On the Adaptivity of today's Energy-Efficient MAC Protocols under varying Traffic Conditions

Philipp Hurni, Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern, 3012 Bern, Switzerland
{hurni, braun}@iam.unibe.ch

***Abstract*—Energy efficiency is a major concern in the design of Wireless Sensor Networks (WSNs) and their communication protocols. As the radio transceiver typically accounts for a major portion of a WSN node's power consumption, researchers have proposed Energy-Efficient Medium Access (E^2-MAC) protocols that switch the radio transceiver off for a major part of the time. Such protocols typically trade off energy-efficiency versus classical quality of service parameters (e.g. throughput, latency, reliability). Today's E^2-MAC protocols are able to deliver little amounts of data with a low energy footprint, but introduce severe restrictions with respect to throughput and latency. Regrettably, they yet fail to adapt to varying traffic loads and changing requirements of the imposed traffic load.**

This paper evaluates the energy-throughput and energy-latency tradeoff of today's most prominent E^2-MAC protocols for WSNs, and motivates the need for more flexible and traffic-adaptive E^2-MAC protocols. It proposes an intuitive definition for the ability of a protocol to adapt to varying traffic load at run-time, and introduces a tri-partite metric to measure and quantify this ability, further called *traffic-adaptivity*, taking into account the protocol energy-efficiency, throughput and latency. The paper concludes with a comparative analysis of the *traffic-adaptivity* of today's E^2-MAC protocols.

***Index Terms*—Wireless Sensor Networks, Energy Efficient Medium Access Control, Traffic Adaptivity, Traffic Awareness**

I. Introduction

Todays Energy-Efficient Medium Access (E^2-MAC) protocols generally reduce the power consumption at the cost of deteriorating quality of service, in particular by an increase of packet latency and a decrease of throughput and reliability. In the tradeoff between energy and quality of service, researchers have recently concentrated almost exclusively on the energy aspect. Many recent E^2-MAC protocols have thus been designed to handle very limited amounts of traffic at a very limited energy cost.

Strong restrictions with respect to throughput and latency may be tolerable in networks with low quality of service requirements. However, many event-based scenarios require reasonable quality of service during short periods of intense activity, and a high energy-efficiency and lower quality of service during long periods of inactivity. Such scenarios can be found e.g. in monitoring systems for the healthcare system, e.g. CodeBlue [1], or in Disaster-Aid-Systems [2], but also in the broad area of environmental monitoring. Varying, temporarily high traffic can further be expected to appear in the emerging field of multimedia sensor networks WMSNs [3]. Once an event has been triggered, e.g. a patient's pulse monitor registering anomalies in a hospital or geriatric clinic, the MAC protocol's primary objective should shift towards delivering good quality of service (high throughput, low delay) rather than saving energy. In such scenarios, today's existing E^2-MAC protocols do not provide reasonable flexibility, as most of them were designed under the assumption of very sparse low-rate traffic.

Without the ability to adapt to changing load, E^2-MAC protocols can not be put into practice for a broad range of applications. With applications for sensor networks growing in popularity, E^2-MAC protocols for WSNs need to become more flexible and adaptive with respect to changing load conditions, hence allow to use the radio truly in an *on-demand* manner. They should reduce the major sources of energy-waste, but still offer reasonable quality of service (high throughput, low delay) in case of increasing network activity. Similar to dynamic frequency/voltage scaling, where the CPU reacts to higher computation load with an increase of the frequency/voltage, a flexible and *traffic-adaptive* E^2-MAC protocol should react to changing traffic requirements by (de)allocation of battery resources by correspondingly tuning the radio transceiver. This paper introduces into recent work in the field of E^2-MAC protocols and traffic-adaptive extensions in Section II. Section IV describes the simulation environment and experiment setup that has been used to assess the protocol behavior under varying traffic conditions. By analyzing the protocols' energy-efficiency, maximum achievable throughput and latency in Section V, the paper illustrates advantages and disadvantages of today's E^2-MAC protocols, and motivates opportunities for improvements that are yet missing. In Section VI, we clearly define our understanding of *traffic-adaptivity* in an unambiguous manner: we introduce a tri-partite metric to quantify the *traffic-adaptivity* of an E^2-MAC protocol, and apply this metric to experimental results of a selection of today's E^2-MAC protocols. Section VII concludes the paper.

II. Related Work

In the past few years, a big number of E^2-MAC protocols have been proposed. The protocols differ in how nodes organize the access to the shared radio channel. [4] distinguishes three classes of organization *random access*, *slotted access* and *frame-based access*.

In *slotted access* protocols, nodes are synchronized to a common sleep/wake pattern. Nodes wake up at designated instants of time to exchange pending traffic. S-MAC [5] is the

9781-4244-3941-6/09/$25.00 ©2009 IEEE

most prominent protocol of this kind. S-MAC synchronizes the wake-up's of the nodes in so-called synchronization clusters. In each slot, nodes stay awake for an active window of fixed duration. S-MAC applies an RTS-CTS scheme for collision avoidance.

Random access protocols are generally based on contention mechanisms to avoid collisions, and do not rely on synchronized clocks, which makes these protocols rather simple and cheap with respect to the maintenance overhead. Prominent protocols of this class are B-MAC [6] and WiseMAC [7]. B-MAC lets the nodes alternate between long sleep intervals and periodic short wake-ups to poll the carrier for a preamble signal. The preamble is a busy tone that alerts the polling nodes to stay awake for the upcoming frame transmission. This scheme is further referred-to as low power listening (LPL) and has been adopted in many MAC studies. Unlike B-MAC, where a preamble always spans for the entire cycle duration, WiseMAC minimizes the length of the preambles to be transmitted and received by learning the schedule offsets of each neighbor.

