Receiver-Initiated Medium Access Control Protocols for Wireless Sensor Networks

Xenofon Fafoutis[a,*], Alessio Di Mauro[a], Madava D. Vithanage[a], Nicola Dragoni[a]

[a] *DTU Compute - Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, 2800 Denmark*

Abstract

One of the fundamental building blocks of a Wireless Sensor Network (WSN) is the Medium Access Control (MAC) protocol, that part of the system governing when and how two independent neighboring nodes activate their respective transceivers to directly interact. Historically, data exchange has always been initiated by the node willing to relay data, i.e. the sender. However, the Receiver-Initiated paradigm introduced by Lin *et al.* in 2004 with RICER and made popular by Sun *et al.* in 2008 with RI-MAC [1], has spawned a whole new stream of research, yielding tens of new MAC protocols. Within such paradigm, the *receiver* is the one in charge of starting a direct communication with an eligible sender. This allows for new useful properties to be satisfied, novel schemes to be introduced and new challenges to be tackled. In this paper we present a survey comprising of all the MAC protocols released since the year 2004 that fall under the receiver-initiated category. In particular, keeping in mind the key challenges that receiver-initiated MAC protocols are meant to deal with, we analyze and discuss the different protocols according to common features and design goals. The aim of this paper is to provide a comprehensive and self-contained introduction to the fundamentals of the receiver-initiated paradigm, providing newcomers with a quick-start guide on the state of the art of this field and a palette of options, essential for implementing applications or designing new protocols.

Keywords: Distributed Embedded Systems, Medium Access Control, Wireless Sensor Networks, Receiver Initiated Protocols

*Corresponding author

Email addresses: xefa@dtu.dk (Xenofon Fafoutis), adma@dtu.dk (Alessio Di Mauro), s090912@student.dtu.dk (Madava D. Vithanage), ndra@dtu.dk (Nicola Dragoni)

1. Introduction

The Medium Access Control (MAC) layer plays a key role in wireless sensor networks. It is primarily responsible for the establishment of communication links between nodes, that are vital to form the network infrastructure. The MAC scheme then regulates the access to the shared wireless channel by multiple nodes. In contrast to conventional networks where Quality of Service (QoS) and bandwidth efficiency are considered the main priority, energy efficiency remains the primary objective of Wireless Sensor Networks (WSNs), rendering traditional MAC schemes inapplicable. Since the radio communication of a sensor node consumes the highest amount of power [2], the main method of preserving power is to duty cycle the node. Duty Cycles are materialized by alternating the node between active and sleeping states, where the node is operational in the active state and shut down in the sleeping state. This poses a particular problem of finding a rendezvous point between a sender and receiver, in which both of the nodes are in an active state and a communication link can be established. MAC schemes for WSNs take a synchronous or asynchronous approach to solve this problem. Figure 1, depicts the synchronous and asynchronous paradigms for coordinating the receiver and the transmitter in duty cycled wireless communications.

In protocols that follow the *synchronous* approach, like S-MAC [3], T-MAC [4] and DSMAC [5], nodes organize the active and sleeping states to align. Synchronous schemes can be based either on contention or on reserved time-slots. In both cases, a portion of the active state is used to synchronize all the nodes to a global active/sleep schedule. When a source node has data to transmit, it waits until the active state to initiate the data transfer. Synchronous schemes are quite tolerant to schedule misalignment, however, they still require a globally synchronized schedule, which creates an additional energy overhead. Additionally, synchronous protocols have a cost associated with the creation and maintenance of the schedule. Furthermore, the coupling of nodes via a global clock also hinders a node's ability to have a fully independent duty cycle, so that each node can adapt, in a fully distributed way, to the current surrounding conditions.

Asynchronous schemes do not require synchronization, as the nodes sleep and wake-up independently of the others. This leads to the need of techniques on deciding a rendezvous point for nodes to communicate. There are two fundamental asynchronous techniques, namely the sender- and the receiver- initiated. The basic technique used in a sender-initiated asynchronous MAC scheme is called preamble sampling, where the sender transmits a preamble to indicate that there is a pending need for communication. The receiver wakes up occasionally into

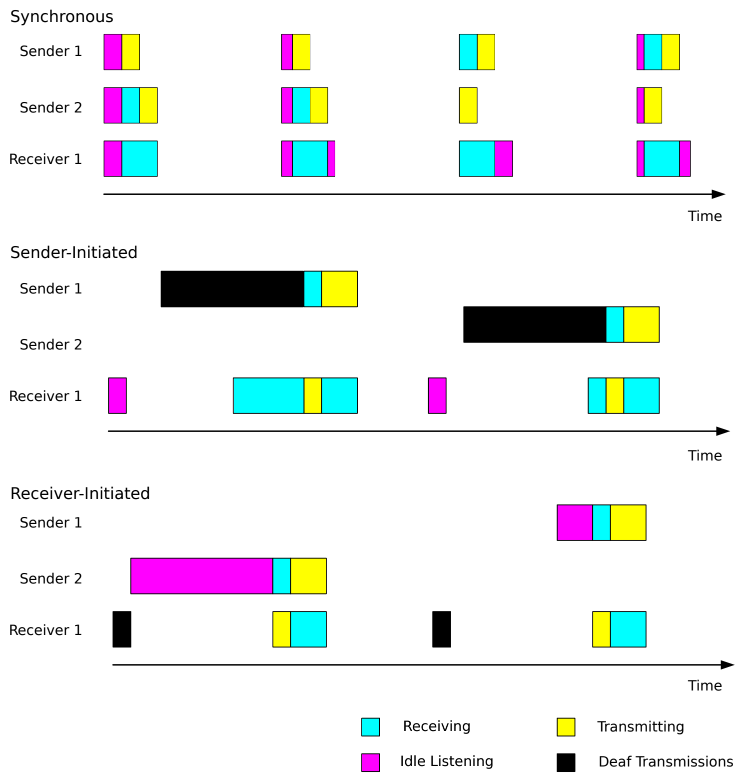

Figure 1: Different approaches of MAC protocols, from top to bottom: synchronous, sender-initiated and receiver-initiated.

