BEAM: A Burst-Aware Energy-Efficient Adaptive MAC Protocol for Wireless Sensor Networks

Markus Anwander, Gerald Wagenknecht, Torsten Braun, Kirsten Dolfus
Institute for Computer Science and Applied Mathematics
University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
anwander|wagen|braun|dolfus@iam.unibe.ch

***Abstract*—Low latency for packet delivery, high throughput, good reactivity, and energy-efficient operation are key challenges that MAC protocols for Wireless Sensor Networks (WSNs) have to meet. Since traffic patterns as well as network load may change during network lifetime, adaptability of the protocol stack, e.g. in terms of duty cycling, and the integration of reliable transport mechanisms are mandatory. So far, given that optimizations for energy-efficiency and performance parameters are contradicting, most MAC protocols proposed have concentrated on either one or the other. In order to close this gap, we designed BEAM (Burst-aware Energy-efficient Adaptive MAC).**

BEAM uses a new adaptive duty cycle mechanism, which reacts quickly to changing traffic loads and traffic patterns. The integration of explicit acknowledgments allows for supporting hop-to-hop reliability, lowering the overall energy expenditures of the network for end-to-end transmissions. The protocol itself is IEEE 802.15.4 conform and can easily be used for energy-efficient TCP/IP networking. As a result, BEAM performs well in entirely different environments and application scenarios, making it an all-purpose MAC protocol for WSNs.

I. Introduction

With the growing maturity of wireless sensor networks (WSNs) concerning both hardware and software, the attractiveness of their flexible deployment scheme enters the focus of applications from a variety of domains. Battery-powered devices, possibly in large quantities, that autonomously operate and communicate are beneficial for industrial scenarios as well as for surveillance tasks driven by a specific research question.

The flexibility of these networks comes however at the price of high demands concerning the implementation of the protocol stack: In order to grant a long network lifetime, energy-efficiency has to be a primary concern at all layers of the stack. Data aggregation techniques, energy-aware routing, adaptive duty cycling and transmission power control are popular techniques to comply with this request. Especially the MAC and the physical layer bear a high potential to cut energy expenditure. MAC layer protocols therefore implement means for collision and idle listening avoidance, and aim to reduce protocol overhead. The choice of the used hardware platform, in particular of the radio transceiver, impacts the energy consumption. Packet-oriented radios such as the CC2420 [1] have a better energy per byte ratio than byte-oriented or bit-oriented radios (e.g. CC1000 or TR1000 radio transceivers), making them the better choice for WSNs.

From an application perspective, energy consumption, hence network lifetime, is only one parameter that needs to be ensured. Likewise, performance parameters such as throughput, network reactivity, and latency throughout application lifetime are of equal relevance. During deployment time, network load is typically subject to change, with phases of burst transfer of data (e.g. due to reprogramming), phases of sporadical transmissions with high load (e.g. during event-detection) but also times of low load. Accordingly, only adaptive protocols that provide appropriate reactivity to the current network state are able to fulfill application demands.

In this paper, we introduce the Burst-Aware Energy-Efficient Adaptive MAC (BEAM) protocol, an adaptive, energy-efficient protocol capable to support applications throughout the deployment. We develop our protocol to a platform using the CC2420 radio transceiver, which implements the physical layer of the IEEE 802.15.4 standard [2], and design BEAM in a standard conform way. Adaptive duty-cycling to enable a quick reaction to varying traffic load and pattern, hop-to-hop reliability via positive acknowledgments and a low overhead are the key concepts fused in BEAM in order to balance energy efficiency and network performance.

The remainder of this paper is structured as follows. After reviewing state-of-the-art energy-efficient MAC protocols in Section II, Section III presents design goals and choices for protocol development. BEAM, our burst-enabled, energy-efficient and adaptive MAC protocol is then described in detail in Section IV. A thorough evaluation of BEAM in the OMNeT++ simulator and discussion of the obtained results is presented in Section V. Section VI concludes the paper.

II. Related Work

Contention-based MAC protocols circumvent idle listening, a major source of unattended energy consumption with duty cycling. A number of these contention-based duty cycle MAC protocols have been proposed in the past. They can be discriminated into synchronous and asynchronous protocols. S-MAC [3], RMAC [4], and PRMAC [5] are typical synchronous MAC protocols, which synchronize their duty cycles with neighbor nodes. In contrast, asynchronous MAC protocols such as B-MAC [6], X-MAC [7], WiseMAC [8], MaxMac [9], RI-MAC [10], and Koala [11] allow nodes to operate with their own independent duty cycles.

S-MAC is based on IEEE 802.11, uses RTS/CTS/ACK packets for medium access control and implements a duty cycle control mechanism. In **R-MAC** the RTS/CTS mechanism is replaced by a Pioneer frame, which is forwarded over multiple hops to inform nodes along the route when they have to wake up to receive data frames. **PRMAC** optimizes RMAC using cross-layer services to enable multiple packet transmission in a flow. In general, synchronous MAC protocols are well suited for steady network conditions. A strong disadvantage is that such protocols have problems to handle bursty traffic or variable load due to a lack of flexibility.