The issue of E^2-MAC protocol adaptivity with respect to changing traffic load has yet been the topic of a few studies. The term *adaptivity* has been used as an ambiguous but popular buzzword in many WSN studies. Yet there is no clear notion of how to assess or measure traffic adaptivity. A comparative analysis of today's state-of-the art E^2-MAC protocols in the presence of varying traffic is definitely missing, a gap we intend to bridge with this paper. We define traffic adaptivity in the context of E^2-MAC protocols as *the ability of the protocol to dynamically and autonomously react to changing traffic requirements with (de)allocation of the respective resources needed to handle the imposed traffic with adequate quality of service at run-time.*

A couple of concepts has yet been applied to reach traffic-adaptive protocol behavior in today's literature on E^2-MAC protocols. In T-MAC [8], an increased traffic-adaptivity of the S-MAC [5] protocol is achieved by prolonging the duty cycles of the nodes when so-called activation events occur. An activation event may be the sensing of any communication in the neighborhood, the end of the own data transmission or acknowledgement, the overhearing of RTS or CTS control messages which may announce further packet exchanges. However, simulations show that the adaptivity of the protocol is still very limited. T-MAC shuts down the radio too aggressively and introduces a high delay for multi-hop transmissions. The performance gain of the traffic adaptivity enhancement further only pays off for non-uniform bursty traffic.

AMAC [9] is another MAC protocol claiming to provide traffic-awareness. AMAC dynamically adjusts the duty cycle, and thus can prevent packet drops to some extent while still saving energy. The scheme is shown to be superior to fixed-duty cycle E^2-MAC protocols.

X-MAC [10] is a recent E^2-MAC protocol based on asynchronous listen-intervals. For each packet, X-MAC sends out a strobe of preambles, in between which the receiver can signal reception-readiness with a so-called early ack. The authors derive a formula for optimal wake/sleep intervals given traffic at a constant rate and outline a mechanism to let X-MAC adapt the duty cycle and the sleep/wake interval to best accomodate the traffic load in the network.

III. IdealMAC as Reference Protocol

The concept of an ideal MAC protocol for wireless sensor networks (being called IdealMAC hereafter) plays a key role in our concept to measure and quantify protocol adaptivity. This concept has been used in [7] to show where the *lower bounds* of E^2-MAC protocol efficiency are. IdealMAC models the physical constraints of E^2-MAC protocols, such as the channel bandwidth, the delays and costs of the transceiver switches, as well as the transmission and reception costs. It however assumes that there is no information asymmetry between senders and receivers. Nodes always *know* when they need to switch to receive/transmit in order to handle data transmissions. IdealMAC is depicted in Figure 1 where a source node (Src) transmits a frame via an intermediate node (Int) to a destination (Dst).

In IdealMAC, nodes are always asleep in case of no traffic. At the very same instant a sender node receives a packet from the upper layer, such as *Src* in Figure 1, the receiver *Int* instantly switches its transceiver from the sleep state to the receive state. After frame reception and acknowledgement, *Int* forwards the frame to *Dest* in the same manner. IdealMAC therefore has the lowest-possible delay any E^2-MAC protocol can possibly have, and the highest possible energy-efficiency. It exhibits no overhead for periodic duty-cycling, periodic synchronization or any kind of control messages. As nodes immediately turn their transceivers to sleep after frame transmissions, they do not suffer from overhearing or idle listening. It is further assumed that nodes can always avoid collisions without introducing an RTS-CTS exchange. Nodes always *know* whether their targeted receivers are *ready* to receive messages or whether their neighbors are occupying the channel.

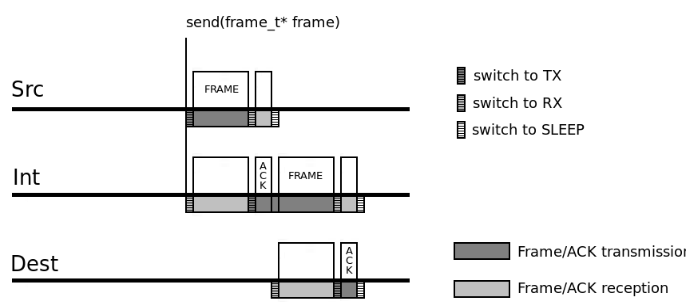


Fig. 1: IdealMAC reference protocol

IV. Simulating Varying Traffic Conditions

We implemented S-MAC, T-MAC, B-MAC, WiseMAC, X-MAC and the reference protocols IdealMAC and simple energy-unconstrained CSMA in the OMNeT++ Network Simulator [11] using the Mobility Framework (MF) [12], which supports simulations of wireless ad hoc and mobile networks on top of OMNeT++.

A. Wireless Propagation Model

In the recent years, network simulation tools and simulation studies in general have been heavily criticised, mainly because

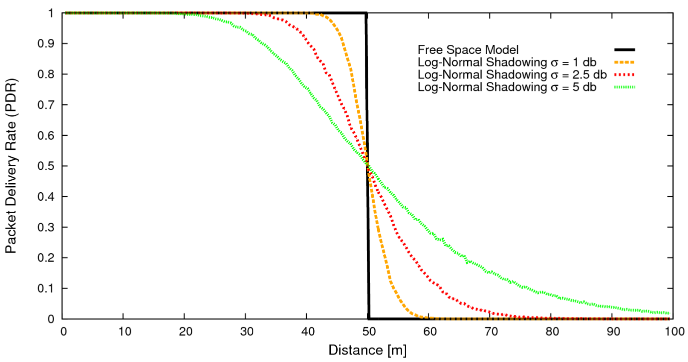


Fig. 2: Free Space vs. Log Normal Shadowing

of oversimplified simulation model assumptions and inadequate parameter settings [13], [14]. Unit-Disk-Graph (also referred-to as *Flat-Earth*) based simulation models, as well as the deterministic Free Space Model have been shown to even produce misleading results [15]. In order to realistically reflect the characteristics of wireless propagation (high packet error rate, shadowing and fading-effects), we applied the Log-Normal Shadowing Model [16], which has been recently introduced into OMNeT++ [17]. This channel model allows for a more realistic simulation of wireless channel properties. It models small-scale shadowing and fading effects - which are typical wireless phenomena - for each frame transmission by adding a random pertubation factor to the reception power. The pertubation factor follows a log-normal distribution with a user-selectable deviation σ. Figure 2 depicts the packet delivery rate (PDR) (y) versus the distance (x) between a sender and a receiver, applying the default OMNeT++ Free Space Model and comparing it to the Log-Normal Shadowing Model with different values of σ. With the Free Space Model, the received power is a simple deterministic function of the distance. When having only one sender at a time, the reception probability immediately drops from 100% to 0% when the distance between the sender and the receiver exceeds 50m (with the given transmission power and SNR threshold settings). Using the Log-Normal Model, the PDR decreases gradually with the distance, exhibiting different slopes with different values of the deviation σ of the random pertubation factor.