the active state, to listen to such a preamble transmission. Once the preamble is detected, the receiver replies with a positive acknowledgment to the sender when the preamble transmission stops. This establishes a communication link between the sender and receiver. Most notable examples of MAC protocols that are based on the sender-initiated paradigm are WiseMAC [6], B-MAC [7] and X-MAC [8]. A thorough survey of sender-initiated schemes is performed in [9], concluding with a guideline to select MAC schemes for a given application.

1.1. Contribution of the Paper

This survey is focused on the latter asynchronous scheme, namely receiver-initiated. In contrast to the preamble sampling technique in sender-initiated schemes, receiver-initiated schemes use another approach to asynchronous communication: instead of long preambles, the sender listens to the channel, waiting for small beacons transmitted by the receiver. The receiver transmits the beacons in a period that is defined by its duty cycle, and is used by the sender to synchronize with the receiver. The receiver-initiated paradigm was originally introduced by Lin *et al.* in 2004 (RICER [10]) and made popular by RI-MAC [1] in 2008. Since the publication of RI-MAC, several MAC protocols that build on the receiver-initiated paradigm have been proposed (see Figure 2).

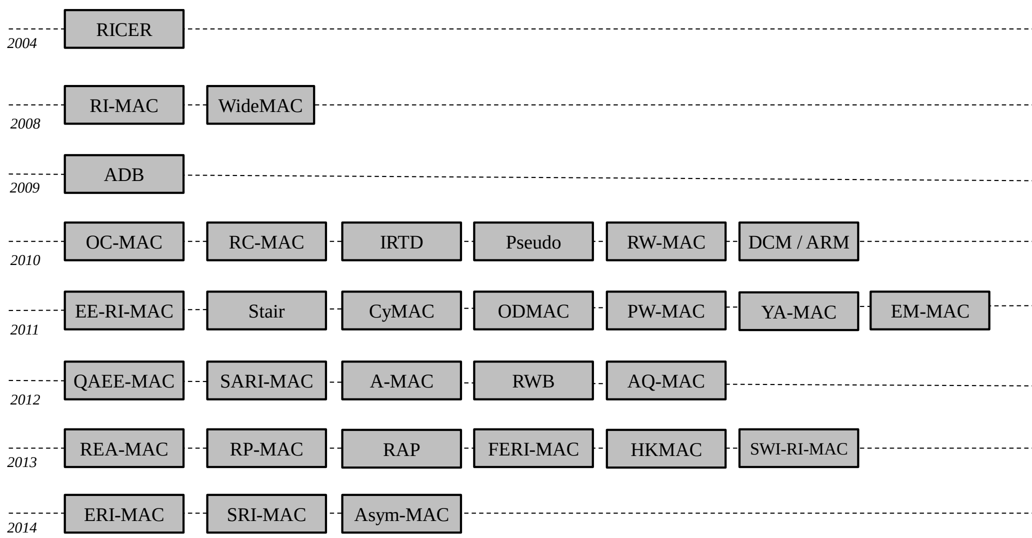

Figure 2: Chronology of Receiver-Initiated MAC Protocols.

Within the asynchronous approach, the receiver-initiated scheme is shown to be more energy efficient [1, 11, 10] than the sender-initiated scheme, making receiver-initiating protocols very suitable for future energy-aware wireless sensor networks. A performance comparison between the three paradigms is out of the scope of this survey. Instead, our focus is on the new design challenges introduced by the receiver-initiated paradigm. To this end, we highlight the key design challenges a receiver-initiated MAC protocol should address. Then, keeping in mind these challenges, we survey all the MAC protocols released since the year 2004 that fall under the receiver-initiated category, analyzing and organizing them according to common features and design goals.

1.2. Outline of the Paper

In Section 2, we present the challenges that receiver-initiated MAC protocols are meant to deal with. Section 3 classifies and presents all the existing MAC protocols that are based on the receiver-initiated paradigm. The classification is based on the (most prominent or novel) features that each protocol implements. In Section 4 we provide an overview of some applications developed using receiver-initiated protocols as core building blocks. In Section 5, we discuss, summarize and compare the surveyed protocols, focusing on how appropriate they are for specific application classes. Lastly, Section 6 concludes the survey.

2. Challenges for Receiver-Initiated MAC Protocols

MAC protocols are typically responsible for controlling the communication between two nodes over a link and for coordinating multiple nodes that share the same medium. Some of these tasks carry over from regular wireless networks, for example *protocol overhead* has to be taken into account: both activating the radio transceiver and producing unnecessary data exchange would lead to performance degradation, therefore the size and number of packets sent should be kept to a minimum. Naturally, the goals (and hence the definition of performance) are different between regular and sensor-based wireless networks; most likely in the first case the dominating factor is throughput while in the second is energy preservation and network lifespan, but in the end the same concept still applies. *Channel error handling* is also a well-known problem with fairly standard solutions. Acknowledgements, re-transmissions, cyclic redundancy checks (*CRC*s) (or authentication code if security is involved) are pretty much standard and consolidated techniques used in almost every MAC protocol for wireless networks.

