

An Energy-Efficient Broadcasting Scheme for Unsynchronized Wireless Sensor MAC Protocols

Philipp Hurni, Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern, 3012 Bern, Switzerland

Abstract—In the past couple of years, many Energy-Efficient Medium Access Control (E^2 -MAC) protocols for all kinds of wireless networks have been proposed. Many of them are based on *Preamble Sampling* (also referred-to as *Low Power Listening*) with *asynchronous* wake-up schemes and brief periodic polls for channel activity. Although these protocols have proven to almost reach the theoretic lower bounds of energy-efficiency in case of unicast point-to-point transmissions, they yet lack an efficient solution to implement a one-hop broadcast and a network-wide broadcast mechanism. This paper proposes an energy-efficient broadcasting scheme for use in E^2 -MAC protocols with asynchronous duty cycles. We integrate the technique into the asynchronous wireless sensor MAC protocol WiseMAC and show that the technique is superior to the existing WiseMAC broadcast mechanism with respect to packet delivery rate, latency and energy consumption through simulation study and experiments with a real-world prototype implementation.

Index Terms—Wireless Sensor Networks, Energy Efficiency, Medium Access Control, Broadcast

I. INTRODUCTION

The challenge in the design of E^2 -MAC protocols consists of finding means to use the wireless transceiver in an *on demand* manner. In wireless sensor networks (WSNs), this task is of crucial importance, as the transceiver hardware is accountable for a major portion of a WSN node’s energy consumption. Hence, all of today’s E^2 -MAC protocols periodically switch the radio transceiver between the costly operation modes receive and transmit, and an energy-conserving sleep mode. Many existing protocols try to synchronize state changes of the nodes in order to exchange pending traffic or control messages in a common interval. Synchronization, however, is not easy to achieve, especially over multiple hops, and periodic synchronization messages may become costly. Hence, wireless sensor MAC protocols renouncing on global or cluster-wise synchronization have recently been proposed. The protocols B-MAC [1], WiseMAC [2], X-MAC [3], C-MAC [4] are based on asynchronous wake intervals and have proven to be more energy-efficient in scenarios with low or varying traffic load. WiseMAC exhibits a very high efficiency for scenarios of low or variable traffic requirements. One crucial drawback of WiseMAC and other WSN MAC protocol based on asynchronous sampling intervals still consists of the difficulty to implement an efficient single-hop and network-wide broadcast. As the nodes all poll the channel in their own wake-up pattern, all receiving nodes need to be alerted by prepending costly long preambles first. Network-wide broadcasting however is very frequently used by routing and application layer protocols in wireless ad hoc and sensor networks. Almost all on-demand routing protocols (e.g. DSR [5], AODV [6], Directed Diffusion [7]) rely on flooding mechanisms to disseminate

route requests. Furthermore, flooding is used in particular in query-driven sensor networks to distribute data requests, and to distribute configuration, status and code updates. Therefore, the development of energy-efficient broadcasting techniques is highly important in WSNs. As sending and receiving long preambles is energetically costly and inefficient, [8] and [9] both similarly criticize this vital drawback of WiseMAC in their survey and evaluation of today’s wireless sensor MAC protocols.

The paper introduces into related work in Section II. The section portrays the basic mechanism of the E^2 -MAC protocol WiseMAC and introduces the difficulties of broadcasting in wireless ad hoc and sensor networks. Section III proposes the energy-efficient broadcasting scheme for use in E^2 -MAC protocols with asynchronous duty cycles. The integration of the technique into WiseMAC forms the entry point for the performance evaluation. The section discusses the algorithm to implement the energy-efficient broadcast scheme, evaluates the prerequisites and assumptions under which this technique delivers energy-efficiency gains and concludes with an analytical proof. Section IV examines the performance of both techniques in a simulator environment and on real-world sensor hardware. Section V concludes the paper.

II. RELATED WORK

A. Wireless Sensor MAC (WiseMAC)

WiseMAC [2] senses the channel for a preamble signal with short periodic duty cycles. All nodes in the network poll the channel with a common basic cycle duration T , but their wake-up patterns are independent and left unsynchronized. When transmitting a frame, a preamble of variable length is prepended for alerting the receiving node. When the receiver’s wake-up pattern is still unknown, the duration of the preamble equals the full basic cycle duration T , as illustrated in Figure 1. The preamble is a simple bit sequence indicating an upcoming transmission to the node’s neighborhood. The own schedule offset (the time to the next wake-up) is then piggybacked to the frame and transmitted to the receiver. After successful frame reception, the receiver node piggybacks its own schedule to

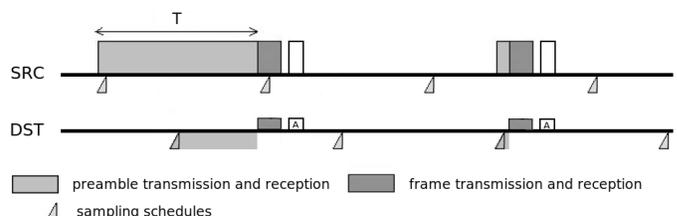


Fig. 1: WiseMAC

the respective frame acknowledgment. Learning each other's schedule offsets allows nodes to minimize the preamble length for upcoming 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 L . Its duration calculates as:

$$P_{\text{WiseMAC}} = \min(4\theta L, T) \quad (1)$$

θ denotes the quartz oscillator clock's drift, L the time since the last update of the neighbor's wake pattern and T the common basic cycle interval duration. θL is the time a clock maximally drifts within L . As it can advance by θL or lag behind by this amount ($2\theta L$), and because two clocks are involved (sender and receiver), the preamble has to span $4\theta L$ in order to guarantee that the receiver is reached. At maximum, a preamble that stretches over the entire interval T is necessary, as each node is guaranteed to wake up and poll the channel within T .