B. Energy Model

We modelled the power consumption of the sensor nodes with a state transition model with respect to the time spent in three

transceiver parameters CC1020 [19]:	
supply voltage U	3 V
transmit current I_{tx}	21.9 mA
recv current I_{rx}	17.6 mA
sleep current I_{sleep}	1 μA
transmission rate R	115'200 bps
simulation model parameters:	
path loss coefficient α	3.5
lognormal deviation σ	2.5 db
carrier frequency	868 MHz
transmitter power	0.1 mW
SNR threshold	4 dB
sensitivity	-100.67 dBm
carrier sense sensitivity	-112 dBm
communication range	50 m
carrier sensing range	100 m
simulation runs	100
simulated time for each run	1000s
ARQ max retries	3
frame header size	14 bytes
payload	50 bytes

TABLE I: Simulation and Experiment Parameters

operation modes sleep, receive and transmit, weighted with the respective energy costs. The same methodology is applied in [18], where the power consumption of a IEEE 802.11 wireless device is modelled with the same three states. Experimental results in [18] confirm the adequateness of the linear state transition model.

Table I lists current, voltage and transmission rate of the CC1020 [19], a byte-level radio transceiver in the 804-940 MHz ISM frequency band. The CC1020 is used by the MSB430 sensor nodes platform [20], which we intend to use in the near future for prototyping maximally traffic-adaptive E^2-MAC protocols on real sensor hardware. Table I further lists the parameters of the wireless channel simulation model of the OMNeT++ Network Simulator [11], as well as experiment-specific settings.

C. Protocol Simulation Models

Table II displays the main simulation parameters of the simulated E^2-MAC protocol models. As the protocol behavior often heavily depends on the choice of the essential protocol parameters (e.g. basic wake interval, duty cycle, etc.), we studied the protocols with different *configurations* of those parameters, by varying the parameters over a wide range.

For the *slotted* protocols S-MAC and T-MAC, we assume

CSMA		B-MAC	
Contention Window CW	10 ms	Basic Interval	[25, 50, 100, 200, 500] ms
S-MAC		Duty Cycle	[8, 4, 2, 1, 0.4, 0.2] %
Listen Interval	[100, 200, 300, 500, 1000, 2000] ms	WiseMAC	
Duty Cycle	10%, 20%	Basic Interval	[25, 50, 100, 200, 500, 1000] ms
T-MAC		Duty Cycle	[8, 4, 2, 1, 0.4, 0.2] %
Frame Length	[50, 100, 200, 300, 500, 1000] ms	Medium Reservation Int.	uniform (0,10) $\times$ t_{rx-tx}
Contention Window CW	5 ms	X-MAC	
SYNC size D_{SYNC}	14 bytes	Max Interval	[100, 200, 500] ms
RTS size D_{RTS}	14 bytes	Min Interval	10 ms
CTS size D_{CTS}	10 bytes	Early-ACK size D_{EACK}	10 bytes
Timeout	$1.5 \times (CW + D_{RTS}/R + D_{CTS}/R)$	Inter-Strobe-Interval	D_{EACK}/R
SYNC period	10 s	Listen Interval	$D_{EACK}/R + t_{rx-tx} + t_{tx-rx}$

TABLE II: E^2-MAC Protocol Parameters

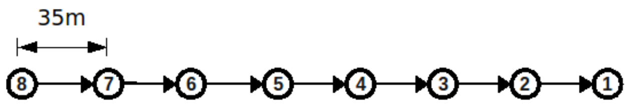


Fig. 3: Chain Scenario with 8 Nodes

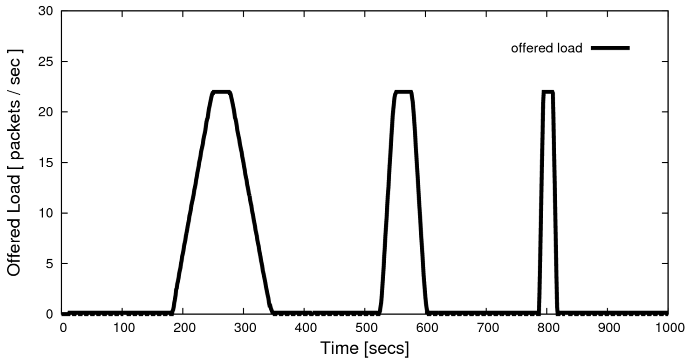


Fig. 4: Offered Load (Packets/s)

that the nodes' wake-up intervals are synchronized from the beginning of the experiment (the same assumption is found in many MAC studies, e.g. in [7]). With X-MAC, we integrated an adaptation algorithm that adapts the wake/sleep intervals according to the rate of incoming packets (as specified in [10]), but remains in-between [Max Interval, Min Interval]. For WiseMAC, we assume that nodes are able to sense transmissions in the channel from stations within their *carrier sensing range* ($\sim 2\times$*max transmission range*) to implement a cheap collision avoidance. Such a mechanism can be accomplished by most of today's radio transceivers by observing the onboard RSSI value and setting appropriate thresholds.

