# MaxMAC: a Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks

Philipp Hurni, Torsten Braun

Institute of Computer Science and Applied Mathematics
University of Bern
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 (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 load at run-time.
This paper presents MaxMAC, an $E^2$-MAC protocol that targets at achieving maximal adaptivity with respect to throughput and latency. By adaptively tuning essential parameters at run-time, the protocol reaches the throughput and latency of energy-unconstrained CSMA in high-traffic phases, while still exhibiting a high energy-efficiency in periods of sparse traffic. The paper compares the protocol against a selection of today's $E^2$-MAC protocols and evaluates its advantages and drawbacks.

**Key words:** Wireless Sensor Networks, Energy Efficient Medium Access Control, Traffic Adaptivity

## 1 Introduction

Today's $E^2$-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 concentrated almost exclusively on the energy aspect, introducing tight restrictions with respect to throughput and latency. Such restrictions may be tolerable in networks with low quality of service requirements. However, many event-based scenarios require reasonable quality of service during periods of increased activity, and a high energy-efficiency during long periods of inactivity. Such scenarios can be found e.g. in monitoring systems for healthcare [1], in Disaster-Aid-Systems [2], but also in the broad area of (event-based) environmental monitoring systems. 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 $E^2$-MAC protocols do not provide reasonable flexibility, as most of them were designed under the assumption of very sparse low-rate traffic.

This paper introduces MaxMAC, an energy-efficient MAC protocol for sensor networks designed for WSN scenarios with varying traffic conditions. While MaxMAC operates similarly as existing $E^2$-MAC protocols in low traffic situations, it is able to maximally adapt to changes in the network traffic load at run-time. Taking advantage of design principles for $E^2$-MAC protocols developed over the last couple of years, the protocol introduces novel run-time adaptation techniques to effectively allocate the costly radio transceiver truly in an *on demand* manner. The protocol reaches the throughput and latency of energy-unconstrained CSMA in situations of high-traffic, yet exhibiting a high energy-efficiency in periods of sparse traffic.
The paper is organized as follows: Section 2 discusses related work on the topic of traffic-adaptive $E^2$-MAC protocols. Section 3 then describes the design of the MaxMAC protocol mechanisms. Section 4 presents simulation setup and environment, followed by simulation results in Section 5. Section 6 concludes the paper.

## 2 Related Work

A couple of concepts has yet been applied to reach traffic-adaptive protocol behavior in today's literature on $E^2$-MAC protocols. However, most approaches are minor variations of existing protocols and still heavily restrain throughput and latency of the MAC layer, a crucial disadvantage which often prevents them to be applied in real WSN deployments.
T-MAC [4] increases the traffic-adaptivity of S-MAC [5] 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 that may announce further packet exchanges. However, simulations show that the adaptivity of the protocol is still very limited and that the performance gain of the traffic adaptivity enhancement further only pays off for non-uniform bursty traffic.
X-MAC [6] is an $E^2$-MAC protocol based on asynchronous listen-intervals. For each packet, X-MAC transmits a strobe of preambles, in between which the receiver can signal reception-readiness with a so-called *EarlyACK*. [6] derives a formula for optimal wake/sleep intervals given traffic at a certain rate and outline a mechanism to let X-MAC adapt the duty cycle and the sleep/wake interval to best accommodate the traffic load in the network. With the basic mechanism of X-MAC still requiring a certain minimal interval between two active intervals

and a generally high per-packet overhead, the maximum achievable throughput of the protocol remains very limited.
AMAC [7] is an $E^2$-MAC protocol targeting at traffic-awareness. It relies on the S-MAC active period structure consisting in SYNC, RTS and CTS windows. With low traffic, AMAC neglects the costly RTS/CTS exchange and operates with a large sleep interval between two active periods. With increasing traffic, it multiplies the amount of active periods by a factor of $2^n$, thus increasing the net duty cycle by the same factor. Applying this adaptation strategy, the protocol can prevent packet drops to some extent while still saving energy.
Z-MAC [8] is a TDMA-based protocol that achieves high channel utilization under high contention. The protocol initially gathers topology information and rigidly synchronizes clocks to maintain a collision-free schedule. Under low traffic, its performance with respect to energy-efficiency however remains low.
BurstMAC [9] is a recent $E^2$-MAC protocol targeting at achieving a low idle-overhead *and* a high throughput in case of correlated traffic bursts, as they occur in event-based scenarios. BurstMAC employs multiple channels and keeps a rigid network-wide synchronization and TDMA-scheme, The protocol achieves high throughput in case of correlated event traffic by efficient on-demand allocation of channels, hence letting node pairs communicate concurrently.

## 3 MaxMAC Design

### 3.1 Basic Media Access Mechanism

Many energy-efficient protocol mechanisms for wireless sensor MAC protocols have been developed during the past couple of years. MaxMAC takes advantage of the substantial work carried out on $E^2$-MAC protocols, especially the asynchronous protocols B-MAC [10], WiseMAC [11] and X-MAC [6]. This section briefly discusses the basic media access mechanisms used in MaxMAC, while Section 3.2 discusses its run-time traffic adaptation mechanisms.