The WiseMAC broadcast is energetically costly and inefficient. The approach consists of prepending a preamble of the duration of the full basic interval duration T to each frame to alert all its (possibly yet undiscovered) neighboring nodes to stay awake for the upcoming transmission of the broadcast frame, essentially the same mechanism as applied in B-MAC. This broadcasting scheme wastes a lot of energy for sending and receiving long preambles, while the actual data (payload) transmission is often comparatively short. If every broadcast message has to be rebroadcast by every node to implement network-wide flooding, the wireless-channel characteristic *broadcast storm problem* is certain to occur.

B. Broadcast Storm

Network-wide broadcasting in wireless ad hoc and sensor networks is still an open research topic. Disseminating packets across multi-hop network topologies yields characteristic wireless channel problems. The most widely known broadcast storm problem, as outlined by *Ni et al.* [10], consists of the following three aspects:

- *Redundancy*: Figure 2 illustrates a node initiating a network-wide broadcast (a). The message is received by its neighboring nodes, which all rebroadcast the packet (b). As illustrated with the *dotted grey arrows*, many transmissions and especially receptions are redundant. As receptions are likewise costly in wireless networks, this problem has severe impact on the energy consumption of the participating

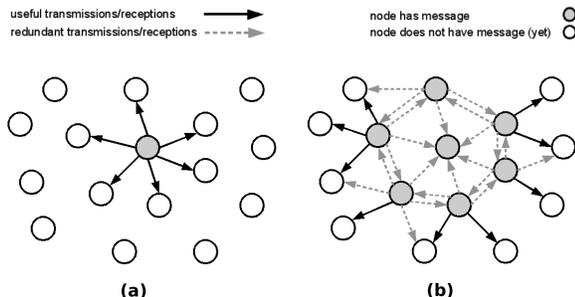


Fig. 2: Broadcast Storm: many frame transmissions and receptions are redundant

nodes, and is harmful in WSNs with scarce energy resources.

- *Contention/Concurrency*: If every node rebroadcasts an incoming broadcast message, transmissions take place more or less simultaneously. All these transmissions from nearby hosts may severely contend with each other. The medium will be locally busy disseminating the broadcasts by all contending neighbors. During this time, the service characteristics of other ongoing transmissions (e.g. unicast point-to-point traffic) are temporarily harmed.
- *Collisions*: Transmission storms will presumably collide with other ongoing transmissions. If collisions occur in the initiation phase of the broadcast, this can lead to starvation of the flood, as broadcasts are unacknowledged and collisions can not be detected by the sender.

A number of solutions to tackle and mitigate the broadcast storm problem have been compared in [11]. There is a broad variety of proposed approaches, ranging from probability-based [10] over location-based [12] to neighbor-designated approaches (calculation of Multi-Point Relays) in [13]. Although the problem is in most cases studied in a MANET (mobile ad hoc network) environment, these investigations are likewise valuable to study broadcasts in (quasi) static wireless sensor networks.

III. THE K-BEST-INSTANTS BROADCAST TECHNIQUE

Without control measures for multi-hop flooding, the WiseMAC broadcast exacerbates the broadcast storm vulnerabilities with the long preambles. Each full-preamble broadcast blocks the channel not only for the immediate neighboring nodes, but also for all nodes in the extended carrier sensing range. After the own rebroadcast, each node is busy with the redundant and costly reception of countless full-preamble broadcasts by each neighbor. Accumulated interference from stations using full-cycle broadcasts frequently leads to collisions with frame transmissions. As using full-cycle preambles is inefficient, the developer of WiseMAC reflects in [14] that *more sophisticated broadcasting and flooding techniques for multi-hop ad-hoc sensor networks and MANETS remain to be designed*. We aim to bridge this gap with the technique being introduced in this section. We propose the (k)-Best-Instants energy-efficient broadcasting technique and integrate it into the WiseMAC protocol. The scheme can however be likewise applied to any other E^2 -MAC protocol based on asynchronous wake intervals.

A. (k)-Best-Instants Broadcast

The (k)-Best-Instants broadcast algorithm optimizes broadcasting in wireless MAC protocols with asynchronous wake intervals. A similar idea has been proposed in [15], although with a different focus and in a different wireless networking environment (IEEE 802.11). The idea of the broadcast algorithm is to calculate a minimum set of instants I in which frames shall be sent such that every neighbor is reached. The key point is the exploitation of similarities in the nodes' wake-up patterns. By limiting the forwarding of the broadcast frames only to the k best of these instants (e.g. $k=2$), the *fan-out* of

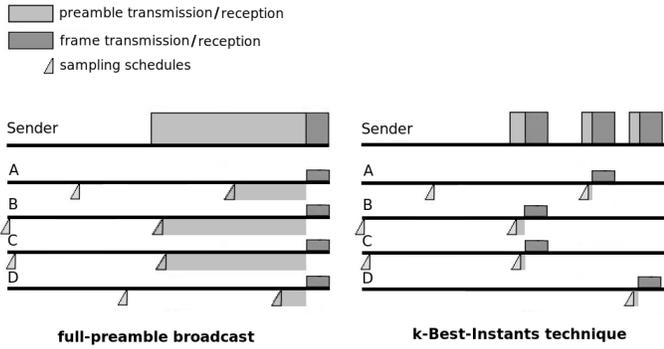


Fig. 3: full-preamble vs. k-Best-Instants broadcast

intermediate nodes can be limited. The algorithm tries to reach as many nodes as possible with as few transmissions as possible. The more nodes that can be reached when transmitting at a certain instant, the higher the priority of this particular instant. When restricting the rebroadcasts to the k best instants, less duplicates occur, and the broadcast storm problem becomes manageable.