In order to allow for a fair comparison, we implemented a packet burst transfer mode for each simulated E^2-MAC protocol, such that nodes can transmit queued packet trains in a burst. Nodes can signalize that they have pending packets to the receiver and continue transmitting packets in a burst, receiving an acknowledgmenet for each frame.

D. Simulation Experiment

We simulated a chain consisting of 8 nodes with the source node (8) generating load, which is then forwarded hop-by-hop towards the sink node (1), similarly as done in the studies on S-MAC [5] and B-MAC [6].

Almost any study on E^2-MAC protocols applies constant rate traffic during one simulation run. In contrast to this, we varied the offered traffic from low rates to high rates and back during each simulation run, as our major point of investigation in this study is the protocol adaptivity at run-time. Figure 4 displays the offered load generated at the application layer of node (8). Traffic is very low (0.1 packets/s) for most of the time, but there are load peaks where the packet rate is linearly increased up to 22 packets/s and then again linearly decreased to the low level. We chose 22 packets/s as the load maximum as this had proved to be the maximum throughput that CSMA could handle without major packet loss. When increasing the rate above this rate, throughput stalls and all additional packets are either dropped due to buffer overflows or are lost due to collisions.

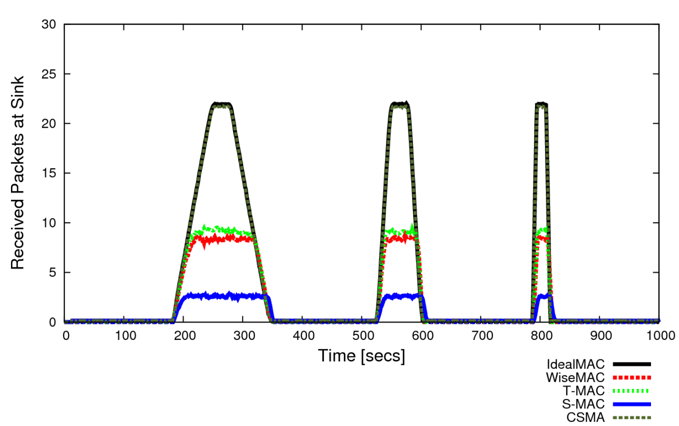
The load is increased linearly, with three different slopes for the load increase. In the first peak, the load is increased from 0.1 packets/s to 22 packets/s within 60s, and decreased over the same time period. In the second peak, load is increased and decreased faster (20s), and in the third peak almost instantaneously (5s). Using these different slopes for the variation in the offered load, we study how the existing E^2-MAC protocols react to slowly and/or rapidly varying traffic conditions.

E. Received Packets and Network Power Consumption

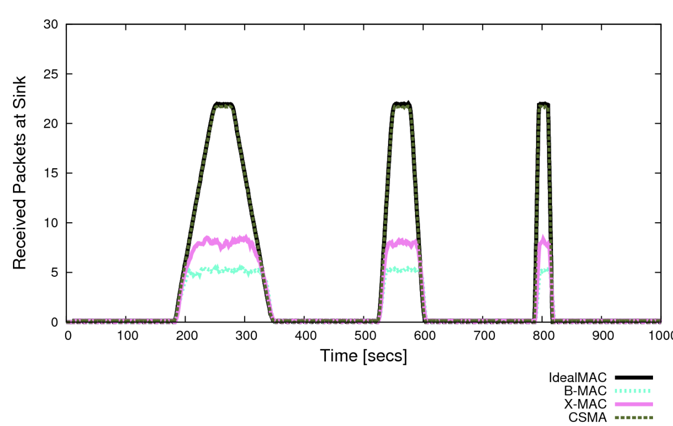
Figure 5 displays the rate of received packets at the sink (1) vs. simulation time. All subsequent curves are averaged from 100 simulation runs for each protocol. As one can clearly see comparing the received packets in Figure 5 with the offered load in Figure 4, IdealMAC manages to handle all packets from source to the sink. CSMA only suffers minor packet loss at the load peaks.

The S-MAC protocol with its static fixed-duration listen interval only manages to handle up to ~ 3 packets/s. T-MAC, B-MAC, WiseMAC and X-MAC reach higher throughput rates at the load peaks, but their throughput stalls at maximum 8 packets/s, which corresponds to $35-40\%$ of that of CSMA.

Figure 5 illustrates the potential for optimization in the design space of today's E^2-MAC protocols. The existing protocols are not yet sufficiently adaptive with respect to varying load conditions, as they do not manage to adapt their behavior with respect to the load conditions, and to allocate the channel

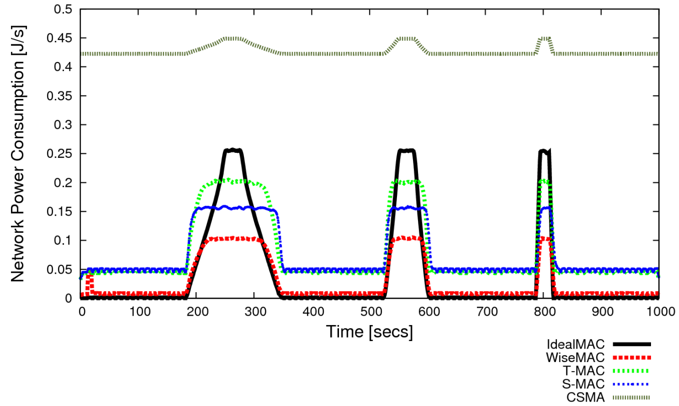


(a) S-MAC, T-MAC and WiseMAC vs. IdealMAC and CSMA

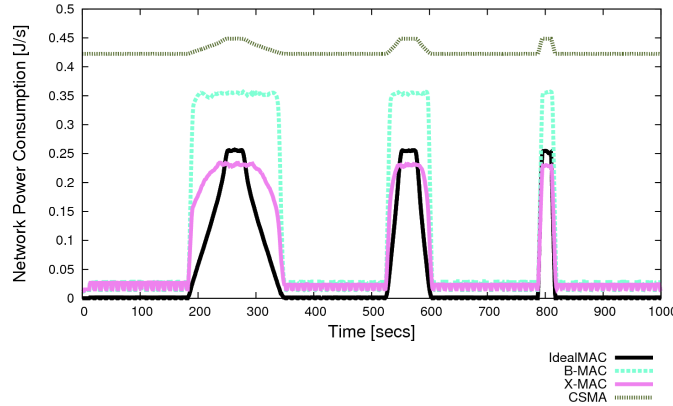


(b) B-MAC, X-MAC vs. IdealMAC and CSMA

Fig. 5: Received Packets at Sink (Packets/s)