***Preamble Sampling:*** With Preamble Sampling (also referred-to as *Low-Power-Listening*) introduced in B-MAC and WiseMAC, nodes keep their radios off for most of the time and only wake up for very brief periodic duty cycles to poll the channel for a preamble signal. The sender node prepends a preamble for each frame that signals the upcoming frame transmission to the receiving node in its short wake-up. In B-MAC, the preamble spans the entire wake-up interval, whereas WiseMAC learns the wake-up schedules of its neighbors to minimize the length of the preambles in future transmissions. A small preamble then only compensates for the maximum clock drift that the two involved node's clocks may have developed during the time since the last schedule exchange. Given that digital crystal oscillators typically exhibit low drifts ($\leq$ 100 ppm), this preamble minimization scheme incurs a low per-packet overhead while still achieving a high packet delivery probability. MaxMAC takes advantage of the WiseMAC preamble-sampling scheme - each node periodically wakes up to sense the channel for a preamble tone within the *Base Interval* T (cf. Figure 1).

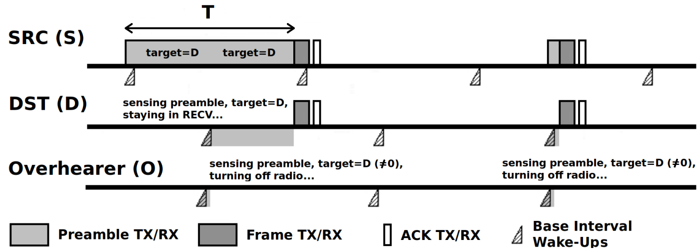


Fig. 1: Preamble sampling with embedded target address in MaxMAC

***Overhearing Avoidance:*** The preamble sampling technique of WiseMAC is already quite efficient in avoiding costly overhearing. With sparse traffic, chances are high that the wake-ups of non-targeted receivers do not coincide with those of the target receivers. With higher traffic, however, and transmissions of queued packet trains, overhearing of preambles and frames becomes an increasing source of energy waste. MaxMAC minimizes overhearing by enriching preambles with target id information, as illustrated in Figure 1. Target nodes turn their radio transceiver on, sense the carrier for *their particular preamble* to receive preamble and frame. Non-target nodes turn their radios on, extract the target information in the ongoing preamble transmission, notice that they are not targeted and immediately turn it off. This concept has been applied in X-MAC [6], where nodes send preamble strobes in-between which receiver nodes can signal reception readiness with a so-called *Early-ACK*. MaxMAC however applies this concept to reduce overhearing in a preamble-sampling MAC protocol, combined with the preamble minimization technique of WiseMAC [11].

### 3.2 Run-Time Traffic Adaptation Mechanisms

In contrast to most of today's $E^2$-MAC protocols, which operate with rather static parameter settings, MaxMAC introduces traffic-adaptation features to instantly react to changing load conditions by altering it's behavior at run-time. MaxMAC attempts to allocate the energy resources of the sensor node in an *on-demand* manner. Similarly as in dynamic frequency/voltage scaling, where the CPU reacts to higher computation load with an increase of the frequency/voltage, a traffic-adaptive $E^2$-MAC protocol should react to changing load conditions by correspondingly tuning the radio transceiver - turning/keeping the transceiver on more frequently when more traffic has to be handled, keeping it permanently on during load peaks, and turning it off again when the load level permits it.

***Allocation/Deallocation of Extra Wake-Ups:*** With $E^2$-MAC protocols alternating between sleep and wake intervals, throughput is often restrained to a couple of frame transmissions in each interval. Latency typically increases sharply, as forwarding nodes need to buffer incoming frames and wait for the next wake-up of their gateway node, which often sums up to some seconds in multi-hop scenarios. The first traffic adaptation feature and essential novelty of

MaxMAC tackles this very decisive $E^2$-MAC protocol restriction. In MaxMAC, nodes change their state (and hence their behavior) and allocate so-called *Extra Wake-Ups* when the rate of incoming packets reaches predefined threshold values, and de-allocate them when the rate drops below the threshold again.

Figure 2 illustrates the state-based adaptivity mechanism with a source node (SRC) sending packets to a receiver node (DST) with increasing rate. Nodes operate in the *Base Interval* state per default, polling the channel periodically within the Base Interval T. Nodes alter their state (and behavior) by switching to states $S_1$, $S_2$ when the corresponding thresholds $T_1$, $T_2$ are reached. Thresholds $T_1$ and $T_2$ are set to 2 and 6 packets/s in the illustration in Figure 2. Each node keeps estimating the rate of incoming packets, using a sliding window of 1s (cf. rate-estimation graph of DST in Figure 2). With the rate of incoming packets reaching the threshold $T_1$, the DST schedules one additional *Extra Wake-Up* in-between each Base Interval, effectively doubling the amount of duty cycles over time. The receiver node DST communicates its increased wake-up frequency in the ACK. SRC receives this announcement and marks the increased wake-up frequency of node DST in its schedule offset table. With the notification sent by DST in the ACK, DST *promises* to remain in the new state and keep its increased wake-up frequency for a predefined timespan S1_LEASE. For each state in MaxMAC, the LEASE timespans (S1_LEASE, S2_LEASE, CSMA_LEASE) define how long a node *promises* to remain in the new state when announcing the state change in the ACK. LEASE timespans can further be *prolonged* in any new ACK transmission. By remaining in a higher state for at least the LEASE duration, fast oscillation between the different states can be mitigated. With the rate of incoming packets reaching the threshold $T_2$, DST changes to state $S_2$, doubles the amount of wake-ups again and announces its state change in the ACK (cf. Figure 2). As soon as these timespans expire, nodes having received prior state change announcements will assume that the corresponding node has fallen back to its default behavior (polling the channel with the Base Interval T), which prevents them from transmitting at instants when the target is not awake. All LEASE timespans are set to 1s in the subsequent experiments.