B-MAC, as a representative of an asynchronous protocol, relies on adaptive preamble sampling to reduce idle listening. Each node wakes up according to its own duty cycle and checks whether there is any preamble signal. The preamble indicates the intention of the sender to transmit a packet and lasts longer than a receiver's sleep interval. Unfortunately, a node may wake up due to a preamble and stay awake unnecessarily, because the data frames are destined for other nodes. **X-MAC**, the protocol which has the most resemblance with BEAM, solves this problem by introducing short preambles that contain the destination address. Nodes not involved in the communication may then go to sleep mode immediately, while the destination node stays awake and returns an early ACK. Furthermore, it incorporates an algorithm to adapt the protocol to variable traffic loads as well as patterns and to closely approximate the optimal duty cycles. X-MAC [12] supports packet-oriented radios and has been implemented on real sensor nodes using the Contiki operating system [13]. B-MAC and X-MAC are very energy-efficient for light traffic loads, but fail to handle bursty traffic due to the chosen adaptation algorithm. **WiseMAC** works similar to B-MAC. To reduce energy consumption, the length of the preamble is variable and calculated based on the duty cycles of its neighbors which it progressively learns. It hence synchronizes implicitly with those neighbors to which it is communicating on a regular basis, although their sleep and wakeup cycles are independent and asynchronous. **MaxMAC** has been designed to minimize latency and maximize throughput in case of temporarily high load. The protocol adaptively switches to CSMA-like behavior when certain load thresholds are exceeded, and switches back to preamble-sampling when the load is low again. A disadvantage of MaxMAC, B-MAC, and WiseMAC is that they do not work on packet-oriented radios such as BEAM or X-MAC, which are more energy-efficient radios than bit- and byte-oriented radios. **RI-MAC** and **Koala** both propose a receiver initiated mechanism, which is called Low Power Probing. Koala is designed for downloading bulk data from all sensor nodes to a certain sink node. In RI-MAC, every node uses beacons to indicate that it is awake and ready to receive data. Thus, every transmitted beacon creates additional interference during periods with no or less traffic.

III. Design Goals and Design Choices

The development of BEAM has been motivated by the lack of availability of an energy-efficient, low latency protocol that is able to cope with different network load and traffic patterns. Furthermore, scalability, reliability support and high data throughput even under changing network conditions along with the ability to recover from a certain link error rate have been in the focus of design considerations. The protocol overhead should be low with no additional bytes in the header and without incurring extra management traffic.

Energy-efficiency is already targeted at the physical layer: BEAM supports packet-oriented radios such as the CC2420 radio transceiver [1], which implements the physical layer of the IEEE 802.15.4 standard [2]. They consume significantly less energy for transmitting, receiving/idle listening, and sleeping as byte- and bit-oriented radio transceivers do. To handle different traffic loads and patterns, the duty cycles have to be adaptive. This request resulted in using asynchronous duty cycles, a choice which prevents further management traffic such as synchronization or beacon messages. Moreover, synchronization mechanisms would increase protocol complexity, thus memory consumption for protocol implementation, which should in general be kept low on sensor nodes.

In fact, X-MAC is a protocol that meets all these requirements as well. However, X-MAC works only properly with light traffic load and low link error rates and runs short in handling bidirectional traffic. Problems occur when end-to-end reliability on the transport layer is aimed for. BEAM handles this deficiency by extending the duty cycle adaptation algorithm, introducing positive acknowledgments, caching MAC frames and including hop-to-hop retransmissions. While the first extension supplies adaptability to diverse transmission patterns (continuous data streams, data bursts and event-based traffic), the latter incorporates reliability at the MAC layer.

IV. Protocol Description

The BEAM protocol comprises two different operational modes to optimize receiver sleep time dependent on the payload size. Both modes rely on positive acknowledgements of MAC frames upon reception. In this section, these modes are presented along with the adaptation algorithm of listen and transmission cycles which allows for traffic awareness, a data frame aggregation scheme to minimize transmitted bytes, and a mechanism to provide hop-to-hop reliability.

A. Basic Operation

BEAM is a contention-based MAC protocol based on asynchronous duty cycles. Its basic operation is shown in Fig. 1. The sender is transmitting periodically short preamble frames including the payload **(1)**. The receiver wakes up and listens to the channel **(2)**. Since the destination address is stored in the short preamble frame header, it can recognize that it is the intended receiver and sends a data ACK frame **(3)**. The sender receives the data ACK frame, stops transmitting the data frame, and goes to sleep **(4)**. If the sender does not receive the data ACK (e.g. due to interferences) it continues transmitting the short preamble frames including the payload. In the optimal case, two transmissions for each transmitted MAC frame are needed. Clearly, longer duty cycles (as long

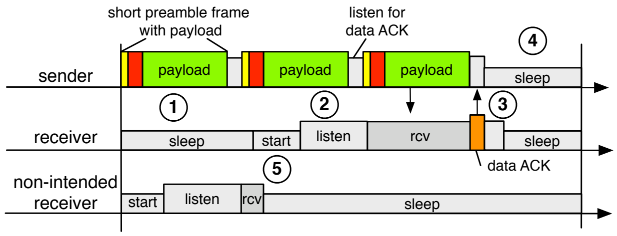


Fig. 1. BEAM using Short Preambles with Payload.

as the data frame) of the receiver affect the energy-efficiency negatively: The non-intended destinations then have to listen longer to conceive that the data is not destined for them **(5)**. Due to the integration of positive acknowledgments, BEAM is reliable in terms of a high probability of a successful delivery of a MAC frame from the sender to the receiver.