(a) S-MAC, T-MAC and WiseMAC vs. IdealMAC and CSMA



(b) B-MAC, X-MAC vs. IdealMAC and CSMA

Fig. 6: Power Consumption (J/s)

and energy resources truly in an *on-demand* manner. Although there is sufficient channel capacity, the existing protocols do not manage to respond to the increased load with allocation of the respective resources needed to handle this imposed load. The radio transceiver is still shut down too aggressively and/or used inefficiently by the E^2-MAC protocols in case of load peaks.

Figure 6 depicts the aggregated power consumption of all 8 sensor nodes' radio interfaces versus simulation time. One can clearly see the big gap between the E^2-MAC protocols and energy-unconstrained CSMA. With low traffic, CSMA wastes a lot of energy on idle listening. The load peaks are hardly visible at all, as the transceiver does not use much more power when sending and receiving data, compared to idle listening. In contrast to CSMA, the IdealMAC reference protocol illustrates the ideal behavior of an E^2-MAC protocol, allocating as much energy as needed to handle the imposed load, and immediately deallocating it with decreasing load.

Figure 6 further depicts that the simulated E^2-MAC protocols use much less energy with low traffic rates, and already exhibit an increasing power consumption at the load peaks. The protocols however differ in the level of power consumption during the low traffic phases and thus their energy efficiency in the sparse-traffic case (c.f. S-MAC vs. WiseMAC), but also in the reaction to the linearly increasing and decreasing load level. The *slotted* E^2-MAC protocols S-MAC and T-MAC exhibit a much higher power consumption at r=0.1 packets/s, due to their overhead to keep their sleep/wake intervals synchronized. With WiseMAC or B-MAC, nodes do not maintain common sleep/wake-schedules. Nodes only wake up briefly to check for the presence of a preamble signal, and do not periodically distribute common schedule information. Thus, these protocols exhibit a much lower power consumption with low traffic rates. WiseMAC basing on preamble sampling and renouncing on costly synchronization schemes, has a very low per-packet overhead, as it minimizes preambles by learning the neighboring nodes' schedules. The protocol thus manages to remain close to the ideal curve, but its power consumption and its throughput stalls at $\sim 35\%$ of that of CSMA. Interestingly, all protocols react to the linearly increasing and decreasing load level in a symmetric manner. The increase of power consumption during the load increase is symmetric to the decrease in power consumption during the load decrease, which is a desireable property.

V. The Energy-Throughput and Energy-Latency Tradeoffs

Today's E^2-MAC protocols typically trade off quality of service versus higher energy-efficiency. Generally, they introduce higher delays and restrain the maximum achievable throughput. In this section, we examine these tradeoffs with the simulated E^2-MAC protocols. By running each protocol with different parameter settings, we thoroughly investigated the behavior of each of the simulated E^2-MAC protocol mechanisms, and not just the behavior of one particular parameter choice. We refer to one parameter tuple for a protocol as a *configuration* hereafter, e.g. one configuration for WiseMAC would be [Basic Interval=200ms, Wake Ratio=1% (2ms)].

Figure 7 and 8 illustrate the energy-throughput and energy-latency tradeoffs of the simulated E^2-MAC protocols. Each dot represents the results of one particular protocol *configuration* in the simulation experiment outlined in Section IV-D. In Figure 7, the tradeoff between maximum achieved throughput and energy-efficiency of the simulated E^2-MAC protocols becomes very well visible. CSMA being energy-unconstrained has a very high maximum throughput. However, with CSMA not turning off the transceiver during the low-traffic phases, its energy-efficiency remains very low. The protocol efficiency is measured in in kBit/J, hence calculating how many *useful* (payload) bits have been transmitted from source to sink for each consumed Joule. A similar concept has been proposed as the energy-per-useful-bit (EPUB) metric in [21] - we however use the reciprocal coefficient in order to obtain a metric where *more is better*.

The IdealMAC protocol, in which a receiver node always *knows* when to switch the transceiver to the receive mode to receive packets, has both a high throughput and a very high energy-efficiency. IdealMAC illustrates where the theoretic lower and upper bounds of the E^2-MAC protocol problems are - it is not possible to reach a higher throughput nor a higher efficiency than IdealMAC. No E^2-MAC protocol will ever get beyond the rectangle that is spanned by IdealMAC in Figure 7.

With the examined protocols, the different choices of the frame-length and basic interval parameter values result in dots