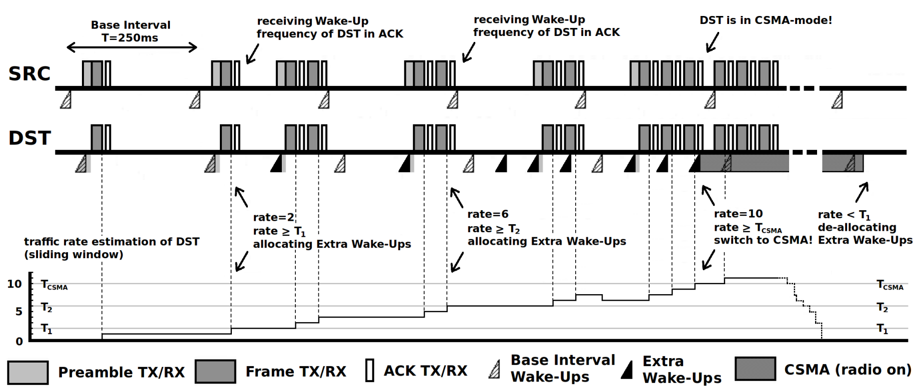


Fig. 2: Adding Extra Wake-Ups with increasing rate of traffic

Increasing the amount of wake-ups is an effective, yet considerably cheap means of increasing network throughput and decreasing end-to-end latency. If SRC needs to forward other packets, the time to wait for the next wake-up of DST is halved with DST being in state $S_1$ or even quartered with DST being in state $S_2$. However, if the additional wake-ups scheduled by DST are not used for transmissions, the waste of energy remains limited, as some few additional channel polls are energetically inexpensive.

***Exploiting the Channel Capacity by switching to CSMA:*** Most existing $E^2$-MAC protocols have been designed under the assumption of sparse low-rate traffic. Hence, these protocols severely restrain throughput, compared to energy-unconstrained wireless channel protocols. In multi-hop scenarios, S-MAC, T-MAC and WiseMAC have been shown to reach only a fraction of that of CSMA [12] [13]. MaxMAC has been specifically designed to achieve a throughput similar as CSMA in situations of increased network activity, after a certain delay for triggering the adaptation mechanisms. While the allocation of *Extra Wake-Ups* helps to achieve a somewhat increased throughput, CSMA-like throughput and latency can not yet be reached with it. MaxMAC thus carries the threshold-based concept one step further. When the rate of incoming packets reaches a further threshold $T_{CSMA}$ (with $T_{CSMA} > T_2 > T_1$), MaxMAC switches to energy-unconstrained CSMA and announces this state change to the sender node (and potentially overhearing child nodes) in the ACK. Figure 2 illustrates node DST measuring the rate of incoming packets to reach $T_{CSMA} = 10$ packets/s in the right part of the figure. Node DST hence switches to the CSMA state, announcing the state change to SRC in the ACK, hence *promising* to remain in the CSMA state for at least the predefined timespan CSMA_LEASE. Within this timespan, SRC can transmit packets without having to wait for a wake-up of DST, as it *knows* that DST keeps its transceiver on for at least the timespan CSMA_LEASE. With CSMA_LEASE expiring, all nodes having received the prior state change announcement of DST assume that DST has fallen back to the Base Interval state, which prevents them from transmitting at times when DST is asleep.

Figure 3 illustrates the state-based adaptivity concept of MaxMAC with the state transitions as a finite state machine. Nodes switch from the Base Inter-

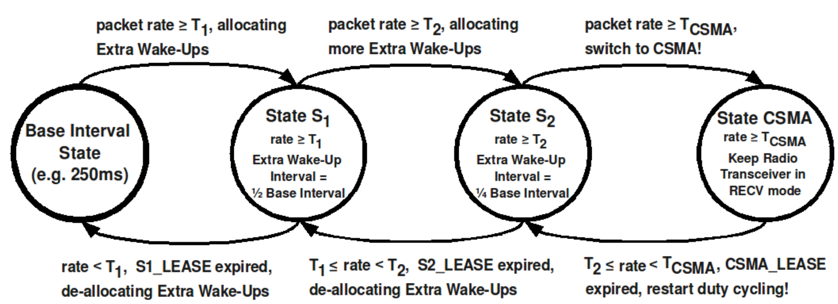


Fig. 3: State-based traffic adaptivity mechanism of MaxMAC

val state to a higher state $S_1, S_2, CSMA$ when the rate reaches the associated thresholds $T_1, T_2, T_{CSMA}$. When switching from the Base Interval state to $S_1$ or $S_2$, nodes schedule *Extra Wake-Ups* and double or quadruple their wake-up frequency, which increases network throughput and reduces end-to-end latency. When the rate reaches the threshold $T_{CSMA}$, nodes switch to energy-unconstrained CSMA and keep their radio transceivers turned on. With the load falling below $T_{CSMA}$ and CSMA_LEASE expiring, nodes switch again to states $S_1$ or $S_2$ and restart alternating between brief channel polls and long sleep intervals. Nodes completely de-allocate all *Extra Wake-Ups* and fall back to the Base Interval state when the packet rate drops below $T_1$ and all LEASE timespans have expired. The MaxMAC traffic adaptation mechanism scales well for multi-hop topologies, as each node measures and reacts upon a given rate increase in a decentralized manner. MaxMAC further communicates state changes efficiently, without introducing any new control messages. All the necessary control information is communicated in the Data frame header and the ACK frames.
This section illustrates the MaxMAC adaptivity concept with three states $S_1$, $S_2$, $CSMA$ - the number of states and thresholds can however be chosen arbitrarily. The threshold values $T_1, T_2, T_{CSMA}$ we choose in Section 5 were calibrated for the particular given scenarios. The thresholds allow the network operator for fine-tuning the MaxMAC protocol and its properties. Choosing e.g. low values for the thresholds makes sense in delay-sensitive applications, whereas higher values can make sense in energy-sensitive and delay-tolerant applications. We intend to study self-parametrization mechanisms based on estimation of available channel bandwidth, link quality, hopcount, network density in the near future.