The appropriate choice of the parameter k shall be identified experimentally for each network environment and topology. The denser the network, the lower k can be set in order to still flood packets across the entire network with a high delivery ratio. The broadcast technique shares similarities with the probabilistic broadcasting schemes in [10], as the choice of the subset of neighbors is based on the individual neighbor's wake-up patterns, which are in turn initially chosen at random.

B. (k)-Best-Instants in WiseMAC

Figure 3 compares the WiseMAC full-preamble broadcast to the k-Best-Instants approach. The sender has neighbors A, B, C, D , and is aware of their individual schedules. The gray areas illustrate the time that nodes spend in the costly states to transmit and receive preamble and frame. It becomes obvious that calculating the best instants and transmitting the frame with a minimized preamble can be more efficient than using one costly full-cycle preamble.

We tailored the k-Best-Instants procedure to WiseMAC's preamble sampling technique. We exploit so-called *near* wake-ups to cover groups of nodes with *near* wake intervals to save precious transmissions. Wake intervals are considered to be *near* if the difference between their starting points is smaller than the transmission time it takes to transmit a node's preamble and the actual payload, as e.g. nodes B and C in Figure 3. It might pay off to group *near* wake-up intervals of neighboring nodes and transmit a slightly longer preamble, as sending preamble and frame twice would be costlier than grouping these two instants and sending the frame only once. We can express the notion of *near* wake intervals analytically as a function of the basic interval duration T , the bandwidth b and the size of the frames d . Let t_A and t_B be the estimated wake intervals of nodes A, B , respectively. Let the duration of the preambles of the nodes p_A, p_B be calculated according to the WiseMAC equation (1). Their wake intervals t_A, t_B are *near* if the following condition holds (assuming $t_B \geq t_A$):

$$\text{near}(t_A, t_B) := ((t_B - t_A) < \frac{p_A}{2} + \frac{d}{b} + \frac{p_B}{2}) \quad (2)$$

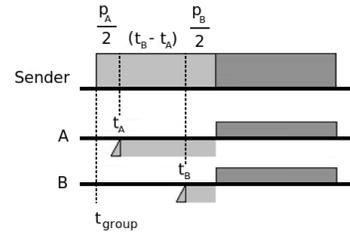


Fig. 4: Preamble composition when grouping near nodes

We omit state transition delays and costs for sleep state in the analytical discussion. In order that their wake-ups are *near* enough for grouping them, transmitting one longer preamble and frame only once is more efficient than sending twice a WiseMAC preamble and a frame, the gap between t_A and t_B shall be not more than two halves preambles $\frac{p_A}{2}, \frac{p_B}{2}$ plus the duration of a packet's transmission $\frac{d}{b}$ away. The wake intervals of nodes B and C nodes depicted in Figure 3 illustrate this property. If $\text{near}(t_A, t_B)$ holds, it makes sense to group these nodes and calculate the preamble that is necessary to reach both of the nodes, and transmit the payload only once.

In case of *near* nodes, the preamble must suffice to alert both nodes. Let them denote as the sooner node A and the later node B with estimated wake-ups t_A, t_B and WiseMAC preambles p_A, p_B . Figure 4 illustrates how the preamble for a group of near nodes is composed. The instant when the transmission of the preamble must be scheduled for is

$$t_{\text{group}} = t_A - \frac{p_A}{2} \quad (3)$$

The duration of the preamble calculates as

$$p_{\text{group}} = \frac{p_A}{2} + (t_B - t_A) + \frac{p_B}{2} \quad (4)$$

Formal proof: we show that grouping wake intervals energetically pays off, exactly when their wake intervals suffice condition (2). The left side of equation (5) denotes transmission costs when transmitting a larger group preamble and one frame. The right side denotes the transmission costs when sending two preambles p_A, p_B and twice the frame $\frac{d}{b}$.

$$\frac{p_A}{2} + (t_B - t_A) + \frac{p_B}{2} + \frac{d}{b} < p_A + p_B + 2 \frac{d}{b} \quad (5)$$

We simplify the in-equation and obtain

$$(t_B - t_A) < \frac{p_A}{2} + \frac{p_B}{2} + \frac{d}{b} \quad (6)$$

which is exactly the property denoted in equation (2) two nodes' wake-ups t_A, t_B have to fulfill. \square

C. Best-Instants Algorithm

Algorithm 1 lists the stepwise calculation of the k-Best-Instants, exploiting the knowledge about node's *near* wake intervals. The algorithm first initializes the set R with the schedules t_1, t_2, \dots, t_n of all its neighbors N_1, N_2, \dots, N_n . Throughout the algorithmic calculation, R denotes the unsorted set of schedules of the yet *uncovered* nodes. The algorithm finishes when all nodes' schedules are *covered*.

In the first loop, the algorithm searches for pairs of *near* nodes for which the property (2) holds. If this is the case, the instant is inserted into I and a group preamble is calculated according to (4) and inserted into P . With no more nodes being *near*

Algorithm 1 Best Instants Algorithm

Input: set of all neighboring nodes' wake-ups $R = \bigcup_{i=1}^n t_i$

Output: set of *Best Instants* I , set of *Preambles* P

$I = \emptyset$

$P = \emptyset$ // initialize the sorted sets I and P

// check for each pair of nodes in R if they are *near* enough

foreach *Pair of Instants* (t_u, t_v) **in** R **do**

 // if near enough, add the sooner instant to set I

if $(t_v - t_u < \frac{P_u}{2} + \frac{d}{b} + \frac{P_v}{2})$ **then**

$t_{group} = \min(t_u - \frac{P_u}{2}, t_v - \frac{P_v}{2})$

$I = I \cup \{t_{group}\}$

 // add corresponding preamble to P, cf. Eq. (4)

$p_{group} = \frac{p_{sooner}}{2} + (t_{later} - t_{sooner}) + \frac{p_{later}}{2}$

$P = P \cup \{p_{group}\}$

 // remove covered instants from the *uncovered* set

$R = R \setminus \{t_u, t_v\}$

end

end

// if any node's wake-up in R is left *uncovered*, add it to I

foreach *Instant* t_i **in** R **do**

$I = I \cup \{t_i - \frac{P_i}{2}\}$

$P = P \cup \{P_i\}$

end

return I, P

enough, the algorithm adds the *uncovered* instants left in R to I , along with the corresponding WiseMAC preambles in the subsequent *for*-loop.