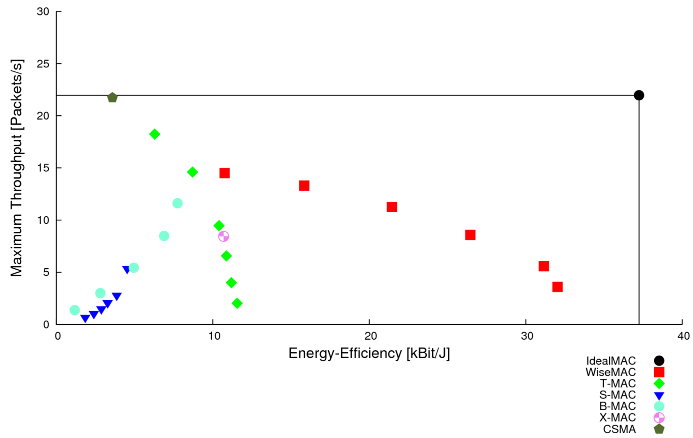


Fig. 7: Throughput vs. Energy-Efficiency

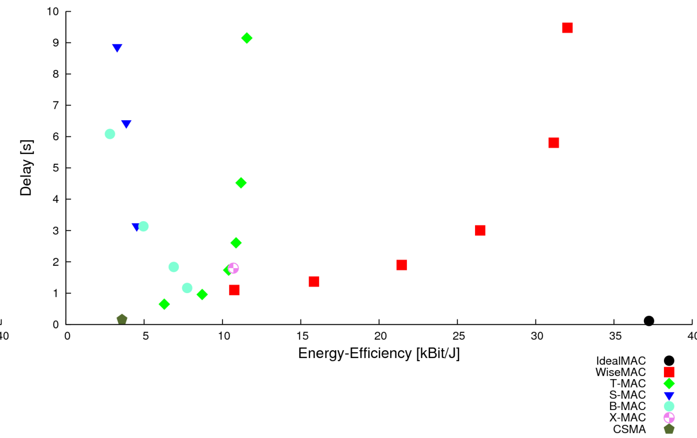


Fig. 8: Delay vs. Energy-Efficiency

forming indifference curves, visualizing how much maximum achievable throughput the existing protocols need to give up to reach a higher energy-efficiency, when moving from the top leftmost dot towards the lower rightmost dot, and vice-vera. E.g. if WiseMAC is being operated with a very long interval between two wake-up's, the protocol almost reaches the energy-efficiency of IdealMAC, but then only achieves a very limited throughput. On the other hand, T-MAC can be tuned to reach almost the same throughput as CSMA, but at the cost of a decreasing energy-efficiency.

The X-MAC protocol with its wake-cycle adaptation algorithm reaches a fair throughput and tolerable delay at a reasonable efficiency, but its performance lags behind that of WiseMAC. The main reason for this is the high per-packet overhead of the preamble strobes. One crucial advantage of X-MAC's strobed preamble mechanism is the possibility to let nodes adapt their wake-sleep cycles to the traffic rate. Nodes with short wake-sleep cycles will respond earlier with an early ACK to the strobed preambles than nodes with a long wake-sleep cycle. Hence the protocol offers self-configuration and adaptation capabilities, while with the other protocols, e.g. WiseMAC and T-MAC, the interval between two wake-up's remains constant and does not adapt to the traffic rate.

Figure 8 similarly depicts the tradeoff between average packet delay and energy-efficiency. One can observe that CSMA exhibits a very low average delay at the cost of a very low energy-efficiency. IdealMAC reaches both, a very low delay at a very high energy-efficiency. IdealMAC again illustrates the lower bounds of the E^2-MAC protocol problem - while it is not possible to reach a higher throughput than IdealMAC, it is neither possible to reach a lower average delay. One can clearly see the energy-latency tradeoff with the different *configurations* of T-MAC and WiseMAC. When increasing the energy-efficiency of the protocol configurations by increasing the interval between two wake-ups, the delay accordingly increases, too. While T-MAC can achieve a lower delay, WiseMAC exhibits a higher energy-efficency.

VI. Measuring Traffic Adaptivity

We investigated a means to *measure* and *quantify* the property of *traffic-adaptivity* of an E^2-MAC protocol under varying traffic conditions. An optimal and maximally traffic-adaptive E^2-MAC protocol should allow to use the radio-transceiver truly in an on-demand manner, using it as much as necessary to transmit and receive packets whenever traffic needs to be handled, and turning it off when nothing has to be sent or received. We find that the question how well a protocol is able to adapt to varying traffic conditions can be rephrased by the question how good the quality of service parameters throughput and delay are in a scenario of heavily varying traffic, and how energy-efficient the protocol remains under such conditions. We thus developed a *tri-partite* metric that quantifies the ability of an E^2-MAC protocol to adapt to varying traffic conditions, based on a comparison with the IdealMAC reference protocol. The metric incorporates the energy-efficiency, the maximum achievable throughput, as well as the average delay, hence we refer to it as tri-partite.

First, we briefly remind the mathematical properties a metric function $d(x, y)$ needs to fulfill:

A metric d is a mapping $d : X \times X \to \mathbb{R}$ on any set X, with $\mathbb{R}$ being the set of real numbers. For all x, y, z in X, this function is required to satisfy the following conditions:

- $\mathrm{d}(x, y) \geq 0$ (non-negativity)
- $\mathrm{d}(x, y) = 0$ if and only if $x = y$ (identity of indiscernibles)
- $\mathrm{d}(x, y) = \mathrm{d}(y, x)$ (symmetry)
- $\mathrm{d}(x, z) \leq \mathrm{d}(x, y) + \mathrm{d}(y, z)$ (triangle inequality)

Our proposed metric for the *traffic-adaptivity* of an E^2-MAC protocol under varying traffic measures the minimal *distance* between the different configurations of the protocol and the IdealMAC reference protocol. This distance is measured in the vector space spanned by the energy-efficiency (x), the maximum throughput (y) and the delay (z) measured in the above experiment of varying traffic. Hence, we actually measure how much *worse* the protocol performs in comparison with IdealMAC. We represent the results of each configuration P_i of the simulated E^2-MAC protocol P as a tuple $(x_{p_i}, y_{p_i}, z_{p_i}) \in X \times Y \times Z$ where X is the energy-efficiency (measured in kBit/Joules), Y the maximum achievable throughput (packets/sec) and Z the average measured delay that the protocol exhibited in the varying load scenario. We further refer to $Id = (x_{Id}, y_{Id}, z_{Id})$ as the tuple representing the results of the IdealMAC protocol hereafter.