In addition to these, new challenges are introduced. For example, receiver-initiated MAC protocols for WSNs have to deal with the fact that wireless sensor nodes are duty cycling between active and sleeping states to save energy. This produces new challenges for the MAC layer, such as minimizing the energy overhead for synchronizing the transmitter and the receiver. Moreover, broadcasting becomes less trivial, as some of the nodes could be sleeping at any given time.

In this section, we summarize the important challenges of the MAC layer for duty cycling nodes that are following the asynchronous receiver-initiated paradigm.

2.1. Idle Listening

According to the receiver-initiated paradigm, each node with data to transmit enters an active state and listens to the medium for a beacon from the intended receiver. Until the time when the receiver wakes up from its sleeping state and transmits the beacon, the transmitter is essentially wasting energy listening to the channel without receiving any useful data. At the receiver's side, after every unanswered beacon, the node also wastes energy listening for a reply. This energy overhead is named *idle listening* and constitutes a weakness that is associated particularly with the receiver-initiated paradigm. As a result, there is significant literature work, focused on mechanisms to mitigate it.

2.2. Collision Avoidance

Contention-based MAC protocols for wireless communication are known to be vulnerable to colliding transmissions, as a radio that is transmitting is unable to detect other transmissions in the wireless medium. Collisions decrease the systems performance and are also a source of energy wastage. Protocols following the asynchronous receiver-initiated paradigm, may be either vulnerable or resilient to collisions depending on the topological structure of the network and the duty cycles of the nodes. This phenomenon rises because of the fact that beacons constitute indirect transmission time-slots. When the beacon transmission rate is significantly higher than the data transmission rate, the stochastic selection of a beacon acts as an indirect proactive collision avoidance mechanism (random access). Yet, there is always the chance for multiple nodes to select the same beacon / time-slot. Hence, when the beacon and data transmission rate is at a similar order of magnitude, collisions are significantly increased and the system is lead to a state where the receivers are flooded with more transmissions than they can handle. This scenario appears either in topologies when few receivers have to handle large numbers of transmitters or in the case of low duty cycle receivers are serving high duty cycle senders. The latter case requires active collision avoidance.

2.3. Adaptive Duty Cycling

The dynamic adaptation of the duty cycles can significantly improve the energy efficiency of the system. A MAC protocol with adaptive duty cycles, that is aware of the structure of the topology, the traffic conditions or the resources of the nodes, can more efficiently use the available energy. For example, the nodes that are closer to the sink typically have more forwarding tasks rather than the nodes that further away. Additionally, independent duty cycle adaptation is vital for WSN that are powered by harvested ambient energy, such as solar energy, vibrations or heat. The system goal of such networks is to operate at a state where the consumed energy is on average equal to the harvested energy (i.e. *ENO-Max* [12]). Due to the chaotic nature of the environmental energy sources, the duty cycles of the node need to be frequently and independently adapted.

2.4. *Quality of Service*

Different types of packets can coexist within the network. According to the requirements of the overlying application, or even the protocol itself, each class of frame might require different handling. For example high priority messages might be relayed before low priority ones, frames could be reordered to minimize delay or again control messages could take precedence over data messages to ensure the correct functioning of the network. All these kind techniques fall under the general definition of *Quality of Service*.

2.5. *Broadcast Communication*

Although trivial for typical MAC protocols for wireless communications, broadcast communication constitutes a challenge in networks of nodes that are duty cycling in an asynchronous manner. Since the sleeping and activity periods of nodes is not synchronized in time, it is unlikely for a transmitter to find a moment where all the nodes are awake and ready to receive a broadcast transmission. Assuming a system-wide known maximum beacon period, this issue can be solved by replacing a broadcast communication with multiple unicast transmissions. Nevertheless, there is work in literature on efficient ways to overcome this challenge.

2.6. *Asymmetric Links*

Asymmetric links refer to links where the quality of the two directions is significantly different. For data transmissions the vital direction of a link is from the sender to the receiver. In asymmetric scenarios the quality of the link from the receiver to the sender may cause unnecessary problems to data transmissions that could be avoided if the communication was not receiver-initiated.

2.7. *Security*

Sensor networks are vulnerable to attacks which are associated with the wireless medium. Wireless channels can be easily eavesdropped and traffic can be easily injected or altered. Attackers are not limited by the resource constraints of sensor nodes and can interact with the network from afar, using much more powerful equipment. Moreover, sensor networks may be deployed in psychically insecure environments and sensor nodes are vulnerable to resource depletion attacks and tampering in general. The security of the MAC layer is fundamental for the security of the system.

Feature	Protocols
Receiver-initiated	RICER
Basic extensions	RI-MAC, OC-MAC, RC-MAC, IRDT, EE-RI-MAC, A-MAC, REA-MAC, RP-MA, ERI-MAC
Wake-up prediction	WideMAC, Pseudo, RW-MAC, PW-MAC, FERI-MAC
Adaptive duty cycling	Stair, ODMAC, SARI-MAC, HKMAC
Quality of service	CyMac, QAEE-MAC, AQ-MAC
Broadcast support	ADB, YA-MAC, RWB
Multi-channel extensions	DCM, EM-MAC
Hybrid approach	SWI-RI-MAC, SRI-MAC, Asym-MAC
Security	RAP

Table 1: A list of the surveyed protocols organized by their prevalent feature.