B. Short Preamble Frame without Payload

With a reduced overhearing time, the non-intended receivers can go to sleep earlier. As shown in Fig. 2, in the second protocol mode the short preamble frame is sent only with the MAC header and without the payload. The sender is periodically transmitting short preamble frames to indicate the transmission intention and waits for an early ACK frame **(1)**. The receiver wakes up, listens to the channel, and with the destination address stored in the short preamble frame header, it recognizes that it is the intended receiver. Then, it transmits back the early ACK frame **(2)**. Upon reception of the early ACK frame, the transmitting node is aware of the intended receiver being awake, and transmits the data frame **(3)**. Successful transmission is acknowledged by the receiver via a data ACK frame **(4)**. Non-intended receivers can thus go to sleep much earlier as in the basic operation mode **(5)**. The main disadvantage here is that at least four transmissions are required in the optimal case. Furthermore, it is less robust and more complex compared to the basic operation mode.

Dependent on the payload size, BEAM switches between the two modes: We found that when using the basic operation scheme with a small payload of up to 40 bytes, the energy consumption is lower than using short preamble mode. This issue will be discussed in more detail in Section V-C.

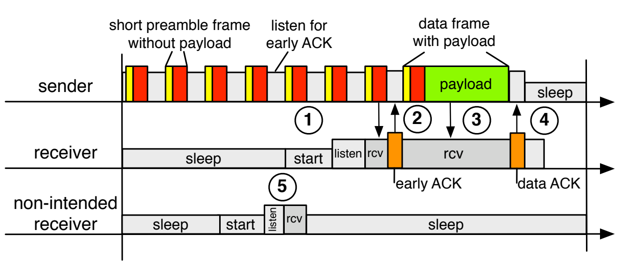


Fig. 2. BEAM using Short Preambles without Payload.

C. Data Frame Aggregation

If there is more than one frame in the buffer to transmit to the same neighbor, these can be aggregated given that there is enough space for them in a single MAC frame. In IEEE 802.15.4, the maximum size is 128 bytes. Aggregation provides a simple means to reduce the number of frames to be transmitted, and as a result the number of collisions and bit errors. The evaluation shows that the advantage given by the reduction of the numbers of frames, is higher than drawbacks by the higher frame error rate that comes with longer frames.

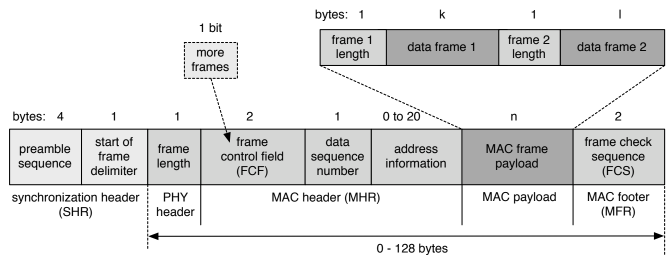


Fig. 3. Data Frame Aggregation.

BEAM's aggregated data frame format is shown in Fig. 3. The whole payload of the MAC frames (data frame) is taken and aggregated into the MAC frame payload of a resulting MAC frame. It is then transmitted to the next hop neighbor. On the receiving side, BEAM splits it back into the original MAC frame payloads according to the frame length and delivers them to the network layer.

D. Listen Cycle Adaptation

In general, traffic in WSNs is neither uniform nor constant during a deployment. For instance, sensors detecting environmental phenomena transmit their data to a sink sporadically or regularly. Such information is normally transmitted in one single packet. A different traffic pattern can be observed when code updates are performed. Starting at the base station, images are sent to a single sensor node. Such traffic is bursty, which means that a large packet (e.g. 500 bytes) is fragmented into a number of small frames, e.g. IEEE 802.15.4 data frames with a maximal length of 128 bytes. The third example for typical traffic is related to event detection. Upon event occurrence in a certain region of the WSN, a number sensor nodes simultaneously react to event detection with issuing notifications towards a base station with possibly minimal delay. Furthermore, the introduction of reliability mechanisms, e.g. hop-to-hop via acknowledgment frames, changes the traffic pattern as well as a nodes' position within a given topology. Since MAC protocols with fixed duty cycles cannot cope with this variation in load and traffic, BEAM incorporates an adaptation algorithm that automatically adapts listen and transmission cycle dependent on current node state.

Our adaptation algorithm is based on the X-MAC adaptation algorithm presented in [7]. There, the probability P_d of receiving a packet within a given time interval $\triangle t$ is calculated. On the receiver's side, the time to awake t_{awake} to receive the next packet is afterwards estimated. While this adaptation works well under steady network conditions, it reacts too slowly in case of severe traffic increase, e.g. due to burst transfer or a number of retransmissions. Even worse, in case of collisions, the calculated probability P_d decreases because packets have not been received. In fact, the receiver can not detect collisions, misinterprets this as low traffic, and adapts the sleep and listen cycles wrongly. Similar misinterpretation can be observed during strong traffic increase. In order to fix this problem, BEAM uses a traffic indicator T_i to inform the receiver whether there are still frames that the sender has to transmit to the receiver. This 1-bit information is added in the frame control field (FCF) of every transmitted frame. The receiver adapts its listen cycle by calculating an earlier time to wake up according to the traffic indicator T_i. Simulations have shown that it is enough to just halve the original time t_{awake} if more frames have to be transmitted. If all frames have been sent, the original time t_{awake} is used.