## 4 Simulation Models and Parameters

We implemented the MaxMAC protocol and compared it to S-MAC [5], T-MAC [5], B-MAC [10], WiseMAC [11], X-MAC [6], and the reference protocols IdealMAC and energy-unconstrained CSMA in the OMNeT++ Network Simulator [14]. The IdealMAC protocol has been used in [11] as a reference protocol 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.
In order to reflect the characteristics of wireless propagation (high packet error rate, shadowing and fading-effects), we applied the Log-Normal Shadowing Model [15] implemented in [16]. This channel model allows for a more realistic simulation of wireless channel properties than usual Unit Disk Graph (UDG) based simulation models. It models small-scale shadowing and fading effects - which are typical wireless phenomena - for each frame transmission by adding a random perturbation factor to the reception power. The perturbation factor follows a log-normal distribution with a user-selectable deviation $\sigma$.

| | | | |
|---|---|---|---|
| CC1020 [18] parameters | | Experiment parameters | |
| supply voltage $V$ | 3 $V$ | simulation runs | 100 |
| transmit current $I_{tx}$ | 21.9 $mA$ | simulated time | 3600 s |
| recv current $I_{rx}$ | 17.6 $mA$ | ARQ max retries | 3 |
| sleep current $I_{sleep}$ | 1 $\mu A$ | frame header size | 14 bytes |
| transmission rate $R$ | 115.2 $kbps$ | payload | 50 bytes |

Table 1: Simulation model parameters

***Transceiver and Energy Model:*** We modeled the state transition delays and the power consumption of wireless sensor nodes using a finite state machine model consisting in the states sleep, receive and transmit, weighted with the respective energy costs. The same methodology is applied in [17], where the power consumption of a IEEE 802.11 wireless device is modeled with the same three states. Experimental results in [17] confirm the adequateness of the linear state transition model. Table 1 lists current, voltage and transmission rate of the CC1020 [18], a byte-level radio transceiver in the 804-940 MHz ISM frequency band. The CC1020 is used by the MSB430 sensor nodes platform [19], which we use for prototyping traffic-adaptive $E^2$-MAC protocols on real sensor hardware.

***$E^2$-MAC Protocol Simulation Models:*** Table 2 displays the main parameters of the simulated $E^2$-MAC protocols. As the protocol behavior often heavily depends on the choice of the essential protocol parameters (e.g. Base Interval, Duty Cycle), we studied the protocols with different *configurations* of those parameters, by varying the parameters over a wide range, and not just one particular parameter choice. One such *configuration* would e.g. be B-MAC [Base Interval=200ms, Duty Cycle=1%(2ms)].
For the *slotted* protocols S-MAC and T-MAC, we assume that the nodes' wake-up intervals are synchronized from the beginning of the experiment (as assumed

| | | | |
|---|---|---|---|
| MaxMAC | | B-MAC | |
| Base Interval | 100, 200, 250 ms | Base Interval | 25, 50, 100, |
| Duty Cycle | 2, 1, 0.8% | | 200, 500 ms |
| LEASE | 1 s | Duty Cycle | 8, 4, 2, 1, 0.4% |
| $T_1$, $T_2$, $T_{CSMA}$ | 4, 8, 12 packets/s | WiseMAC | |
| S-MAC | | Base Interval | 25, 50, 100, |
| Listen Interval | 100, 200, 300, 500 | | 200, 500 ms |
| | 1000, 2000 ms | Duty Cycle | 8, 4, 2, 1, 0.4% |
| Duty Cycle | 10% | Medium Reservation | u[0,10] $\times$ $t_{rx-tx}$ |
| T-MAC | | X-MAC | |
| Frame Length | 50, 100, 200 ms | Max Interval | 200 ms |
| | 300, 500 ms | Min Interval | 10 ms |
| SYNC & RTS size | 14 bytes | EarlyACK size | 10 bytes |
| CTS size | 10 bytes | CSMA | |
| SYNC period | 10 s | Contention Window | 10 ms |

Table 2: $E^2$-MAC Protocol Parameters

in many MAC studies, e.g. in [11]). With X-MAC, we integrated an adaptation algorithm that adapts the wake/sleep intervals according to incoming packet rate (as specified in [6]), but remains in-between [Max Interval, Min Interval].
WiseMAC implements a cheap collision avoidance using a larger *carrier sensing range* ($\sim 2 \cdot hop\ distance$). 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 of the $E^2$-MAC protocol models, we implemented the same packet burst transfer mode for each protocol. Nodes signal pending packets to the receiver and can transmit queued packet trains in bursts, receiving an acknowledgment for each frame.