The algorithm returns the sorted set I of *covered* instants and the corresponding sorted set P containing the duration of the preambles associated with the instants of I that are needed to alert the nodes in their short preamble sampling intervals. I contains the m ($\leq n$) minimal set of instants t_1, \dots, t_m to reach all neighbors. I is sorted and instants by that two *near* nodes can be reached are listed first. The Best-Instants algorithm can be generalized to consider groups of three, four or more *near* nodes. Finding groups of more than two nodes with similar wake-up pattern is however rare and improbable, except for very dense networks. To ensure simplicity of the illustration of the concept, Algorithm 1 only exploits groups of two *near* nodes.

D. Energy Cost Trade-Off

The question whether it pays off to apply the k-Best-Instants algorithm or it is cheaper to use the WiseMAC full preamble broadcast depends on how many neighbors exist in the neighborhood, and how many of them need to receive the message. We can express the trade-off analytically for the single-hop case, when considering one node aiming to broadcast a message to all its n neighbors. The node calculates that it can reach it using the k-Best-Instant with k ($\leq n$) transmissions. Let e_{tx} and e_{recv} denote the power consumption of the nodes in the transmit and receive states, respectively.

If we are dealing with non-cooperative, selfish nodes that only focus on minimizing their particular energy consumption (as e.g. in MANETs), the minimization function only considers

the sender part. It then pays off for the sender to apply the k-Best-Instants technique, if the cost of transmitting all preambles p_1, \dots, p_k and all transmissions at the respective instants t_1, \dots, t_k is less than the respective cost for one WiseMAC full-preamble broadcast. The condition is expressed in equation 7, where the left part of the inequation corresponds to the cost of the k-Best-Instants broadcast, and the right part the cost of the WiseMAC full-preamble broadcast.

$$e_{tx} \cdot \left\{ \sum_{i=1}^k p_i + (k \cdot \frac{d}{b}) \right\} < e_{tx} \cdot \left\{ T + \frac{d}{b} \right\} \quad (7)$$

For the receivers, the best-instants technique pays off in any case, as listening to a full-cycle preamble is costlier than listening to a preamble that only serves to compensate for the individual node's clock drifts.

If nodes cooperatively aim to minimize the overall energy consumption of the network (which is generally the case in sensor networks), we can express the trade-off for the 1-hop broadcast by aggregating transmission and reception costs. With the k-Best-Instants technique, the energy cost c_{kBI} for transmitting and receiving preambles and frame by k neighbors - given no overhearings of each others transmissions - calculates as:

$$c_{kBI} = e_{tx} \cdot \left\{ (k \cdot \frac{d}{b}) + \sum_{i=1}^k p_i \right\} + e_{recv} \cdot \left\{ (k \cdot \frac{d}{b}) + \sum_{i=1}^k \frac{p_i}{2} \right\} \quad (8)$$

With WiseMAC, the expected duration of the preamble received by the neighboring nodes comes to $\frac{T}{2}$, as the node's schedules are assumed to be uniformly distributed over one cycle (c.f. [14]). The respective cost for the transmission and reception using the WiseMAC full-preamble broadcasts $c_{WiseMAC}$ calculates as:

$$c_{WiseMAC} = e_{tx} \cdot \left\{ T + \frac{d}{b} \right\} + e_{recv} \cdot \left\{ n \cdot \left(\frac{T}{2} + \frac{d}{b} \right) \right\} \quad (9)$$

Considering the total energy consumption caused by one broadcast over one hop, the k-Best-Instants technique pays off if the cost is lower than the respective cost of the WiseMAC broadcast:

$$c_{kBI} < c_{WiseMAC} \quad (10)$$

IV. EXPERIMENTAL EVALUATION

We implemented and tested the behaviour of the approaches WiseMAC broadcast and k-Best-Instants broadcast in a network simulator, as well as on a real sensor hardware testbed. In Section IV-A we motivate our simulation model and examine the performance of the WiseMAC full-preamble broadcast and the k-Best-Instants technique in a simulation scenario. Section IV-C then describes the prototype implementation on real sensor nodes and evaluates the performance of both broadcast techniques in a small-scale experiment. We show that by choosing the essential system parameters of the simulation model according to the real-world implementation on the real sensor nodes, the model is indeed adequately calibrated. We validate the model by cross-comparing the results of the small-scale experiment in simulation and real-world experiment, and hence underline that the simulation results do not rely on inappropriate choice of parameters.

path loss coefficient α	3.5
carrier frequency	868 MHz
transmitter power	0.1 mW
SNR threshold	4 dB
sensitivity	-101.2 dBm
sensitivity carrier sensing	-112 dBm
communication range	50 m
carrier sensing range	100 m
supply voltage	3 V
transmit current	5.0 mA
recv current	4.5 mA
sleep current	2.0 mA
recv to transmit	4 ms
transmit to recv	2 ms
sleep to recv	1 ms
recv to sleep	1 ms
baud rate	19'200 bps
bit rate	9'600 bps

TABLE I: calibrated OMNeT simulation parameters

basic interval duration T	500 ms
duty cycle	1%
minimum preamble	5 ms
medium reservation preamble	uniform [0,6] ms
MAC header	104 bit
payload	96 bit

TABLE II: WiseMAC simulation parameters

A. Simulation Experiment

We implemented the WiseMAC protocol [2] in the OMNeT++ Network Simulator [16] using the Mobility Framework [17], which supports simulations of wireless ad hoc and mobile networks. The framework calculates SNR (Signal-to-Noise) ratios according to a free space propagation model basing on the equation for received power P_r of a node at distance d as

$$P_r(d) = \frac{P_t \lambda^2}{(4\pi)^2 d^\alpha} \quad (11)$$

where P_t is the transmitted signal power, λ the wavelength of the signal and α the path loss coefficient.

