d) Packet delivery ratio with Dijkstra's.
(e) Average power consumption with Dijkstra's.
(f) End-to-end latency with Dijkstra's.

Figure 5: Performance of TrawMAC and WiseMAC with routing protocols based on flooding and Dijkstra's algorithm.



(a) Average per node power consumption.



(b) Packet delivery ratio.

Figure 6: Comparison between implementation and simulation results of TrawMAC and B-MAC+ in a single-hop network.

In a preamble sampling protocol, per-hop latency is dependent on the sampling period. The longer the sampling period is, the longer is the per-hop latency. Both Figure 5(c) and Figure 5(f) show better latency performance for MAC protocols with shorter sampling periods. Since nodes are placed more densely together in the 20-node network than in the 10-node network, the average latency is lower since the average distance from nodes to the sink is smaller. Introduction of mobility degrades latency performance. In a mobile network, a node is incapable of finding its shortest path to the sink without the knowledge of the current positions of other nodes. Therefore, on average a packet propagates over more hops to reach the sink in a mobile network than in a static network.

5.2 Evaluation based on Hardware Implementation

In order to validate the trend and behaviour of TrawMAC observed from the simulation runs and to investigate possible hardware constraints and their impact on the protocol performance, we implemented TrawMAC on TmoteSky in TinyOS 2.x and carried out small scale experiments in two scenarios. The results are compared to the simulation results of TrawMAC as well as the implementation result of B-MAC+ on the same hardware platform. B-MAC+ is chosen due to its wide usage and code availability.

Figure 6 shows the protocol behaviour in terms of power consumption and application layer packet delivery ratio in a network size of 5 nodes within transmission range of each other and transmitting to the sink with different sampling periods at various packet generation rates. TrawMAC and B-MAC+ implementations exhibit similar behaviour in this scenario. It is due to the unidirectional traffic in the network as stated in Section 5.1 where the nodes do not benefit from the neighbourhood timing information they have gathered. Both TrawMAC and B-MAC+ are capable of preamble strobing upon receiving acknowledgement. Therefore, these two protocols show similar performance results.

The simulation results of TrawMAC show the same trend as its hardware implementation. Average power consumption increases with the increase of packet generation rate. Although the trend is the same, simulation results show a lower power consumption offset as compared to the real hardware implementation. It is due to different channel polling time for the two setups. The channel polling time in the simulation is set to be eight samples each consists of four symbol duration [14]. It is to cover the wait-for-acknowledgement gap between two successive frame transmissions in unicast as mentioned in Section 5.1. In the implementation case, due to the hardware constraints on TmoteSky and SPI bus speed limitation, reloading the transmission buffer each time before frame transmission takes a significant amount of time (approx. 8 ms). This reloading time is not modeled in the CC2420 radio model in OMNeT++. During this period of time, there is no transmission going on in the channel. Therefore, in order to make sure that the nodes detect channel activity upon waking up, the channel polling time needs to cover both the buffer reloading time and the time waiting for an acknowledgement reception which leads to a length ten times the length in simulation.

The difference in channel polling time not only introduces an offset between the simulation and implementation result at the same sampling rate, it also affects the general trend of power consumption at different sampling rates. As described in Section 4, the optimum sampling rate is dependent on the channel polling time. From Figure 6(a) we can see that in simulation, power consumption is lower with 50 ms sampling period while in implementation, power consumption is lower with 500 ms sampling period. It is because for simulations at 500 ms sampling period, the most significant energy spending is on long preamble transmissions in stead of channel polling since the nodes have a short channel polling time. For implementation at 50 ms sampling period, channel polling is the main source of energy consumption and as a result, an overall high energy consumption behaviour is observed. The difference in channel polling time does not affect the packet delivery ratio much since the sampling rate is set to be the same, the capacitance of handling packet generated at the application layer is the same.

In the second scenario setup, a multi-hop scenario with fixed addressing is created as show in Figure 7. All the nodes are placed within the vicinity of each other. Two hop latency is taken as the time interval between the instance when a packet is generated at node A and the time when the same packet, after hopping at node B, is received at node A. Similarly, four hop scenario uses node B and C as relay nodes. As the number of relay nodes increases, number of end-to-end hops increases. The application level latency of a preamble sampling based protocol typically increases linearly with the increase of number of hops end-to-end. With 1000 ms sampling period, the average latency induced by one hop should be half of the sampling period which is 500 ms since the wake-

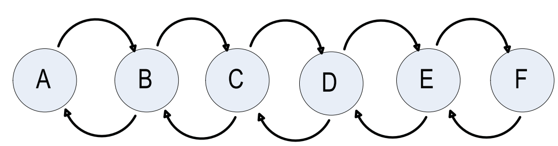
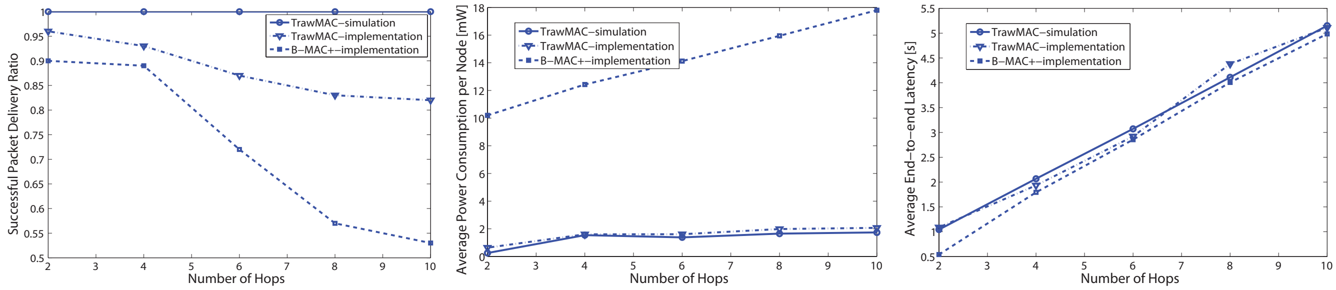


Figure 7: Experimental setup for multi-hop measurements.