## 5 Simulation Results

***5.1 Traffic along a Multi-Hop Chain:*** We simulated a chain consisting of 8 nodes. The source node is generating load, which is then forwarded hop-by-hop towards the sink node, similarly as done in the studies on S-MAC [5] and

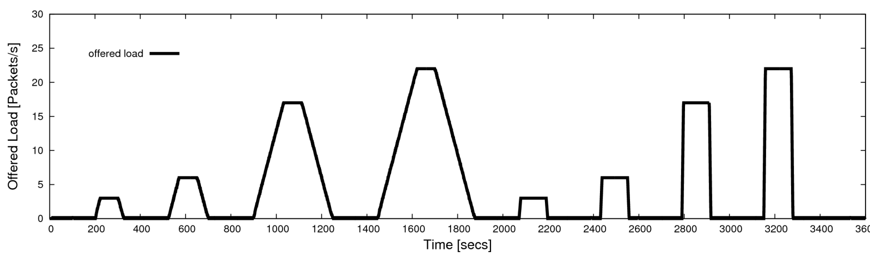


Fig. 4: Offered Load (Packets/s)

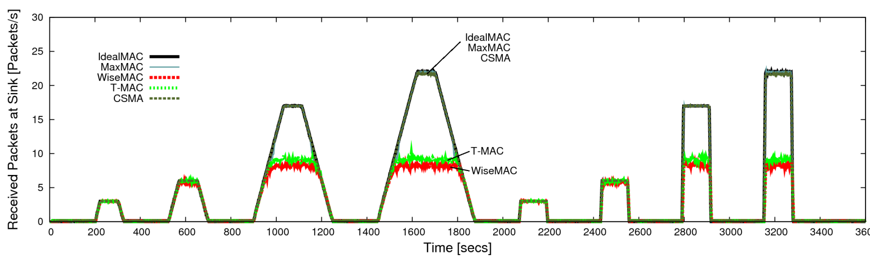


Fig. 5: Throughput at Sink

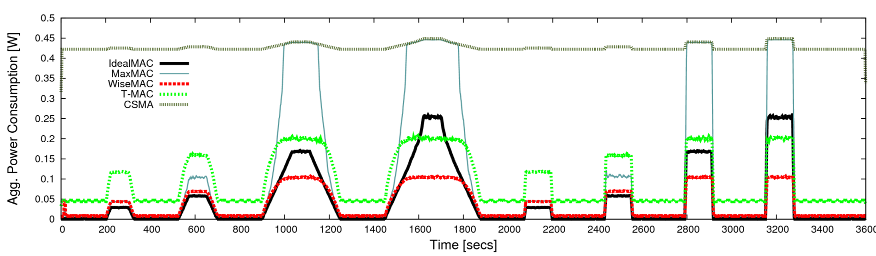


Fig. 6: Aggregated Network Power Consumption

B-MAC [10]. Almost every existing study on $E^2$-MAC protocols applies constant rate traffic during each simulation run. In contrast to this, we varied the offered traffic from low rates to high rates during each run, as our major interest is the protocol adaptivity during *run-time*.
Figure 4 displays the offered load generated at the application layer of the source node. The load is low (0.1 packets/s) for most of the time, but there are peaks where the packet rate is increased, up to a maximum rate of 22 packets/s. 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 additional packets are either dropped due to buffer overflows or are lost due to collisions.

***Throughput and Power Consumption:*** Figure 5 displays the rate of received packets at the sink node vs. simulation time. The 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 sink. CSMA only suffers minor packet loss at the load peaks. The throughput of WiseMAC and T-MAC stalls at maximum 8 packets/s and 9 packets/s, respectively, which corresponds to $\sim 35 - 40\%$ of that of CSMA. Figure 5 clearly shows that MaxMAC with its state-based run-time traffic adaptation mechanism reaches the same throughput as energy-unconstrained CSMA. As the protocol adaptively allocates more duty cycles or even totally switches to CSMA-like behavior at high traffic rates, the protocol manages to handle the load peaks without major packet loss.
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 consume much more power when transmitting, compared to idle listening [18]. 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. WiseMAC renouncing on costly synchronization schemes has a low per-packet overhead, minimizing preambles by learning adjacent nodes' schedules. It exhibits a low power consumption during the low traffic phases, its throughput however stalls at $\sim 35\%$ of that of CSMA. T-MAC achieves a slightly higher throughput, but its idle power consumption is above that of WiseMAC, mainly due to the SYNC message overhead to keep the nodes' wake-ups synchronized.
Thanks to the run-time traffic-adaptivity mechanisms of MaxMAC, namely the scheduling of *Extra Wake-Ups*, and the switch to energy-unconstrained CSMA-like behavior with higher traffic load, MaxMAC reaches the same energy-efficiency in the low-traffic-phases as WiseMAC, but is able to handle the load peaks with much lower packet loss. As MaxMAC switches to the CSMA-state with the rate reaching $T_{CSMA} = 12$ packets/s (cf. Table 2), the power consumption of MaxMAC accordingly jumps to the level of CSMA at this rate, too. Figure 6 further illustrates that the *on-demand* resource allocation scheme of MaxMAC further