E. Transmission Cycle Adaptation

Quick traffic increase, a high number of collisions and retransmissions will fill the receiver's buffer. In this situation, a sender should not transmit new frames to the receiver in order to avoid buffer overflow. BEAM uses two bits, referred to as buffer indicator Bi in the following, in the FCF field of the data ACK frame or the early ACK frame to inform the sender about the buffer state of the receiver. The sender then reduces the number of transmission attempts accordingly: If the buffer is empty ($Bi = 0$), the sender transmits the new frame immediately without any delay. If there are only a few frames in the buffer ($Bi = 1$), the sender goes to sleep and schedules the packet in the next listen cycle, while a full buffer ($Bi = 2$) triggers a delay of two slots.

F. Hop-to-Hop Reliability Support

Especially in environments with high bit error rates and high packet loss, it is necessary to ensure reliability, an aspect not discussed in X-MAC [7] [12]. Since X-MAC does not incorporate data acknowledgments, a sender is not aware whether a transmitted frame has successfully arrived at the receiver. Reliability has to be ensured by upper layers such as the transport layer. This has the effect that in environments with high link error rates, X-MAC performs very poorly and results in high packet loss. Thus, an efficient operation with X-MAC is not possible.

In [14] we propose the H2HR (Hop-to-Hop Reliability) protocol, which ensures hop-to-hop reliability. The MAC protocol has to support H2HR by acknowledging successfully received data frames and works as follows: After receiving a data frame, the receiver acknowledges this by sending back a data ACK frame to the sender (shown in Fig. 1 and Fig. 2). The sender turns to sleep mode, otherwise H2HR would retransmit the data frame until it receives the data ACK frame. The receiver hands the received data frame to the upper layers (H2HR/IP), which will then decide whether the frame has to be forwarded. If this is the case, the receiver stays awake and, if the channel is free, it transmits the frame to the next hop with a calculated backoff time. This backoff time is calculated according to the congestion of the destination node. A busy channel will trigger the node to return to its regular sleep cycle and forward the frame during the next awake cycle.

TABLE I
OVERVIEW X-MAC AND BEAM.

	traffic load adapt.	**link layer ACKs**	**data frame aggreg.**	**cycle adapt.**	**H2HR**
X-MAC	yes	no	no	no	no
BEAM	yes	yes	no	no	no
BEAM / FA	yes	yes	yes	no	no
BEAM / FA / CA	yes	yes	yes	yes	no
BEAM / FA / CA / H2HR	yes	yes	yes	yes	yes

V. Evaluation

We implemented BEAM in the OMNeT++ simulator [15]. First, we compared the two operation modes of BEAM (short preamble frames with/without payload) regarding energy consumption and transmission time to find the optimal payload size for switching between both modes. Second, we compared BEAM with the different optimizations (data frame aggregation (FA), listen/transmission cycle adaption (CA), hop-to-hop reliability (H2HR)) with X-MAC using different traffic patterns (data stream, data burst, event detection) regarding packet loss, transmission time, and energy consumption. Table I shows an overview of X-MAC and BEAM with the optimizations.

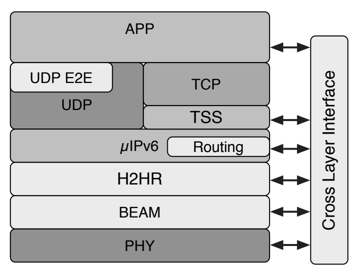
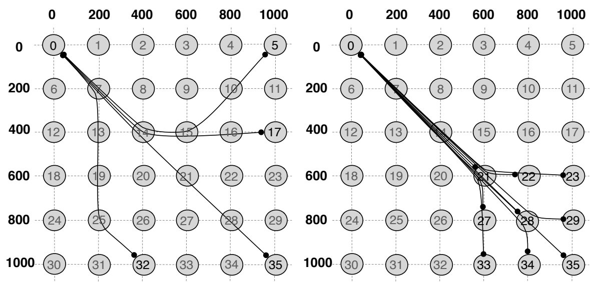


Fig. 4. Protocol Stack including Cross Layer Interface.

A. Protocol Stack

We integrated BEAM into the protocol stack as shown in Fig. 4. On top of BEAM, the H2HR protocol (Section IV-F) has been implemented. As stated above, it guarantees reliable transport of packets to the next hop. Furthermore, H2HR detects and handles congestions and broken links. To connect WSNs with the Internet, we have used the μIPv6 protocol implementation [16] and have enhanced the TCP/IP stack by adding TCP Support for Sensor Nodes (TSS) [17]. This protocol monitors the TCP sequence numbers to detect packet loss, caches packets on intermediate nodes and avoid end-to-end retransmission by resending cached packets. On top of UDP, we have implemented a simple end-to-end reliability protocol (UDP E2E), which is based on negative acknowledgments. All layers may exchange information using a cross layer interface.

B. Simulation Scenarios



(a) Stream & Burst. (b) Event Detection.