3. Receiver-Initiated MAC Protocols

The receiver-initiated paradigm of asynchronous communication for duty cycling nodes was introduced by RICER [10] in 2004. In 2008, Koala [13] defined a receiver-initiated mechanism, named Low Power Probing (LPP), which uses the receiver-initiated paradigm for the purpose of waking up all sensor nodes, while it is not involved in the actual data transfer. Later, the receiver-initiated paradigm was popularized by RI-MAC [1], which triggered vast research that builds upon the paradigm and optimizes its performance.

Each protocol that extends the receiver-initiated paradigm focuses on one or more of the challenges enumerated in Section 2. The rest of the section and the surveyed protocols are organized as follows. First, we present the receiver-initiated paradigm, as it was introduced by RICER [10] (Section 3.1). Section 3.2 surveys the receiver-initiated MAC protocols that provide an extension of the paradigm with focus on the fundamental challenges of *Idle Listening* and *Collision Avoidance*. The focus in Section 3.3 is on mitigating *Idle Listening* in the particular direction of predicting the wake-up of the receiver. Section 3.7 surveys protocols that focus on the direction of using multiple channels to distribute the transmissions and decrease the contention. The remaining subsections can be directly mapped to a respective challenge in focus, as listed in Section 2. Table 1 summarizes the organization of the protocols according to their key design feature.

3.1. The Receiver-Initiated Paradigm of Communication

The receiver-initiated paradigm operates as follows. Each node periodically wakes up to check for incoming data. After each wake-up event, a *beacon* is broadcasted. This beacon announces to the neighbors that it is ready to accept incoming data. After the beacon has been transmitted, the receiver continues to listen to the channel for a short period of time. Whenever a node with data ready to be sent enters the active state, it listens silently to a beacon from the intended receiver. Once the beacon is received, the sender immediately starts transmitting the data, and waits for a time period to receive a frame which acknowledges the reception of the data. If there is no incoming data from the sender after transmitting the beacon, the receiver enters the sleeping state. Both the sender and receiver, then resume their cycles.

In comparison to the sender-initiated paradigm, the receiver-initiated communication paradigm significantly reduces the amount of time for which a pair of nodes occupy the channel, allowing more contending nodes to communicate with each other, increasing the capacity and throughput of the network. It is more efficient in detecting collisions and recovering lost data, because access to the channel is mainly controlled by the receiver. Since receivers only wait a short period of time for incoming data, after beacon transmission, overhearing is greatly reduced [1, 11, 10].

3.1.1. Receiver-Initiated CyclEd Receiver (RICER) [10]

Beyond introducing the paradigm, RICER also defines several features that improve the performance of the protocol. First, it uses a random delay between the hearing of the wake-up beacon and starting the data transmission to avoid collisions. Furthermore, the authors note that a significant reduction of the energy consumption can be achieved by introducing multiple potential receivers. However, no particular receiver selection policy is specified, as it is considered a task of the routing layer. Lastly, a semi-synchronous mode is defined to decrease the energy consumption. With globally known duty cycles, nodes can keep record of the wake-up times of neighboring nodes to predict with approximation the upcoming wake-up.

3.2. Basic Extensions

RI-MAC [1] and other MAC protocols build upon the paradigm with features that optimize their performance.

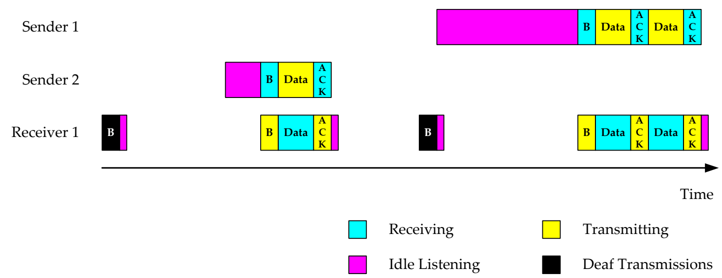

Figure 3: Mechanics of RI-MAC, the protocol that made the receiver-initiated paradigm popular. Beacons are sent out by the receiver in order to communicate its availability to receive data.

3.2.1. Receiver-Initiated MAC (RI-MAC) [1]

RI-MAC builds on the receiver-initiated paradigm and provides an implementation that is incorporated in TinyOS [14]. RI-MAC extends the paradigm with the following features. After data transmission and if the sender has more data packets to send, it uses the acknowledgment beacon as a Ready-To-Receive (RTR) indicator, to start transmitting the next data packet. If there is no incoming data from the sender after transmitting a beacon, the receiver enters the sleep state. The beacon frame in RI-MAC plays a dual role. It is used both as a RTR, broadcasting the request to initiate data transmission, in essence, creating a time-slot for rendezvous, and as an ACK, which informs the sender that the data has been received successfully. An optional destination address field is used in the ACK reply to signify a unicast transmission, so that other nodes waiting for a beacon can ignore it. The duty cycle of the beacon transmissions are controlled by varying the sleep state, L, of the node. To prevent coincidental synchronization, a node sets the sleep period randomly between $0.5L$ and $1.5L$, before entering the sleep state. This essentially makes the average duty cycle of RI-MAC static. An overview of the communication in RI-MAC is shown in Figure 3.