We modeled the power consumption of the sensor nodes with a state transition model with respect to the time spent in three 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. We calibrated the energy model with energy consumption values and state transition delays evaluated in a preliminary study with the WiseMAC prototype on the ESB nodes (c.f. Section IV-C). Furthermore, state transition delays are incorporated to model transceiver switches. The essential parameters of the environment and the WiseMAC implementation are listed in Tables I and II.

We simulated 36 nodes arranged in a grid topology on an area of 175m x 175m (c.f. Figure 5). The source node S in the upper left corner starts flooding the network with data packets of 200 bits size (including MAC and routing header) during 1h with a constant rate (CBR) during each measurement run. The flooding mechanism is straightforward: Packets carry monotonically increasing sequence numbers which are stored

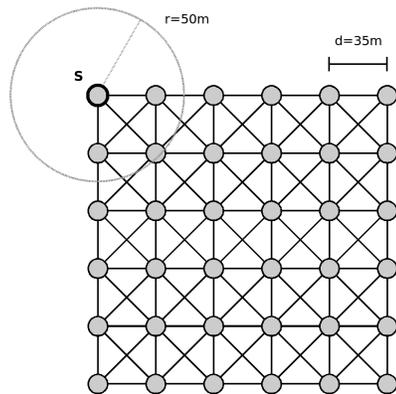


Fig. 5: 6x6 nodes grid topology

in each node upon reception. If the packet is received for the first time, the packets are rebroadcast. Duplicates are not rebroadcast but discarded.

Random Assessment Delay (RAD): Suppose a source node originates a broadcast. As radio waves propagate with the speed of light, all neighbors receive the packet almost simultaneously. If every neighbor receiving the packet immediately rebroadcasts it, the concurrent transmissions inevitably lead to collisions, even when applying carrier-checks before transmission (e.g. due to concurrent transceiver switches). Many protocols therefore apply some uniform random jitter in-between packet reception and the rebroadcast to omit collisions. [11] refers to this small jitter as *RAD (Random Assessment Delay)*. This mechanism alleviates the impact of collisions in beginning of the flood, but does not solve all aspects of the broadcast storm problem. The redundancy issue of countless duplicate transmissions and receptions remains unsolved. Yet, the technique alleviates the concurrency/contention issue and makes sure that the broadcast flood can spread across the network and reach the majority of the nodes. We integrated a RAD to achieve a reliable network-wide broadcast with the full-preamble WiseMAC approach, which allows for a fair comparison with the k-Best-Instants technique.

Performance Metrics: In the following, we measure three characteristics of the two broadcast schemes WiseMAC full-preamble broadcast and the k-Best-Instants broadcast:

- *Delivery Ratio* is the mean ratio between the number of nodes receiving the packets and the number of nodes that have to be reached in total (=35).
- *Broadcast Delay* of the flooded packets is the time packets are generated in the application layer of the sink node until their reception by the nodes in the grid.
- *Energy Consumption* of all nodes is summed up during the simulation time of 1h according to the energy and transceiver model outlined in Section IV-A.

Figures 6, 7 and 8 illustrate the three characteristics measured with both broadcasting techniques using 50 simulation runs.

B. Performance Results

We identified suitable values for the parameters k and *RAD* in a preliminary investigation that led to similar delivery ratios for both the k-Best-Instants broadcast and the WiseMAC full preamble broadcast.