Fig. 5. Simulation Scenarios.

Three different scenarios have been used to evaluate BEAM. 36 sensor nodes are arranged in a grid with a distance of 200 meters in a row. The first scenario (data stream) shown in Fig. 5(a) consists of four sources (nodes 5, 17, 32, and 35) sending constant data streams to the sink. Data is transmitted from theses leaves to the root. One to five packets per second are sent per stream, e.g. for transmitting the current temperature to the base station, which equals to one UDP packet with 10 bytes payload. The second scenario (data burst) also shown in Fig. 5(a) is the inversion of the first scenario as data is transmitted from the root to the leaves. One source (node 0) transmits a data burst of 500 bytes to one to four sinks (nodes 5, 17, 32, and 35), e.g. for updating software on the sensor nodes. According to our previous work [14], we are using UDP in combination with UDP E2E as transport protocols with end-to-end reliability. Using TCP as transport protocol does not work with X-MAC. Without BEAM and H2HR a TCP connection cannot be established in case of frequent packet loss. The third scenario (event detection) is shown in Fig. 5(b). Here, an event occurs in the right lower corner of the grid (close to node 28) and the nodes affected by the event send one packet to the sink (node 0). For each scenario, 500 simulation runs of 30 minutes each have been used for evaluation.

In order to gain realistic results even from an evaluation based on simulations, we implemented a radio model according to the CC2420 manual [1] and the Castalia Simulator [18] to calculate the signal to noise ratio (SNR). Table II shows these parameters. We calculated the bit error rate (BER) depending on the SNR and derived from real measurements with the CC2420 radio transceiver. We implemented the CC2420 state machine with the real switching times and energy consumption according to [1] and [18]. Besides the calculated BER, we assume an additional packet error rate of 5% which simulates random noise and interferences with other electronic devices. Furthermore, we assume that the global ring buffer for the packets is limited to 240 bytes.

Due to our results not having a Gaussian distribution, boxplot diagrams are used to visualize the results.

C. Short Preamble Frames with/without Payload

First, we evaluated the energy consumption and the transmission time (in terms of how long a packet takes from source to sink) using short preambles with and without payload. This is done using the first scenario (shown in Fig. 5(a)), where four sources send streams to one sink. We distinguish between the energy consumption of all 36 nodes and the energy consumption of the 15 nodes involved in sending/forwarding traffic.

TABLE II
SIMULATION PARAMETERS.

carrierFrequency	2.4E+9 Hz
bit-rate	250 kbps
sensitivity	-94 dBm
thermalNoise	-110 dBm
transmission power	1mW
modulation	O-QPSK

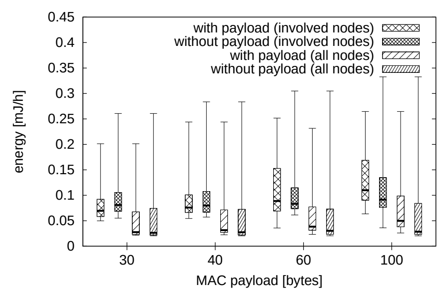


Fig. 6. Energy Consumption using Short Preambles with/without Payload.

As shown in Fig. 6, the energy consumption with small MAC payloads (30 to 40 bytes) is lower using short preambles with payload than using them without payload. With larger MAC payload (40 bytes and more), using short preambles without payload is the better choice to transmit data.

When relying on short preambles without payload, four transmissions are necessary (shown in Fig. 2), which results in a high bit error rate and thus in a higher number or retransmissions. This implies a high energy consumption. If the payload is integrated into the preamble, the number of transmissions may be reduced to two (shown in Fig. 1). Hence, there is a lower probability of collisions, bit errors and retransmissions. With a certain payload size, the preamble becomes so long that the advantage turns into a disadvantage. A longer preamble impacts the non-involved receivers to take a longer (listen) time to find out that they are not the destination. Further, the packet error ratio increases with the length of the packet.

Regarding the transmission time, we get similar results. For a payload of less than 40 bytes using short preambles with payload, the packets take less time from source to sink. Above a payload of 40 bytes using short preambles without payload is more efficient. Thus, BEAM switches between both modes at a payload of 40 bytes.

D. Traffic Pattern: Data Stream

Now, we compare the different BEAM variants (BEAM, BEAM/FA, BEAM/FA/CA, and BEAM/FA/CA/H2HR) with

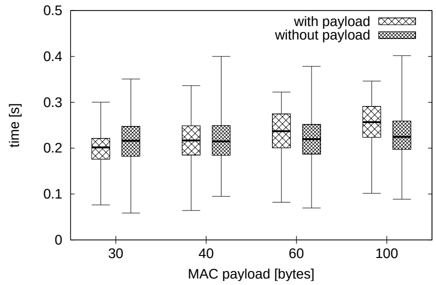


Fig. 7. Transmission Time using Short Preambles with/without Payload.