If two or more senders contend for the same base beacon, the data packets will be transmitted simultaneously. The experiments conducted in RI-MAC, have shown that due to the presence of the capture effect [15] in FM radios, also called co-channel interference tolerance, such a contending scenario does not necessarily lead to collisions. This property demonstrates that the traditional assumption that a packet collision always results in data corruption is false. For this reason, senders in RI-MAC immediately transmit the data upon receiving a base beacon, without any backoff. The receiver listens for a short period of time after transmitting the beacon, known as the *dwell time*, which is determined by the current

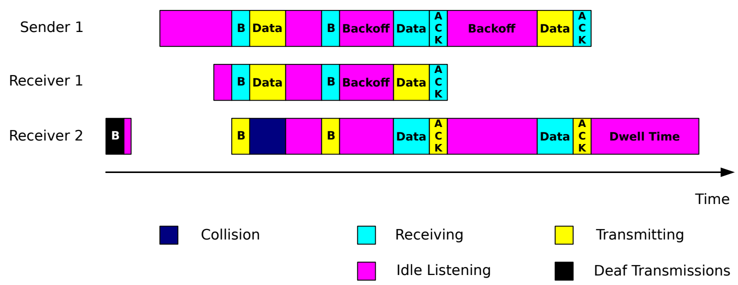

Figure 4: Collision avoidance mechanism in RI-MAC, a form of binary exponential backoff.

backoff window size. Concurrently, it measures the channel power level and processes the bit pattern received. If a valid data frame header is not detected in time, and the measured power level indicates that a transmission is in progress, then, this condition is classified as a collision. Figure 4 shows the collision avoidance technique used by RI-MAC. If a collision occurs, the receiver performs a *Clear Channel Assessment* (CCA), waiting for the channel to be free. Once a clear channel is determined, the receiver transmits a beacon with a backoff window specified, informing the senders of the failed transmission. The senders, that are waiting for an ACK, use the backoff window specified in the beacon to perform a random backoff. The senders listen to the channel, while waiting for the random period to expire, before re-transmitting the data. If a transmission from another sender is detected, the sender withholds the transmission, and waits for an ACK beacon, before resuming with a new random backoff. If a collision happens again, the receiver increments the backoff window using a Binary Exponential Backoff (BEB) [16] strategy, until the maximum window size is reached, after which, the senders and the receiver accept a failed transmission and go back to sleep, retrying at a later point in time.
Beacon-on-Request is an optimization feature, defined by RI-MAC, for instance when the intended receiver is already active, as shown in Figure 5. After a CCA, a sender that has data to transmit, immediately broadcasts a beacon with a backoff window size specified and the destination address set to the intended receiver. The beacon acts as a Ready-To-Send (RTS) indicator, and if the receiver happens to be awake, it replies with a base beacon after a random backoff period. Data exchange then occurs using the normal RI-MAC communication mechanism.

3.2.2. Opportunistic Cooperation MAC (OC-MAC) [17]

OC-MAC [17] extends the beacon-on-request feature to reduce the time that a sender waits for a beacon. Neighboring senders in OC-MAC are allowed to

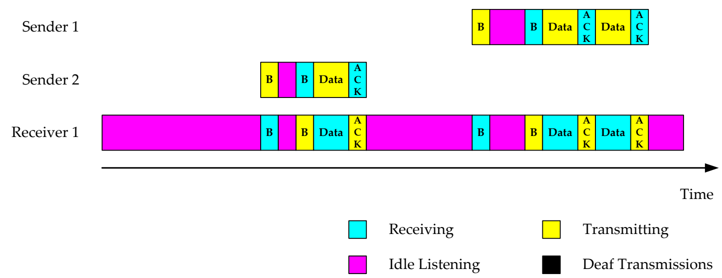

Figure 5: Beacon-on-request mechanism in RI-MAC, beacons can be requested explicitly if the intended receiver happens to be awake.

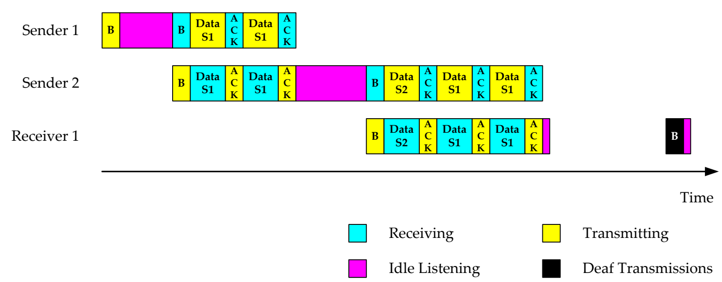

Figure 6: OC-MAC extends the beacon-on-request functionality of RI-MAC in a sender oriented manner.

exchange data aggressively while waiting for the receiver to wake-up. Figure 6 provides an overview of the mechanism used in OC-MAC. Similar to the beacon-on-request feature of RI-MAC, when a node has data ready, it transmits a RTS beacon, if the channel is idle. The beacon contains its residual energy, the destination address, and a request for other senders to relay the data. Notice that, in contrast to the beacon-on-request feature of RI-MAC which is directed towards receivers, the beacon-on-request in OC-MAC is directed only towards senders. By not loading the receivers, this ensures that the channel is not drained of beacons, which would reduce the throughput of the network. After the beacon is transmitted, the sender listens to the channel for a period of time. If it does not receive a response within this duration, the sender looses its right to cooperative communication, and continues to wait silently for a beacon from the receiver or another contending sender.