The Euclidean distance d between IdealMAC and any configuration P_i of the simulated E^2-MAC protocol denoted as:

$$d(P_i, Id) = \sqrt{(x_{p_i} - x_{Id})^2 + (y_{p_i} - y_{Id})^2 + (z_{p_i} - z_{Id})^2}$$

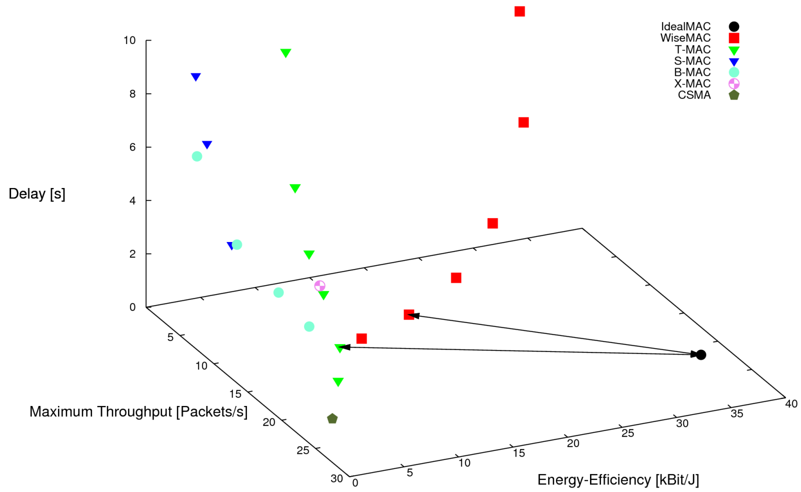


Fig. 9: Energy-Efficiency (x) vs. Maximum Throughput (y) vs. Delay (z)

measures how much *worse* the configuration P_i behaves under the examined experiment conditions compared with IdealMAC. Figure 9 illustrates this difference between configurations of WiseMAC and T-MAC and the reference protocol IdealMAC (black vectors).

Applying the Euclidean metric to the measured values yields distances that are dependant of the scale of the axis. Any metric assessing the adaptivity of E^2-MAC protocol should however be independent of the axis scale, and take the energy-efficiency, the maximum achievable throughput as well as the latency into account at equal ratios. We thus normalize the x, y and z-axis to take values in between the interval $[0, 1]$ and obtain the normalized distance d_{norm}:

$$d_{norm}(P_i, Id) = \sqrt{(\frac{x_{p_i}-x_{Id}}{x_{Id}})^2 + (\frac{y_{p_i}-y_{Id}}{y_{Id}})^2 + (\frac{z_{p_i}-z_{Id}}{z_{max}-z_{Id}})^2}$$

For the energy-efficiency and throughput, the upper bounds are determined by the efficiency of IdealMAC (x_{Id}) and the maximum throughput of IdealMAC (y_{Id}).The value z_{max} corresponds to the *worst* measured delay of the simulated E^2-MAC protocols that does not exceed *10s*. Note that in contrast to the efficiency and the throughput, this particular choice of the maximum value for the delay has an influence on the metric itself.

We define the *traffic-adaptivity* of every simulated E^2-MAC protocols as the distance of its *best* configuration P_i and IdealMAC. The best configuration of every E^2-MAC protocol P is its configuration $P_i = (x_{p_i}, y_{p_i}, z_{p_i})$ with the minimal distance to the IdealMAC reference protocol. The traffic-adaptivity TA of a protocol P denoted as a set of its configurations $P_0, P_1, \cdots, P_k$ then yields as:

$$TA(P) = \min d_{norm}(P_i, Id) \quad (P_i \in P)$$

Obviously, the TA-metric based on the normalized Euclidean distance fulfills the mathematical properties of a metric function.

Applying the Metric to today's E^2-MAC protocols

Applying this metric to simulated E^2-MAC protocol yields the results depicted in Figure 10. The figure depicts the protocols' TA-value measured in the abovementioned experiment on the y axis. With taking the normalized distance function d_{norm}, the TA value is in between [0, $\sqrt{3}$] for every protocol.

WiseMAC [7] has proved to be the protocol which offers the best *traffic-adaptivity* of the simulated set of protocols. The preamble sampling mechanism and the learning of the neighboring nodes' schedules leads to a high energy-efficiency, and choosing suitable values for the basic listen interval leads to reasonable multi-hop delays. [22] already supposed that the *dynamic preamble length adjustment results in better performance under variable traffic conditions*, a conjecture for which we have just delivered a proof. However, as the protocol does not yet integrate any self-configuration and self-adaptation mechanisms, the protocol performance is very much dependent of choosing suitable parameter settings for any given scenario. WiseMAC however restrains the maximum achievable throughput quite heavily, compared to energy-unconstrained CSMA.

The protocol T-MAC [8] has proven to achieve the highest throughput at load peaks and a very low latency. Its *traffic-adaptivity* measured using the TA-metric is furthermore not far from that of WiseMAC. The time-out mechanism that prolongs the T-MAC duty cycle noticeably pays off, as T-MAC achieves a much higher throughput than its predecessor S-MAC, and - given suitable parameters for the frame length and contention window - almost reaches the throughput of CSMA. The drawback of the time-out mechanism however is the massively decreasing energy-efficiency.