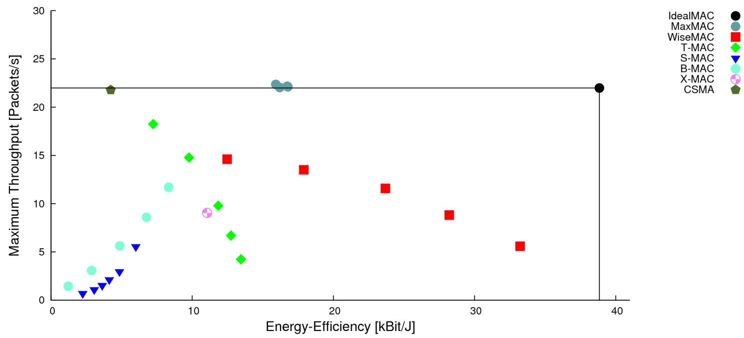


Fig. 7: Throughput vs. Energy-Efficiency

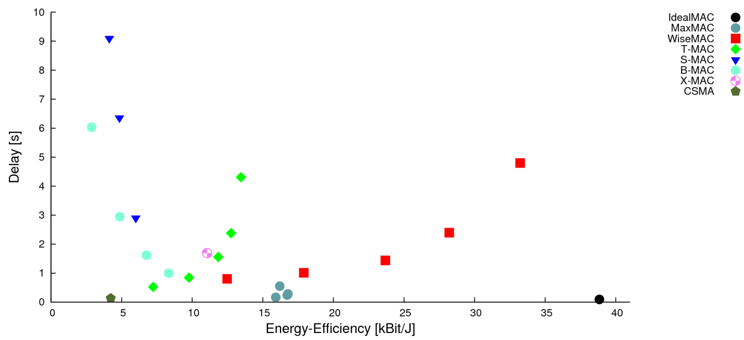


Fig. 8: Delay vs. Energy-Efficiency

succeeds astonishingly well when the packet rate decreases. With traffic rates decreasing towards 0.1 packets/s after the load peaks, MaxMAC quickly falls back to the states $S_2$ and $S_1$ and finally the Base Interval state, where it again exhibits a very low energy-footprint.

***Energy-Throughput and Energy-Latency Tradeoffs:*** $E^2$-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 subsection we examine the MaxMAC protocol with respect to the energy-throughput and energy-latency tradeoffs and compare it with existing $E^2$-MAC protocols. Figure 7 and 8 illustrate the measured tradeoffs in the aforementioned experiment. Each dot represents the results of one particular protocol *configuration* in the simulation experiment outlined in Section 4. In Figure 7, the tradeoff between maximum achieved throughput and energy-efficiency of the simulated $E^2$-MAC protocols becomes well visible. 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 [20] - we however use the reciprocal coefficient in order to obtain a metric where *more is better*. CSMA obviously achieves a high maximum throughput. However, as CSMA never turns off the transceiver, its energy-efficiency remains very low.
IdealMAC illustrates the lower bounds of the $E^2$-MAC protocol problem in Fig-

ures 7 and 8: while it is not possible to reach a higher throughput or a higher efficiency coefficient than IdealMAC, it is neither possible to reach a lower delay. WiseMAC with its short channel polls achieves a high energy-efficiency, especially the configurations with long intervals between two channel polls. The efficiency gain however comes at the cost of a massively restrained maximum throughput and increasing end-to-end latency (cf. Figure 8).
Thanks to its run-time traffic adaptation mechanisms, MaxMAC reaches the same throughput as energy-unconstrained CSMA, but exhibits a much higher energy-efficiency in terms of kbit/J. Although MaxMAC switches to CSMA-like behavior in the high traffic phases, its efficiency coefficient is higher than that of most of today's $E^2$-MAC protocols. The advantage of achieving the high throughput of CSMA *and* a much better energy-efficiency than most $E^2$-MAC approaches is a clear novelty in the design space of today's $E^2$-MAC protocols.

Figure 8 similarly depicts the tradeoff between average packet delay and energy-efficiency. One can observe that CSMA exhibits a very low average delay, however at the cost of a low energy-efficiency. IdealMAC reaches both, a very low delay at a very high energy-efficiency. Thanks to the scheduling of *Extra Wake-Ups*, which reduces the interval between two wake-ups, and the switch to CSMA-like behavior at even higher rates, MaxMAC reaches a far lower average end-to-end latency as other $E^2$-MAC protocols. MaxMAC achieves a delay which is - given the best examined configuration - only 70% higher than that of CSMA (compared to some 1000% with other $E^2$-MAC protocols), but achieves an energy-efficiency that is more than three times better than that of of CSMA.

Figure 9 represents the results of each configuration of the simulated $E^2$-MAC protocols as a tuple in the vector space $X \times Y \times Z$ where $X$ is the energy-efficiency (measured in kbit/J), $Y$ the maximum achievable throughput (packets/s) and $Z$ the average measured delay. The figure illustrates the potential for optimization in the design space of today's $E^2$-MAC protocols. In [13], we surveyed and compared the adaptivity of the protocols under variable load, using the distance to

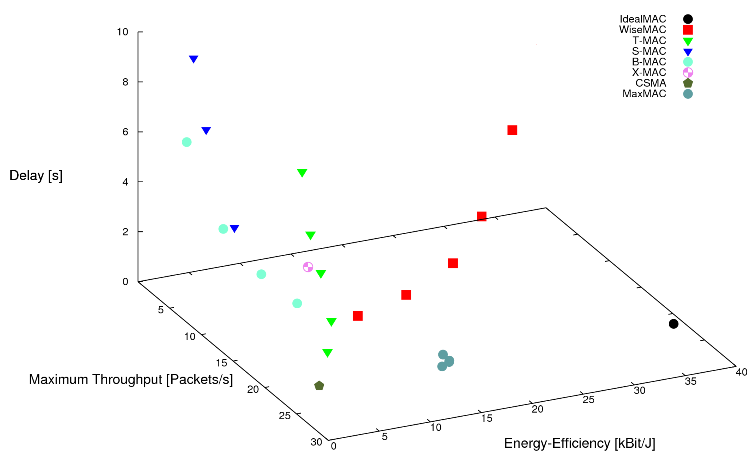


Fig. 9: Energy-Efficiency (x) vs. Maximum Throughput (y) vs. Delay (z)

IdealMAC as a metric to assess the adaptivity of a protocol. [13] concludes that most protocols are not sufficiently adaptive, as they do not alter their behavior with respect to the load conditions. Although there is sufficient channel capacity, most existing protocols still turn their radio transceivers off too aggressively. MaxMAC is clearly distinguishable from the examined reference protocols by its ability to reach the same throughput and a similarly low latency as energy-unconstrained CSMA, while still exhibiting a good energy-efficiency during the considerably long periods of sparse network activity. The three examined *configurations* of MaxMAC hence exhibit the shortest distance to the IdealMAC protocol in the lower right corner in Figure 9, due to the high throughput, low delay and good energy-efficiency measured in the experiment.