When an RTS beacon is received by a sender that coincidentally happens to be awake, it compares its residual energy to the contender. The sender ignores the request if the contending sender has more residual energy than itself. If the contender has less residual energy than the sender, it transmits a CTS beacon, similar to the base beacon in RI-MAC, after a random backoff. The backoff prevents collisions, in case multiple senders are active. The rest of the mechanism is similar to the beacon-on-request feature in RI-MAC. Once the exchange of data is completed, the contending sender enters the sleep state, while the sender which received the data, transmits another RTS beacon to check if any opportunity exists to relay both its own data, and the data from the contending sender. Hence, a sender is only permitted to broadcast a RTS beacon immediately after waking up, or after completing a cooperative communication with a contender.

3.2.3. Receiver-Centric MAC (RC-MAC) [18]

RC-MAC is a MAC protocol designed for event-driven applications with heavy traffic loads. It adopts the receiver-initiated paradigm for as long as the network has low traffic for higher efficiency. Differently from RI-MAC where beacon senders transmit immediately upon a beacon reception, RC-MAC requires a initial random backoff in order to increase the fairness between nodes with different transmission power. This approach, on the other hand, is also increasing the energy overhead, since the idle listening is increased. Additionally, in case of collision the senders will retry with a binary exponential backoff whenever the ACK packet is not been received. The receiver is expected to be awake because it just received a frame and it is waiting for a beacon from the next hop. The amount of retries is limited by a predefined number of re-transmission attempts. If this limit is reached, the sender discards the beacon and waits for a new one.

3.2.4. Intermittent Receiver-driven Data Transmission (IRDT) [19]

IRDT is extending the paradigm with two additional control packets, namely the RACK and the DACK. After the reception of the beacon, named ID, the sender is transmitting the RACK frame to establish the connection. Then, the data frame transmission follows which is acknowledged with the DACK frame. Additionally, the protocol is defining three collision avoidance mechanisms. The first is CCA with random backoff similar to RI-MAC. The second is based on the frequency of beacon transmissions. The idea is that by increasing the beacons, the senders are stochastically distributed over more beacons and the probability of collision decreases. However, this solution can work only if the receivers are capable of offering their energy resources for forwarding more traffic.

The third collision avoidance mechanism is based on data aggregation. By aggregating multiple data packets into larger frames, the total amount of attempted transmissions falls; thus, the probability of a collision decreases. However, this approach has a negative impact on the delay of each individual data packet. The authors define two methods of collision avoidance with data aggregation, a static one and a dynamic one. According to the static method, the protocol is using a constant buffer of n packets. The node keeps collecting packets from other nodes and locally generated packets into a buffer. When the buffer is full, it is transmitted as a single MAC frame. According to the dynamic method, a sender with a single packet to transmit is waiting normally for the beacon. While waiting, it periodically transmits its own beacons in order to collect packets from neighbors. When the beacon is received, the sender transmits a single frame with as many packets as it managed to collect during that time.

3.2.5. Energy Efficient RI-MAC (EE-RI-MAC) [20]

EE-RI-MAC is an enhancement for RI-MAC, defining another approach to increase the energy efficiency of the senders. In particular, EE-RI-MAC uses a technique inspired by X-MAC [8], where, instead of listening for a beacon, a sender alternates between the active state and sleep state within this duration. Figure 7 shows an overview of this approach. In order to further reduce the idle listening, senders, enter the sleep state after listening to the channel for a period W, and wakes up after a duration S. The authors of EE-RI-MAC, opted to use simulations to determine the optimal duty cycle for alternating between the active and sleep state during the idle listening period. It was found that a duty cycle of 37.5%, resulted in the optimum case, outperforming RI-MAC in terms of energy usage. The choice of the value used in the two important parameters, W and S, determines the performance of the scheme. Additionally, even though EE-RI-MAC achieves the same throughput as RI-MAC with higher energy efficiency, the latency of the network suffers.

3.2.6. A-MAC [21]

The key extension of A-MAC to the receiver-initiated paradigm is an extra control packet that aims to reduce the time that a receiver waits for a sender to reply after a beacon transmission. In particular, in A-MAC, the beacon is acknowledged by a short packet named *HACK*. The purpose of this acknowledgment is to quickly inform the receiver of the existence of pending traffic. If the beacon does not trigger a *HACK* packet, the receiver goes directly to sleep. As a result, the receiver wastes less energy in idle listening after each unanswered beacon. In case different *HACK* packets from multiple senders collide, the receiver is still able to assess that there is pending traffic and keeps the radio on. Furthermore,

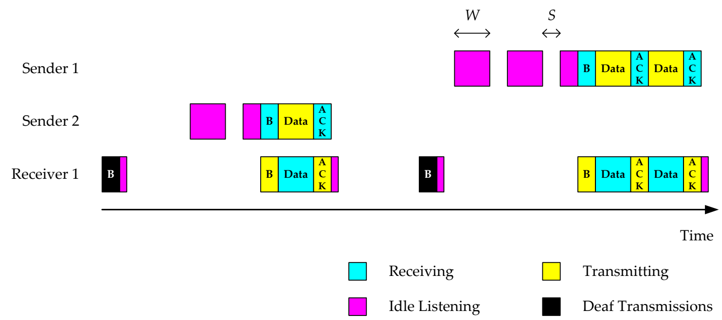

Figure 7: EE-RI-MAC introduces the use of duty cycled waiting for beacons in order to reduce idle listening.