X-MAC regarding the performance with data streams as shown in Fig. 5(a). The four sources transmit one or four data packets per second to the sink. The packet loss ratio is shown in Fig. 8. When transmitting just 1 packet per second X-MAC has a packet loss ratio from 3% - 10%. BEAM in all variants does not have any packet loss. For a higher transmission rate of four packets per second, X-MAC has a high packet loss ratio (30% - 40%). The basic variant of BEAM experiences packet loss (0% - 15%), but much less than X-MAC. This is caused by the UDP E2E protocol, which stops retransmitting packets due to congestion after a certain time. Data frame aggregation (FA) decreases these values down to 0% - 5%. Especially in the stream scenario with the small packets it happens quite often, that frames can be aggregated and congestion can be avoided. Using additional listen/transmission cycle adaptation (CA) decreases the packet loss ratio to about 1%. With hop-to-hop reliability support (H2HR) the packet loss ratio is 0%.

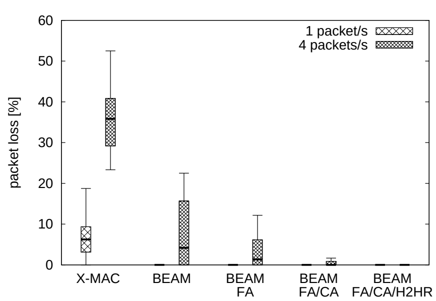


Fig. 8. Data Stream: Packet Loss Ratio.

The average exposure time of a packet is shown in Fig. 9. The scale of the diagram is logarithmic as it depicts the dimension of the discrepancy of the X-MAC performance compared to the BEAM performance better. With a low data rate of 1 packet per second, X-MAC is almost as good as BEAM (0.2s - 0.3s), yet it may happen that a packet takes 15s to reach its destination. These maximum values can be decreased significantly by using BEAM (about 4s), BEAM/FA (about 3.5s) and BEAM/FA/CA (about 2.5s). Using hop-to-hop reliability support, the maximum value is decreased to 0.3s due to avoiding time consuming end-to-end retransmissions. Transmitting 4 packets per second is faster than transmitting 1 packet per second due to a higher number of packets, which implies lower sleep cycle and this implies faster transmission. In the optimal case, 4 packets per second should be transmitted four times faster than 1 packet per second. However, due to packet loss this effect is not achievable.

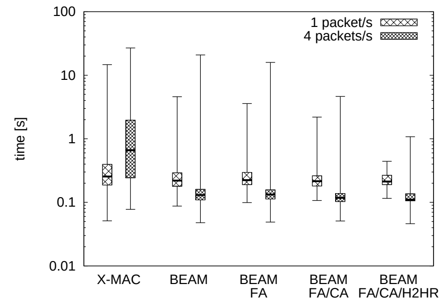


Fig. 9. Data Stream: Packet Exposure Time.

The energy consumption per node and per transmitted byte is shown in Fig. 10. Although X-MAC has the largest packet loss ratio (meaning lowest number of received bytes), it has the highest energy consumption per node and per transmitted byte. The reason is the high number of required end-to-end retransmissions, especially with the high transmission rate of 4 packets per second. BEAM in the basic operation mode performs better and the optimizations to decrease the packet loss ratio does not cost more energy.

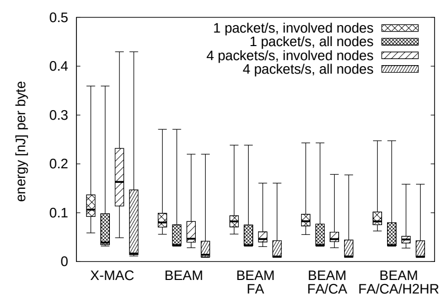


Fig. 10. Data Stream: Energy Consumption per Node and Transmitted Byte.

Fig. 11 compares of the maximum throughput of X-MAC and BEAM with the optimizations. We are using four streams that transmit a number of packets per second to the sink (node 0). The number of transmitted packets is increased from 2 up to 20 per second. At the sink, the number of received packets per second is measured. X-MAC reaches a maximum throughput of about 2.5 packets per second at an input of 4 packets per second. The basic operation mode of BEAM can handle 4 packets per second properly. Afterwards, the output decreases below 2 packets per second. The data frame aggregation optimization improves the performance. The maximum throughput of 6 packets per second is reached at an input of 8-10 packets per second. Then the throughput decreases to a value of below 6 packets per second. The cycle adaptation increases the throughput to a maximum of slightly more than 8 packets per second. The best performance can be reached using the hop-to-hop reliability optimization: Up to an input of approximately 14 packets per second, all packets are delivered successfully. The maximum throughput of 16 packets per second is reached with an input of 18 packets per second.

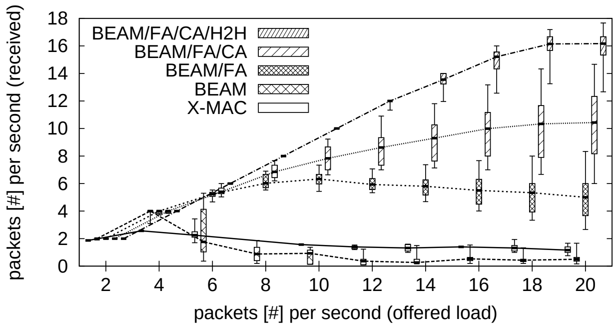


Fig. 11. Data Stream: Throughput.

Concluding the stream scenario, we have shown that BEAM performs better in terms of packet loss, transmission time, energy consumption, and throughput. Furthermore, including the BEAM optimizations, which decreases the packet loss ratio and transmission time significantly, does not cost more energy.