***5.2 Random Correlated Event Traffic:*** With our second experiment we examine the behavior of MaxMAC (and the reference protocols) in a larger scenario with a *correlated event workload* model [21]. We simulate a 49-node grid network (7x7) with the center node forming the sink. The distance between two adjacent nodes is 30m. With our parameter settings of the LogNormal channel model [15], packet error rates are $\sim 1\%$ and $\sim 15\%$ on a straight link (30m) and a diagonal link (42.42m), respectively.
We apply a simple event traffic model that mimicks the effects of spacially-correlated events, as proposed in [21] and [22]. Spacially-correlated events are expected to occur in many event-based scenarios for WSNs, e.g. monitoring applications in healthcare [1] systems, disaster-aid systems [2] or tracking applications. The traffic model picks a uniform random (x,y) location for each event. Every node within the event sensing range $R$ of this location then reports data packets with a rate of $r_{event}$ during $t_{event}$ towards the sink. We chose values of $R = 30m$, $r_{event} = 6$ packets/s and $t_{event} = 10s$ for the events being triggered each $30s$ at a random location (x,y) of the simulated network. In large event-based scenarios (e.g. a monitoring application), the packet delivery rate (PDR) is usually given higher priority than the throughput per second. We hence measured the packet delivery rate, the average source-to-sink packet delay and the energy-efficiency (in terms of kBit/J) during 100 runs of $3600s$. Packets are routed along the *shortest path*. Nodes select their parent node randomly in the initiation phase of the experiment if there are multiple nodes advertising the same hop count. Energy-efficiency is measured as the total received data bits divided by the aggregated energy spent by all the node's radio interfaces.

Figure 10 depicts the packet delivery rate (PDR) vs. energy efficiency of the different *configurations* of the $E^2$-MAC protocols in the random correlated event experiment. Energy-unconstrained CSMA and IdealMAC reach a PDR of almost 100%. Some packets are lost due to buffer overflows, as the transmit buffer is assumed to be limited to 10 packets. As CSMA does not turn off the transceiver during the long periods where no traffic occurs, its energy-efficiency remains very low (cf. top-left corner). IdealMAC modeling the *ideal* $E^2$-MAC protocol behavior reaches the same PDR and a very high efficiency (cf. top-right corner). The configurations of T-MAC and WiseMAC with a short *Base Interval* reach a high PDR, however at the cost of decreasing energy-efficiency. B-MAC and X-MAC

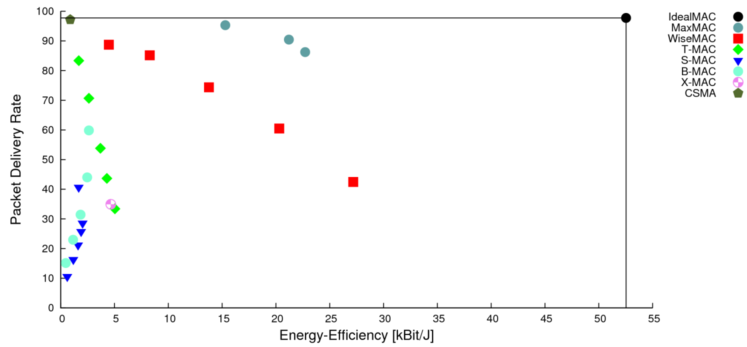


Fig. 10: 49-nodes grid scenario: Packet Delivery Rate (PDR) vs. Energy-Efficiency

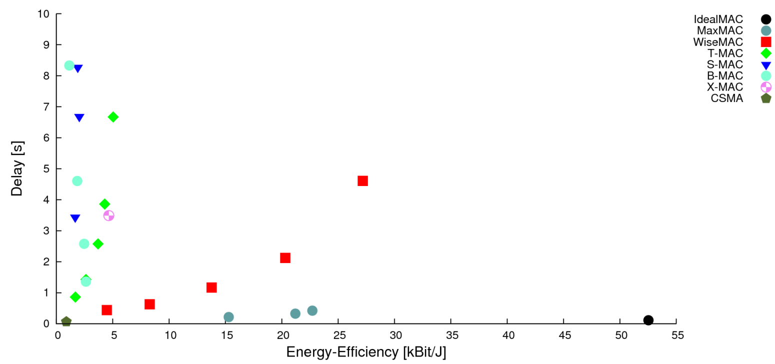


Fig. 11: 49-nodes grid scenario: Delay vs. Energy-Efficiency

reach a modest PDR, but the high per-packet overhead of the B-MAC preambles (which stretch over one entire Base Interval) and the X-MAC preamble strobes negatively impact on their efficiency. Thanks to its run-time adaptation mechanisms, MaxMAC reaches a similar PDR as energy-unconstrained CSMA, while still exhibiting a much higher energy-efficiency. Although the protocol switches to CSMA in the high traffic phases, its overall efficiency is still higher than that of most other $E^2$-MAC protocols. The combination of a high PDR and a high energy-efficiency achieved by MaxMAC's adaptation mechanisms is well-visible in Figure 10 and constitutes a clear benefit.