A-MAC incorporates the LPP [13] mechanism for asynchronous network wake-up from deep sleep. In case of no traffic, the network can fall in a deep sleep where the nodes just wake-up to transmit beacons very infrequently. Upon an event that should trigger a network wake-up, a node turns on and keeps its radio enabled, listening for beacons. These beacons are answered to with wake-up requests. Nodes that receive such request will propagate it, progressively awaking the whole network. The maximum time required for an asynchronous network wake-up depends on the beacon frequency of the nodes in deep sleep.

3.2.7. Routing-Enhanced Asynchronous MAC (REA-MAC) [22]

REA-MAC builds on the receiver-initiated paradigm by coordinating the beacon transmissions. The proposed mechanism uses the distance in number of hops of each node from the sink, which is a cross-layer information from the routing layer, to form an *operation cycle*. This cycle is a network-level duty cycle that is built on top of the duty cycles of individual nodes. If N is the maximum distance (in hops) of a node from the sink, the *operation cycle* is split into N wake-up time-slots. Instead of transmitting beacons independently, each node transmits during the time-slot which corresponds to its particular distance to the sink. Therefore, the beacon transmissions in a network are coordinated to form a multi-hop path like a pipeline and the waiting time in each hop is significantly reduced. Furthermore, a node that has generated data, can keep the radio off during the irrelevant frames to save additional power. The proposed idea is compared to RI-MAC and the simulations show significant reduction of the delivery latency and the power consumption.

3.3. Wake-up Prediction

Idle listening constitutes by far the most prevalent source of energy consumption in a receiver-initiated MAC scheme [23]. Several protocols work towards mitigating the time a sender is waiting for a beacon by predicting the next wake-up of the intended receiver.

3.3.1. Wide-band MAC (WideMAC) [24]

WideMAC assumes globally known and static duty cycles, i.e. beacon periods, which are used to predict the next wake-up and decrease the idle listening overhead. In particular, at the beginning a node operates similarly to RI-MAC. Once a node has received a beacon from a receiver node, it predicts the time of the next beacon transmission of the specific node by using the globally known beacon period. Due to clock drifts, the value of this prediction decreases over time, up to a point where it is not longer useful. Whenever a node receives a beacon, it also updates this information.

3.3.2. Pseudo-Random Asynchronous Duty Cycle MAC (Pseudo) [25]

In this work, the authors are using a hash function to create pseudo-random wake-up intervals that are uniformly distributed in the range of $[T_{\text{mean}} - T_{\text{range}}/2, T_{\text{mean}} + T_{\text{range}}/2]$, where T_{mean} is the average long term wake-up interval (i.e. the average duty cycle) and T_{range} defines the range of the randomization. Such a randomization, distributes the frame transmissions in the dimension of time, thus decreasing the collisions. Moreover, the hash function is globally known by all the nodes. Thus, each node is able to estimate the next wake-up time of each receiver. Additionally, the authors consider that potential channel contention may introduce delays that can affect the predictions. So, the beacon is enriched with a sequence number and the difference between the wake-up time and the start time of the base transmission. The receiver of the beacon is using the beacon sequence number as input to the hash function in order to predict the next wake-up time. Then, this prediction is corrected by adding the aforementioned delay. Lastly, each sender wakes up some time before the calculated wake-up time of the receiver, to account for clock drifts. This time is calculated based on the upper bounds of clock drift, given in the datasheets of the micro-controllers.

3.3.3. Receiver Wake-up MAC (RW-MAC) [26]

The energy wasted during the idle listening period of the sender is significantly reduced by predicting the wake-up time of the receiver in RW-MAC. The sender uses the remaining sleep time T_{interval} of a receiver, which is piggybacked on the beacon, to estimate it's wake-up time. Each node maintains a table with the previous time t_{prev} a beacon *should* be received from its neighbors. Initially the sender has to remain awake for a period of time to populate the neighbor table. A sender with data to transmit wakes up after extending the sleep state by the sleep wait time T_{wait} and listens for a beacon from the receiver. T_{wait} is calculated by taking into account the worst case frequency drift θ of the quartz crystal, the static duty cycle T_{cycle} of nodes, and t_{prev}. The maximum time the sender listens to the channel after waking up is set to T_{cycle}, beyond which the node is considered offline or not in the neighborhood.

The beacon and data transmissions are prone to collisions due to the lack of CCA. RW-MAC introduces a stagger wake-up concept as a collision avoidance mechanism. When a sender is initially powered up, it listens to the channel for two consecutive cycles in order to find the maximum gap between two received beacons. It then calculates a non-optimal stagger wake-up offset T_{offset}, based on the midpoint of the gap and T_{cycle}, which is used to permanently shift the beacon cycles of the node. The experimental results show that RW-MAC outperforms RI-MAC for high traffic loads. It supports a higher number of concurrent data flows and consumes less energy than its counterparts due to its low duty cycle.