With the k-Best-Instant technique, we found that $k = 2$ led

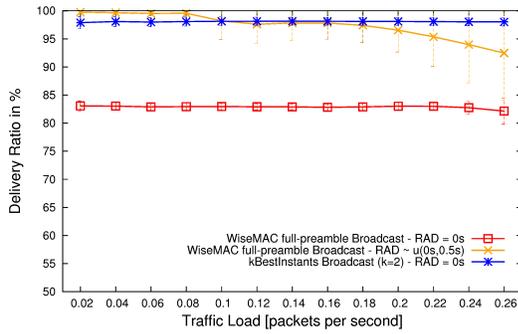


Fig. 6: Delivery ratio vs. traffic load

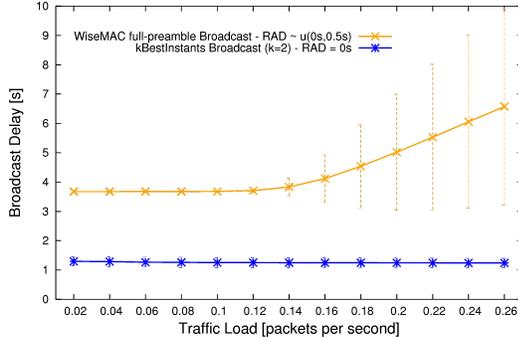


Fig. 7: Delay vs. traffic load

to high ($\geq 97\%$) delivery ratios. Adding a RAD did not further increase the delivery ratio. For the WiseMAC full-preamble broadcast, the introduction of a RAD is inevitable. Setting a value of $RAD \sim u(0, 0.5s)$ proved to deliver a similarly high delivery ratio as the k-Best-Instants broadcast with $k = 2$ (cf. Figure 6). We further focus on the service characteristics of these two approaches, as they both implement a network-wide broadcast with a similar reliability, which allows for a fair comparison. Figure 6 illustrates the delivery ratios versus the traffic load generated by the sink node for the abovementioned approaches. With the WiseMAC full-preamble broadcast, the introduction of RAD led to a significant improvement in the delivery ratio of roughly 15%. Obviously, the original WiseMAC technique suffers from the effects of interference and congestion with increasing traffic load, starting at rate 0.2, while the k-Best-Instants technique still exhibits the same delivery ratio. As the WiseMAC full-preamble broadcast prepends preambles of full cycles T to each frame, each transmission blocks the channel for a large area. Stations aiming to rebroadcast packets but never winning the channel competition may discard incoming packets due to buffer overflows. These mechanisms explain the deterioration in the delivery ratio with increasing packet rate. As the k-Best-Instants technique has a smaller per-packet overhead with respect to the size of the preamble, congestion effects do not yet occur with the examined traffic rates, but they nevertheless decrease the delivery ratio at a higher load level too.

Figure 7 illustrates the end-to-end delay. The WiseMAC full-preamble broadcast needs more time to spread across the network, as the long preambles both take long to be transmitted and likewise block the network for other transmissions. With the WiseMAC full-preamble broadcast, congestion effects lead to a sudden increase in the end-to-end delay with increasing

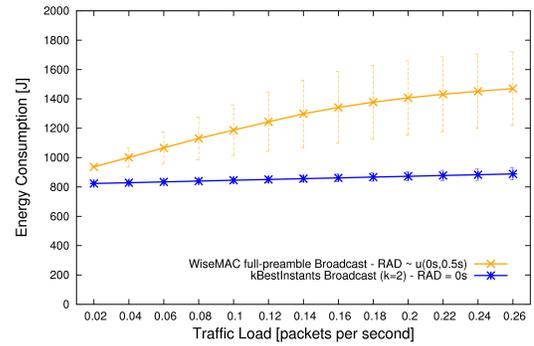


Fig. 8: Overall energy consumption vs. traffic load

traffic rate, starting at 0.14 packets per second. With the k-Best-Instants broadcast, the average delay is much lower, and congestion effects can not be observed for the examined traffic rates. As the technique does not waste time and energy on long preambles, packets distribute quickly across the network. The broadcast storm vulnerability of concurrent medium access is reduced by the fact that each node most probably chooses different receiver nodes than its neighbors. If every node chooses $k=2$ instants to rebroadcast a packet, and only intends to reach a subset of its neighbors, chances are high that transmissions only rarely collide, which renders the introduction of a RAD obsolete.

Figure 8 exhibits an increasing energy consumption with linearly increasing traffic load for both broadcasting techniques. The major improvement of the k-Best-Instants broadcast technique becomes apparent with respect to the energy efficiency. While maintaining a similar delivery ratio, the k-Best-Instants technique saved up to 40% energy, compared to the WiseMAC full-preamble broadcast that applies a $RAD \sim u(0, 0.5s)$. Thanks to the fact that with the k-Best-Instants broadcast only short preambles are prepended to the frames, it is likely that transmissions fall in-between the schedules of untargeted nodes. Hence, nodes do not spend much energy on overhearing long preambles.

C. Model Validation on Real Sensors

Simulating performance improvements in protocols for wireless ad-hoc or sensor networks is undoubtedly easier and more convenient than realizing and proving them on hardware prototypes, although this final step often reveals important effects and difficulties that are not obvious from the simulation perspective. The credibility of simulation experiments has therefore recently been questioned. Several studies underlined the lack of rigor in the application of simulation tools. Inadequate simulation models and improper data analysis have been shown in [19] to produce inconsistent or misleading results. [20] similarly criticizes the lack of realism of simulation studies, and concludes that *model validation* is indispensable in simulation studies of communication protocols. We hence examined the feasibility and real-world behavior of our broadcast mechanism by porting WiseMAC and both broadcast mechanisms to the Embedded Sensor Boards (ESB) [21], a sensor node developed at Freie Universität Berlin [22]. ESB's are equipped with a TR1001 radio transceiver [23]. The parameters of the simulation model listed in Tables I and II were chosen according to the protocol implementation on the ESB nodes. In this Section, we describe the implementation of

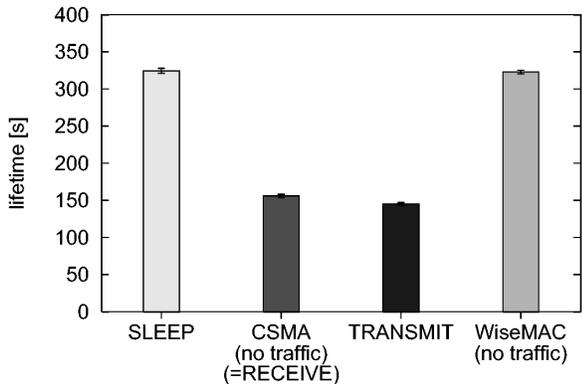


Fig. 9: Lifetimes of ESB nodes different states [25]

WiseMAC and both broadcast techniques on real-world sensor hardware and illustrate our measurement methodology. In a small experiment conducted in both simulation and real-world implementation, we validate the simulation model applied in Section IV-A.