E. Traffic Pattern: Data Burst

The next typical traffic pattern we have analyzed is a data burst. We transmit 500 bytes from the source to one or four sinks using UDP. To ensure end-to-end reliability, our simple UDP E2E protocol has been used. A data packet of 500 bytes is divided into 5 frames, which are transmitted to the sink nodes. Such data bursts stress the network significantly.

The transmission time is shown in Fig. 12. BEAM in the basic operation mode performs slightly better than X-MAC. Specifically, BEAM eliminates the maximum outliers. This is due to the fact that X-MAC only relies on one transmission after receiving the early ACK frame, while BEAM transmits the data frame until it receives a data ACK frame. Data frame aggregation decreases the transmission time since the negative acknowledgments from UDP E2E are aggregated. Furthermore, the utilization of the listen/transmission cycle adaptation decreases the average (median) and maximum transmission time significantly, which proofs its proper work in burst scenarios. Additional hop-to-hop reliability support decreases the transmission time due to avoidance of end-to-end retransmissions.

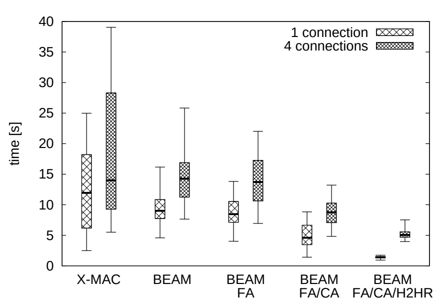


Fig. 12. Data Burst: Transmission Time.

The energy consumption per node and transmitted byte is shown in Fig. 13. The basic operation mode of BEAM performs similarly to X-MAC, but once again has smaller maximum outliers. If we use data frame aggregation, the transmission time is reduced, which implies also a reduction of the energy consumption. The use of listen/transmission cycle adaptation increases the energy consumption significantly. The algorithm reacts to a burst of data by decreasing the duty cycles, which however increases the energy consumption. Once again, H2HR support decreases the energy consumption as end-to-end retransmissions are prevented. In general, the different BEAM optimizations reduce the transmission time. While this results first in a higher energy consumption (e.g. due to decreasing duty cycles), the saved time allows nodes to go to low energy mode much earlier. Overall, this saves more energy than is used for the BEAM optimizations.

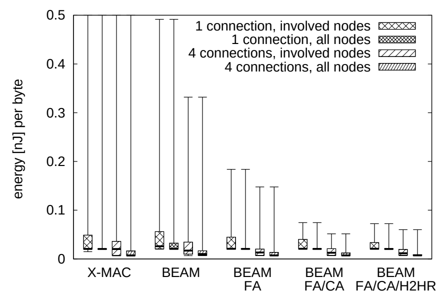


Fig. 13. Data Burst: Energy Consumption per Node and Transmitted Byte.

Concluding the burst scenario it has been shown that BEAM, especially with the proposed optimizations, can handle data burst much better than X-MAC in terms of packet loss ratio, transmission time, and energy consumption.

F. Traffic Pattern: Event Detection

In event detection scenarios, reactivity is the key issue. A sink or base station should be informed about environmental phenomena immediately, with as little delay as possible. In the scenario shown in Fig. 5(b), an event occurs in the right lower corner and 4-9 nodes are affected. The affected nodes send each one packet to the sink to inform it about the event. No end-to-end reliability mechanism such as UDP E2E is incorporated. In Fig. 14, the transmission time is shown (in terms of how long it takes until the sink is informed about the event, meaning arrival time of the first packet).

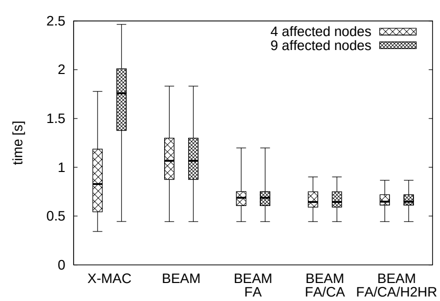


Fig. 14. Event: Transmission Time.

Figure 14 reveals BEAM to be more reactive than X-MAC. In the best cases (minimal outliers), X-MAC performs as well as BEAM. However, in the average (median) and worst (maximal outliers) cases, BEAM reacts faster than X-MAC in case of 9 affected nodes. Using BEAM with the data frame aggregation cuts the number of the maximum outliers since packets are aggregated on nodes 28 and 21, and thus there is no need to transmit them concurrently to the sink. Usage of the listen/transmission cycle adaptation improves the performance by decreasing the sleep cycles. H2HR has only a slight effect as this scenario lacks end-to-end retransmissions.

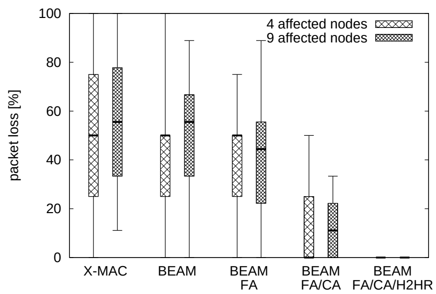


Fig. 15. Event Detection: Packet Loss Ratio.