(a) Packet delivery ratio. (b) Average per node power consumption. (c) End-to-end network latency.

Figure 8: Comparison between implementation and simulation results of TrawMAC and B-MAC+ with 1000 ms sampling period over a multi-hop network.

up time of the receiver is uniformly distributed over one sampling period. This observation is coherent to our result for the implementation as well as the simulation as shown in Figure 8(c). The results are obtained as an average over 100 samples. Figure 8(b) shows that TrawMAC has a much better power consumption performance in this case when compared to B-MAC+. It is due to the use of the neighbourhood information as the packets are routed in the network in a circular fashion periodically. The simulation and implementation results differ only slightly due to the long sampling period used in this experiment which diminishes the significance of energy used in channel polling activities. In Figure 8(a), TrawMAC simulation shows a perfect packet delivery ratio while TrawMAC implementation shows a slightly lower ratio which decreases with the increasing number of hops. Packet loss can be due to imperfect channel medium. TrawMAC reduces the channel occupancy by shortening the preamble which leads to a much better delivery performance as compared to B-MAC+. More number of hops means more nodes operating in the same neighbourhood at same frequency band which explains the rapid decline of packet delivery in B-MAC+ due to channel congestion.

6. CONCLUSION

In this paper, we have described in detail TrawMAC protocol, which combines the advantages of state-of-the-art preamble sampling MAC protocols and offsets their disadvantages. One of the key features of our MAC protocol is its ability to efficiently merge various preamble optimization procedures based on traffic conditions. Also, the scheme to replace a broadcast transmission with multiple unicast transmissions and data aggregation mechanisms give TrawMAC an edge over other MAC protocols. We have also derived an analytical expression for the optimum sampling period/duty cycle under given network parameters and have shown that our simulation results in OMNeT++ on CC2420 radio model perfectly adhere to it. Furthermore, we have carried out performance comparison of TrawMAC with other state-of-the-art MAC protocols both in simulations and actual sensor node implementation. The results from sensor node implementation are compared to simulation results to verified the correctness of the hardware characteristics modeling in our simulations in order for the simulation results to be realistic. We have conducted experiments with metrics such as power consumption, average end-to-end latencies and successful packet delivery ratios for various network sizes, different traffic conditions and different channel sampling periods. Both our simulations and hardware implementation on TmoteSky sensor node platform show a comparative edge of TrawMAC to the considered state-of-the-art MAC protocols. We are also planning a possible public release of the TrawMAC code, both the implementation in TinyOS 2.x and the simulation in OMNeT++ for interested parties.

Acknowledgment

We would like to thank financial support from European Union (project IST-034963-WASP), Deutsche Forschungsgemeinschaft through UMIC Research Centre and RWTH Aachen University. We would also like to thank Jean-Dominique Decotignie and Jérôme Rousselot for their help on the WiseMAC code in OMNeT++.

7. REFERENCES

[1] A. Bachir et al.. Micro-frame preamble MAC for multihop wireless sensor networks. In *Proc. Of ICC'06*, Istanbul, Turkey, 2006.

[2] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio. Increasing the efficiency of preamble sampling protocols for wireless sensor networks. In *Proc. of MCWC'06*, Amman, Jordan, 2006.

[3] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless networks. Technical report, University of Colorado at Boulder, May 2006.

[4] G. Cugola and M. Migliavacca. A context and content-based routing protocol for mobile sensor networks. In *EWSN*, pages 69–85, 2009.

[5] A. El-Hoiydi and J.-D. Decotignie. Wisemac, an ultra-low power MAC protocol for WiseNET wireless sensor network. In *MICS 2003 Scientific Conference*,2003.

[6] A. El-Hoiydi and J.-D. Decotignie. Wisemac: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *ISCC '04*, pages 244–251, 2004.

[7] A. El-Hoiydi et al. Poster abstract: WiseMAC, an ultra low power MAC protocol for the WiseNET wireless sensor network. In *SenSys03*, Los Angeles, CA, November 2003.

[8] N. Hutchinson et al. The flooding sink-a new approach to local area networking. *Comput. Netw. ISDN Syst.*, 11(1):1–14, 1986.

[9] I. Demirkol et al. MAC protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):, April 2006.

[10] J. Ansari et al.. Poster Abstract: Traffic Aware Medium Access Control Protocol for Wireless Sensor Networks. In *EuroSys*, 2008.

[11] J. Ansari, X. Zhang and P. Mähönen. Multi-radio Medium Access Control Protocol for Wireless Sensor Networks. In *Proc. of Workshop on Energy in Wireless Sensor Networks*, 2008.

[12] Moteiv Tmote Sky Datasheet. http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf. Last visited: 15.05.2009.

[13] OMNeT++, Discrete Event Simulation System. http://www.omnetpp.org. Last visited: 15.05.2009.

[14] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys*, November 2004.

[15] TinyOS, an Open Source OS for the Networked Sensor Regime. http://www.tinyos.net. Last visited: 15.05.2009.

[16] S. Varadarajan, N. Ramakrishnan, and M. Thirunavukkarasu. Reinforcing reachable routes. *Comput. Netw.*, 43(3):389–416, 2003.

[17] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *SenSys '06*, NY, USA, 2006.

[18] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. MAC Essentials for Wireless Sensor Networks. In *IEEE Communications Surveys and Tutorials*.