WiseMAC on Embedded Sensor Boards: We implemented the WiseMAC protocol [2] on the ESB nodes with the same parameters as listed in Table II. The WiseMAC prototype implementation achieved a very low duty cycle of 1% for $T = 500ms$. We applied a well-tested and established methodology to investigate the energy consumption of ESB nodes via lifetime measurements. The approach was proposed in [24] and several other studies. It consists in charging GoldCap capacitors and measure the time a node can live on a certain charge. This allows comparing the ESB node’s energy consumptions in different operation modes and to quantify the efficiency gains of the E^2 -MAC protocol and both broadcast techniques. For each measurement run, we charged a previously discharged capacitor for $t_{charge} = 120s$ at 5.5 V. We observed the supply voltage of the capacitor with a customary multimeter. After t_{charge} , we disconnected the capacitor from the mains adapter. Sourcing the ESB node with energy slowly discharges the capacitor, and the voltage on the capacitor continuously decreases. We measured the time until the voltage drops below 3 V as the node’s lifetime (cf. [25] for more details on the evaluation methodology).

By applying the GoldCap methodology, we obtained robust and stable results with low variance. Figure 9 depicts the results of a preliminary lifetime study of an ESB node in different states when applying the GoldCap methodology. It becomes clear that WiseMAC with its low duty cycles is quite energy-efficient. The third bar illustrates the lifetime of an ESB node running simple CSMA, which keeps the transceiver permanently in the receive state, and the fourth bar the most costly transmit state. According to these lifetime measurements, we concluded that the ratio of the power consumption between sleep and receive and transmit on the ESB nodes is approximately 1 : 2.25 : 2.5. This ratio has been applied in the simulation of Section IV-A.

Model Parameters: We chose suitable state transition delays that realistically reflect all steps of the medium access mechanism (e.g writing status bits to the transceiver interface, tuning the radio). Because of implementation specific issues, switching from receive to transmit approximately takes 4 ms and transmit to receive roughly 2ms. The bit rate is 9’600

bps and the baud rate 19’200 bps, as the nodes apply the Manchester encoding to ensure more reliable communication and avert bit-errors. These values were integrated into the simulation model and are listed in Table II. The parameters reflect the actual implementation on the ESB nodes quite realistically.

The simple experiment setup is outlined in Figure 10. Node 1 broadcasts frames to its three neighboring nodes 2, 3 and 4. By increasing the rate of broadcasts, we analyze how the packet rate impacts on the power consumption. We run the experiments with both broadcasting techniques, the WiseMAC full-preamble broadcast and the k-Best-Instants technique with $k = 3$, in order to allow for a fair comparison of both approaches.

D. Experiment Results

Figure 12 depicts the lifetime of the broadcasting node as a function of the rate of packets emitted by node 1 on the ESB prototype platform. Lifetime is measured as the time that the broadcasting node can live on its initial charge of the GoldCap capacitor. In Figure 13, we applied the same concept of the GoldCap-capacitor-based lifetime methodology in the OMNeT++ simulator by measuring the time until the broadcasting node 1 depletes of an initial energy endowment (20 Joules). Note that the absolute values of the lifetimes are of no particular importance. As we could not assess the absolute value of the charge of the GoldCap capacitor, the absolute values of the lifetime curves of Figure 12 and Figure 13 can not be compared. We only focus on the slope of the curves and the difference between the two approaches WiseMAC full-preamble broadcast and k-Best-Instants broadcast.

As expected, lifetime decreases linearly with increasing traffic rate. In both figures, the k-Best-Instants technique leads to a small performance improvement. The measured performance improvement of the k-Best-Instants technique over the WiseMAC full-preamble broadcast on the ESB prototype does not exceed 5% for any measured traffic rate. The results on the simulator are similar, yet a gap clearly becomes apparent. As expected, the gap between both techniques becomes larger with increasing broadcasting rate. The more the broadcast is used, the more it pays off to use the k-Best-Instants technique. The results of the experiment on the simulator are in the same range, but the gap becomes clearly visible with increasing broadcasting rate.

With the small-scale experiment in this section we validated the simulation model in Section IV-A. As we obtained similar differences between the approaches WiseMAC full-preamble broadcast and k-Best-Instants broadcasts, and lifetime curves with a similar slope, we conclude that our simulation model is well calibrated. Unfortunately, our measurement approach