Fig. 15 shows the packet loss ratio in the event scenario. In this scenario, it may not be important how many packets concerning a detected event reach the sink as one notification may be sufficient. When using X-MAC and BEAM in the basic operation mode it can however happen that no packet reaches the receiver. Thus, an event may not be detected. It may also be the case that it is important to know which nodes detected an event, thus then it is essential to minimize the packet loss. Naturally, aggregating data frames decreases the packet loss only slightly, as these may get lost as well. Using the cycle adaptation improves the situation significantly due to the ability to handle a burst of packets. Only hop-to-hop reliability support ensures that all packets are delivered to the source.

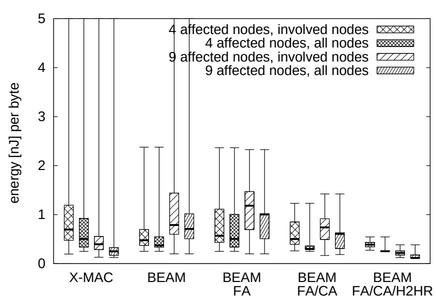


Fig. 16. Event Detection: Energy Consumpt. per Node and Transmitted Byte.

The energy consumption per node and transmitted byte is shown in Fig. 16. The maximum outliers occur, if there is congestion, since then only few packets arrive at the sink. Thus, the relation between consumed energy and the number of successfully received bytes is poor. If all packets arrive at the sink (which is the case for BEAM with the H2HR optimization), the relation between energy and transmitted bytes is much better as congestion is handled well.

Summarizing these results, the event scenario showed that BEAM (with its optimization) performs better than X-MAC in terms of transmission time, packet loss, and energy consumption.

VI. Conclusion

In this paper, we present BEAM, an energy-efficient and hop-to-hop reliability supporting MAC protocol for packet-oriented radio transceivers such as CC2420. We use adaptive duty cycles, which are tuned by our robust adaptation mechanism to handle different traffic loads and traffic patterns. We showed that BEAM performs very well in terms of energy consumption, throughput, transmission time, reactivity, and hop-to-hop reliability with three different traffic patterns (data stream, data burst, and event detection). Thus, BEAM is a robust and high-performing MAC protocol, which can be used in different and varying environments and use cases.

References

[1] CC2420: Datasheet for the Chipcon CC2420 2.4 GHz IEEE 802.15.4 compliant RF Transceiver, *Online, Mar'07*.

[2] IEEE 802.15.4: Wireless Medium Access control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), *IEEE Computer Society, Feb'06*.

[3] W. Ye, J. Heidemann, D. Estrin: Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks, *IEEE/ACM Transactions on Networking 12(3):493-506, Jun'04*.

[4] S. Du, A. Kumar Saha, D. B. Johnson: RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks, *INFOCOM'07, 1478-1486, Anchorage, AK, USA, May'07*.

[5] T. Canli, A. Khokhar: PRMAC: Pipelined Routing Enhanced MAC Protocol for Wireless Sensor Networks, *ICC'09, Dresden, Germany, Jun'09*.

[6] J. Polastre, J. Hill, D. Culler: Versatile Low Power Media Access for Wireless Sensor Networks, *SenSys'04, 95-107, Baltimore, USA, Nov'04*.

[7] M. Buettner, G. V. Yee, E. Anderson, R. Han: X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks, *SenSys'06, 307-320, Boulder, CO, USA, Oct/Nov'06.*.

[8] A. El-Hoiydi, J.-D. Decotignie: WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks, *ISCC'04, 244-251, Alexandria, Egypt, Jun'04*.

[9] P. Hurni, T. Braun: MaxMAC: a Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks, *EWSN'10, Coimbra, Portugal, Feb'10*.

[10] Y. Sun, O. Gurewitz, D. B. Johnson: RI-MAC: A Receiver Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Load, *SenSys'08, 1-14, Raleigh, NC, USA, Nov'08*.

[11] R. Musaloiu-Elefteri, C. M. Liang, A. Terzis: Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks, *IPSN'08, 421-432, St. Louis, MO, USA, Apr'08*.

[12] P. Suarez, C. G. Renmarker, A. Dunkels, T. Voigt: Increasing ZigBee Network Lifetime with X-MAC, *REALWSN'08, 26-30, Glasgow, United Kingdom, Apr'08*.

[13] A. Dunkels, B. Grönvall, T. Voigt: Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors, *LCN'04, 455-462, Tampa, FL, USA, Nov'04*.

[14] G. Wagenknecht, M. Anwander, T. Braun: Hop-to-Hop Reliability in IP-based Wireless Sensor Networks - a Cross Layer Approach, *WWIC'09, 61-72, Enschede, The Netherlands, May'09*.

[15] OMNeT++: Discrete Event Simulation System, *http://www.omnetpp.org*.

[16] M. Durvy, J. Abeille, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, A. Dunkels: Making Sensor Networks IPv6 Ready, *SenSys'08, 421-422, Raleigh, NC, USA, Nov'08*.

[17] T. Braun, T. Voigt, A. Dunkels: TCP Support for Sensor Networks, *WONS'07, 162-169, Obergurgl, Austria, Jan'07*.

[18] H. N. Pham, D. Pediaditakis, A. Boulis: From Simulation to Real Deployments in WSN and Back. *WoWMoM'07, Helsinki, Finland, Jun'07*.