Figure 11 depicts the tradeoff between average source-to-sink packet delay and energy efficiency in the random correlated event experiment. CSMA again exhibits a very low average delay at the cost of a very low energy-efficiency, while IdealMAC reaches both, low latency and high energy efficiency. The configurations of T-MAC and WiseMAC with a short *Base Interval* reach a lower average delay, however at the cost of decreasing energy-efficiency. B-MAC and X-MAC have a considerably high delay. As these protocols use long preambles or preamble strobes, latency increases sharply over multiple hops, and sums up to a couple of seconds in the given scenario.
Thanks to the scheduling of *Extra Wake-Ups* and switching to CSMA at higher

rates, the three examined *configurations* of MaxMAC reach a far lower average source-to-sink latency as all the other $E^2$-MAC protocols. The adaptivity concept of MaxMAC further fits to the event-based traffic: with an event being triggered at a random location, nodes start reporting data along the shortest path to the sink. With the load reaching the MaxMAC thresholds $T_1, T_2, T_{CSMA}$, nodes alter their behavior in order to deliver the pending load. After the event has been processed and the packet stream ends, the LEASE timespans time out and MaxMAC again falls back to the default behavior in the Base Interval state.
A drawback of MaxMAC is the fact that the protocol requires a certain time during which the adaptation mechanisms are triggered. In multi-hop scenarios, all nodes forming a route from the event source to the sink first need to reach the given thresholds. During this *adaptation phase*, packets are lost mainly due to buffer overflows, as the PDR in Figure 10 exhibits. Thereafter the traffic adaptation strategy achieves a high throughput and a low average delay.

## 6 Conclusions

In this paper we have presented MaxMAC, an $E^2$-MAC protocol that targets at achieving maximal run-time traffic adaptivity. The protocol targets at event-based sensor network applications where at certain instants, the provision of high throughput and fast end-to-end response time becomes more important than the conservation of energy. We envision such applications e.g. in healthcare, where nodes attached to patients need to rely on the provision of higher throughput and fast response times when critical values have been sensed, in order to communicate with central entities.
The paper examines MaxMAC in a network simulator and compares it against a selection of other well-known $E^2$-MAC protocols, an ideal $E^2$-MAC protocol model and energy-unconstrained CSMA. In both scenarios, MaxMAC is clearly distinguishable from the examined reference protocols by its ability to reach the same throughput and a similarly low latency as energy-unconstrained CSMA, while still exhibiting a good energy-efficiency during long periods of sparse network activity, which are often encountered in event-based monitoring systems. The MaxMAC protocol hence combines the advantages of energy unconstrained CSMA (high throughput, high PDR, low latency) with those of classical $E^2$-MAC protocols (high energy-efficiency).

## References


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

[2] Gao, T. et al: The Advanced Health and Disaster Aid Network: A Light-weight Wireless Medical System for Triage, IEEE Transactions on Biomedical Circuits and Systems (2007)

[3] Akyildiz, I., Melodia, T., Chowdhury, K.: Wireless Multimedia Sensor Networks: A Survey, Elsevier Computer Networks (2007)

[4] Van Dam, T., Langendoen, K.: An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks (TMAC), ACM SenSys (2003)

[5] Ye, W., Heidemann, J., Estrin, D.: An Energy Efficient MAC protocol for Wireless Sensor Networks, INFOCOM (2002)

[6] Buettner, M., Y., G.V., Anderson, E., Han, R.: X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks, ACM SenSys (2006)

[7] Lee, S.H., Park, J.H., Choi., L.: AMAC: Traffic-Adaptive Sensor Network MAC Protocol through Variable Duty-Cycle Operations, ICC (2007)

[8] Injong Rhee, Ajit Warrier, M.A., Min, J.: Z-MAC: a hybrid MAC for Wireless Sensor Networks, ACM SenSys (2005)

[9] Ringwald, M., Roemer, K.: BurstMAC: An Efficient MAC Protocol for Correlated Traffic Bursts, IEEE Conference on Networked Sensing Systems (INSS) (2009)

[10] Polastre, J., Hill, J., Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks, ACM SenSys (2004)

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

[12] El-Hoiydi, A.: Energy Efficient Medium Access Control for Wireless Sensor Networks, PhD Thesis EPFL Lausanne (2005)

[13] P. Hurni and T. Braun: On the Adaptivity of Today's Energy-Efficient MAC Protocols under varying Traffic Conditions, IEEE Conference on Ultra-Modern Technologies (ICUMT) (2009)

[14] Varga, A.: The OMNeT++ Discrete Event Simulation System, European Simulation Multiconference (2001) http://www.omnetpp.org.

[15] Rappaport, T.S.: Wireless Communications: Principles & Practise, Prentice Hall, 2nd Edition (2001)

[16] Kuntz, A., Schmidt-Eisenlohr, F., Graute, O., Hartenstein, H., Zitterbart, M.: Introducing Probabilistic Radio Propagation Models in OMNeT++ MF and Cross Validation Check with NS-2, 1st Intl. Workshop on OMNeT++ (2008)

[17] Feeney, L., Nilsson, M.: Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment, INFOCOM (2001)

[18] Texas Instruments CC1020: Single-Chip FSK/OOK CMOS RF Transceiver

[19] Baar, M., Koeppe, E., Liers, A., Schiller, J.: The ScatterWeb MSB-430 Platform for Wireless Sensor Networks, SICS Contiki Workshop (2007)

[20] Ammer, J., Rabaey, J.: The Energy-Per-Useful-Bit Metric for Evaluatling and Optimizing Sensor Network Physical Layers, SECON (2006)

[21] Hull, B., Jamieson, K., Balakrishnan, H.: Mitigating Congestion in Wireless Sensor Networks, ACM SenSys (2004)

[22] Yanjun Sun and Shu Du and Omer Gurewitz and David B. Johnson: DW-MAC: a Low Latency, Energy Efficient Demand-Wakeup MAC protocol for Wireless Sensor Networks, ACM MobiHoc (2008)