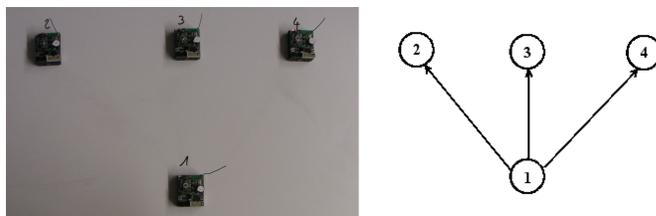


Fig. 10: Broadcasting scenario

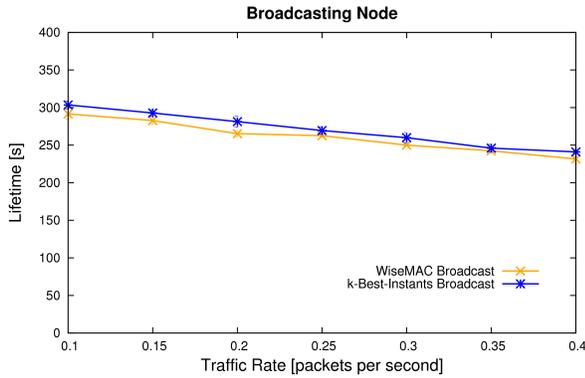


Fig. 11: Broadcasting node on the ESB

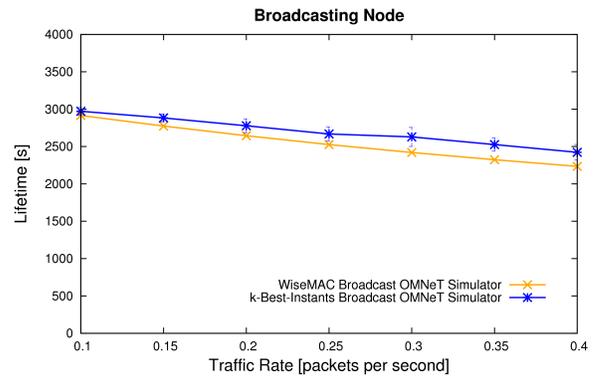


Fig. 12: Broadcasting node in simulation

only allows us to measure one single node's lifetime at a time, and hence the performance improvement of the *k-Best-Instants* technique on the ESB seems to be very limited. One has to keep in mind that the performance gains impact on the sender and on each and every receiver. The reduced the end-to-end delay in multi-hop topologies, and the higher reliability of the *k-Best-Instants* scheme can unfortunately not be easily shown using the small real-world experiment. However, as simulation results have confirmed, the *k-Best-Instants* technique is more suitable for network-wide broadcasting over multiple hops, as it blocks the channel only for short transmissions rather than for long full-cycle preambles. The reason for the only slight performance gain is mainly caused by the size of the experiment. As the main benefit of the *k-Best-Instants* technique lies in saving energy by avoiding overhearing and avoiding costly collisions and retransmissions, the benefit of the *k-Best-Instants* technique increases with the network size.

V. CONCLUSIONS

This paper proposes the energy-efficient broadcasting technique *k-Best-Instants* for wireless MAC protocols based on asynchronous wake intervals. The technique has been integrated into the preamble sampling E^2 -MAC protocol WiseMAC. The necessary prerequisites and assumptions, under which this technique proves to deliver energy-efficiency gains in comparison to existing techniques have been analytically evaluated and formally proved. The *k-Best-Instants* technique can be used in dense networks applying asynchronous energy-efficient MAC protocols whenever multi-hop network-wide flooding is required, as it leads to a lower medium utilization, lower power consumption and fewer interference and congestion problems. Experiments quantify the energy-saving effect in simulation and in practice on a sensor hardware testbed. The technique proves to deliver significantly better results with respect to the delivery ratio, the end-to-end delay and the energy consumption in simulation. Slightly better results with respect to the energy consumption could be shown in practice in a small-scale scenario on real sensor hardware.

REFERENCES

- [1] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks." *ACM Sensys*, 2004.
- [2] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: An ultra low power mac protocol for multihop wireless sensor networks." *ALGOSENSORS*, 2004.
- [3] M. Buettner, V. Y. Gary, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks." *ACM SenSys*, 2006.
- [4] S. Liu, K.-W. Fan, and P. Sinha, "Cmac: An energy efficient mac layer protocol using convergent packet forwarding for wireless sensor networks," in *4th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07*, 2007.
- [5] D. Johnson, D. Maltz, and J. Broch, *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*.
- [6] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *2nd IEEE Workshop on Mobile Computing Systems and Applications, WMCSA '99, February 25-26, 1999*, 1999.
- [7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," in *IEEE/ACM Transactions on Networking*, 2003.
- [8] K. Langendoen, "Medium access control in wireless sensor networks," in *Medium Access Control in Wireless Networks, Volume II: Practice and Standards*. Nova Science Publishers, 2007.
- [9] I. Demirkol, C. Ersoy, and F. Alagoz, "Mac protocols for wireless sensor networks: A survey," *IEEE Communications Magazine*, 2006.
- [10] S.-Y. Ni, Y.-C. Tseng, C. Y.-S., and S. J.-P., "The broadcast storm problem in mobile ad hoc networks?" *ACM MOBICOM*, 1999.
- [11] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *ACM MOBIHOC*, 2002.
- [12] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli, "Optimized stateless broadcasting in wireless multi-hop networks," in *INFOCOM*, 2006.
- [13] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9*. Washington, DC, USA: IEEE Computer Society, 2002, p. 298.
- [14] A. El-Hoiydi, "Energy Efficient Medium Access Control for Wireless Sensor Networks." PhD Thesis, École Polytechnique Fédérale de Lausanne, 2005.
- [15] P. Hurni, T. Braun, and L. Feeney, "Simulation and evaluation of unsynchronized power saving mechanisms in wireless ad hoc networks." *Intl. Conf. on Wired/Wireless Internet Communications (WWIC)*, 2006.
- [16] A. Varga, "The omnet++ discrete event simulation system." *European Simulation Multiconference*, 2001, <http://www.omnetpp.org>.
- [17] 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>.
- [18] L. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment." *IEEE INFOCOM*, 2001.
- [19] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: the incredibles," *SIGMOBILE Mobile Computer Communications*.
- [20] T. R. Andel and A. Yasinsac, "On the credibility of manet simulations," *IEEE Computer*, 2006.
- [21] J. Schiller, A. Liers, H. Ritter, R. Winter, and T. Voigt, "Scatterweb - low power sensor nodes and energy aware routing." *38th Annual Hawaii International Conference on System Sciences*, 2005.
- [22] J. Schiller, A. Liers, and H. Ritter, "Scatterweb: A wireless sensor network platform for research and teaching," *Computer Communications*, 2005.
- [23] RF Monolithics, "Datasheet for the tr1001 868.35 mhz hybrid transceiver." <http://www.rfm.com/products/data/TR1001.pdf>.
- [24] H. Ritter, J. Schiller, T. Voigt, A. Dunkels, and J. Alonso, "Experimental evaluation of lifetime bounds for wireless sensor networks." *European Workshop on Wireless Sensor Networks (EWSN)*, 2005.
- [25] P. Hurni and T. Braun, "Calibrating wireless sensor network simulation models with real-world experiments," in *IFIP NETWORKING, May 11 - 15, Aachen, Germany*, 2009.