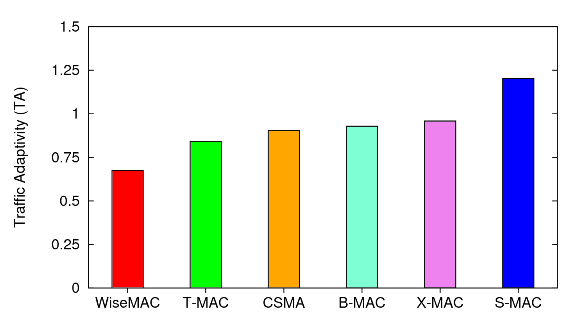


Fig. 10: Traffic Adaptivity TA

Weighted Metric

There is an infinite number of mappings between any three-dimensional space $X \times Y \times Z$ and the real numbers R. For certain scenarios, it might make sense to define a metric where throughput and latency is *more important* than the energy-efficiency and thus has a larger weight. One could therefore redefine $d_{norm}(P_i, Id)$ as $d_{norm}(P_i, Id, \omega_x, \omega_y, \omega_z)$ with weight factors $\omega_x, \omega_y, \omega_z \in [0, 1]$, which account for the importance of the energy-efficiency, throughput and delay:

$$d_{norm}(i, j, \omega_x, \omega_y, \omega_z) =$$

$$\sqrt{\omega_x(\frac{x_{p_i} - x_{Id}}{x_{Id}})^2 + \omega_y(\frac{y_{p_i} - y_{Id}}{y_{Id}})^2 + \omega_z(\frac{z_{p_i} - z_{Id}}{z_{max} - z_{Id}})^2}$$

Setting different values for the weight factors $\omega_x, \omega_y, \omega_z$ hence yields different orderings of the examined protocols.

VII. Conclusions and Future Work

In this paper we have explored the design space of today's most frequently cited E^2-MAC protocols with respect to their ability to react to changing traffic conditions. By comparing against an idealized concept of an E^2-MAC protocol, we have shown how far today's E^2-MAC protocols still are from the goal of being able to truly allocate the radio transceiver in an *on-demand manner*. Many of todays E^2-MAC protocols exhibit a very high energy-efficiency - some of them yet come close to the theoretic lower bounds. This gain in efficiency however comes at the cost of severely restrained maximum throughput, as well as massively increasing end-to-end packet latency.

We envisage to move towards an E^2-MAC that is able to achieve a very high efficiency in case of low traffic (as e.g. WiseMAC), but that is capable to adapt its behavior in case of higher traffic. In such situations, E^2-MAC protocol should be able to exploit the entire channel capacity and achieve a throughput that is similar to that of energy-unconstrained CSMA. Such a maximally-adaptive behavior would be very advantageous in many event-based WSN application scenarios, and would constitute a real novelty in the design space of E^2-MAC protocols.

As we have seen in Section V, different protocol configurations lead to very different behavior and QoS characteristics. Throughout this paper we have unambiguously shown how big the impact of the E^2-MAC protocol parameters are. We envisage to achieve significant improvements by intelligent adaptive tuning of those parameter settings at *run-time*, and the introduction of novel protocol mechanisms in the near future.

References

[1] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An ad hoc Sensor Network Infrastructure for Emergency Medical Care." MobiSys 2004 Workshop on Applications of Mobile Embedded Systems, 2004.

[2] T. Gao, T. Massey, L. Selavo, D. Crawford, B. Chen, K. Lorincz, V. Shnayder, L. Hauenstein, F. Dabiri, J. Jeng, A. Chanmugam, D. White, M. Sarrafzadeh, and M. Welsh, "The Advanced Health and Disaster Aid Network: A Light-weight Wireless Medical System for Triage." IEEE Transactions on Biomedical Circuits and Systems, 2007.

[3] I. Akyildiz, T. Melodia, and K. Chowdhury, "Wireless multimedia sensor networks: A survey." Computer Networks (Elsevier), 2007.

[4] K. Langendoen, "Medium access control in wirelss sensor networks." Nova Science Publishers, May 2008, bookchapter.

[5] W. Ye, J. Heidemann, and D. Estrin, "An Energy Efficient MAC protocol for Wireless Sensor Networks." IEEE Conference on Computer Communications (INFOCOM), 2002.

[6] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks." ACM Conference on Embedded Networked Sensor Systems (SenSys), 2004.

[7] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks." ALGOSENSORS, 2004.

[8] T. Van Dam and K. Langendoen, "An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks (TMAC)."

[9] S. H. Lee, J. H. Park, and L. Choi., "AMAC: Traffic-Adaptive Sensor Network MAC Protocol through Variable Duty-Cycle Operations." IEEE International Conference on Communications (ICC), 2007.

[10] M. Buettner, G. V. Y., E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks." Intl. Conference on Embedded Networked Sensor Systems (SenSys), 2006.

[11] A. Varga, "The omnet++ discrete event simulation system." European Simulation Multiconference, 2001, http://www.omnetpp.org.

[12] W. Drytkiewicz, S. Sroka, V. Handziski, A. Koepke, and H. Karl, "A mobility framework for omnet++." 3rd Intl. OMNeT++ Workshop, 2003, http://mobility-fw.sourceforge.net.

[13] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: the incredibles," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[14] T. R. Andel and A. Yasinsac, "On the Credibility of Manet Simulations," *IEEE Computer Magazine*, 2006.

[15] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," 2004.

[16] T. S. Rappaport, "Wireless Communications: Principles & Practise." Prentice Hall, 2nd Edition, 2001.

[17] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein, and M. Zitterbart, "Introducing Probabilistic Radio Propagation Models in OMNeT++ Mobility Framework and Cross Validation Check with NS-2." 1st International Workshop on OMNeT++, 2008.

[18] L. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment." IEEE Conference on Computer Communications (INFOCOM), 2001.

[19] Texas Instruments CC1020, "Single-Chip FSK/OOK CMOS RF Transceiver for Narrowband Apps in 402-470 and 804-940 MHz Range." http://focus.ti.com/docs/prod/folders/print/cc1020.html, last visit March 2008.

[20] M. Baar, E. Koeppe, A. Liers, and J. Schiller, "The scatterweb msb-430 platform for wireless sensor networks." SICS Contiki Hands-On Workshop, Kista, Sweden, 2007.

[21] J. Ammer and J. Rabaey, "The Energy-Per-Useful-Bit Metric for Evaluating and Optimizing Sensor Network Physical Layers." IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), 2006.

[22] I. Demirkol, C. Ersoy, and F. Alagoz, "Mac protocols for wireless sensor networks: A survey." IEEE Communications Magazine, 2006.