3.3.4. Predictive Wake-up MAC (PW-MAC) [27]

PW-MAC, is a receiver-initiated scheme that reduces the energy consumption of senders, inspired by WiseMAC [6]. PW-MAC, uses an independently generated pseudo-random sequence for controlling the wake-up times of each node, allowing senders to accurately predict the time when a receiver will wake-up, similarly to [25]. An on-demand prediction error correction mechanism helps to compensate for timing challenges caused by unpredictable hardware, operating system delays, and clock drift. Furthermore, the predictable wake-up times are used to improve the performance in case of collisions and channel errors. In case there is need for a re-transmission, senders in RI-MAC stay awake until receivers wake-up again. On the contrary, senders in PW-MAC wake-up at the next predicted receiver wake-up time, minimizing the energy spent waiting for the receiver.

3.3.5. Reordering Passive MAC (RP-MAC) [28]

RP-MAC extends RW-MAC with a feature called *Frame Reordering (FR)*. The FR scheme reduces the delivery latency by using the next wake-up information of several receivers to reorder the transmission buffer of the sender. For instance, consider the scenario depicted in Figure 8, where the buffer of the sender has a frame for $R1$ that is followed by a frame for $R2$. However, the next wake-up of

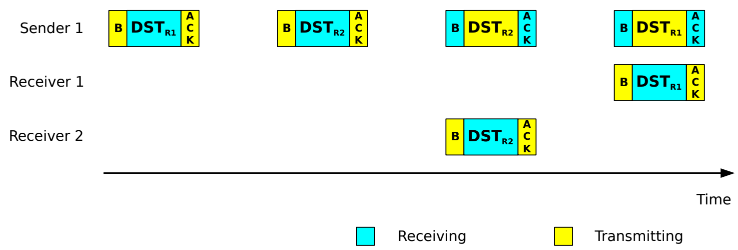

Figure 8: Frame reordering in RP-MAC, frames are sent out according to the beacon interleaving pattern.

$R2$ will happen before the next wake-up of $R1$. The FR scheme reorders the two frames to significantly reduce the waiting time. Compared to RI-MAC, RP-MAC achieves better energy efficiency and lower end-to-end delay.

3.3.6. Traffic Estimation for Underwater Networks (FERI-MAC) [29]

FERI-MAC is designed specifically for Underwater Acoustic Networks (UANs), and tackles problems that are characteristic of this setting, such as the extended preamble of acoustic modems and the increased signal attenuation. The main technique used by FERI-MAC is traffic estimation, used to predict when a sender node will have data to transmit, and consequently when to send a beacon.

The protocol is composed of four phases, first a receiver sends a *Request-To-Receive* (RTR) message which states the intention of receiving data from its immediate neighbors. This packet contains the address of the node, the address of the next-hop, the addresses of the intended neighbors and the slot for the following data transmission. After this, the second phase requires the contacted neighboring nodes to answer with an *Available-To-Send* (ATS) packet. Each node should answer according to the order advertised in the previous RTR packet. Each ATS packet contains the number of packets queued by the sender and its address. The former value is a useful hint for the receiver to feed to the traffic estimation algorithm, while the exchange of RTR and ATS packets is also useful to inform all the neighboring nodes of a soon-to-happen transmission, hence avoiding collisions. After this the data transmission phase takes place. Here the time is divided into mini slots, assigned to the nodes from the RTR received in phase one. Since the allocation is generated from the traffic estimation, some of these slots can be unused due to lack data if the estimation process is not accurate. As a result FERI-MAC is not a good match for delay-critical applications. Finally the fourth and last phase consists of a single integrated ACK sent out from the receiver and intended to each sender that has successfully carried out the communication.

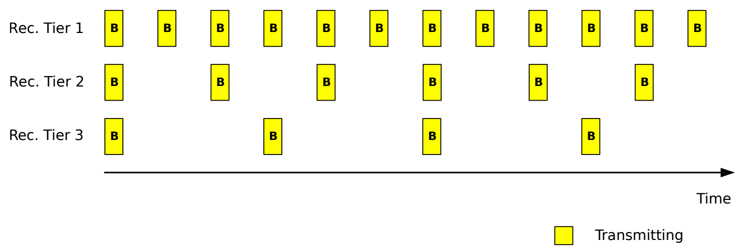

Figure 9: Stair-like beaconing pattern determined by a node's tier (lower numbers mean shorter distance from the sink).

Energy savings are achieved in three ways: collisions are mitigated through slots allocation, packet aggregation helps to reduce the number of packets and hence the related overhead and, finally, the number of ACKs is greatly reduced.

3.4. Adaptive Duty Cycling

Dynamic adaptation of the sleeping schedules is optimizing the performance of the paradigm to given dynamic conditions. Dynamic duty cycling can be based on several parameters such as the topological structure, the traffic conditions or the energy input.

3.4.1. Stair-like Sleep Asynchronous RI MAC (Stair) [30]

A fundamental limitation of receiver-initiated sensor networks is that the amount of energy expended by a sender node while waiting for a beacon, highly depends on the availability of a receiver node to transmit such beacon. Therefore, a low duty cycling receiver will force the transmitter to waste a significant amount of energy, leading to sub-optimal network performance. To make matters worse, the closer a node is to the sink, the more network traffic it has to serve. The authors propose an asynchronous receiver-initiated protocol that builds upon this limitations. In particular, the authors show via simulations that the overall network performance, in terms of packet delivery ratio, packet delay and energy efficiency, can be significantly improved by adapting the duty cycles considering the number of hops of each node from the sink. Such an adaptation would lead to stair-like sleeping pattern (Figure 9), in which the closer a node is to the sink the more time it stays active. Despite the promising results at a network level, the individual node's energy capability to support the higher duty cycles should be taken into consideration. Furthermore, it is interesting to note that the same beneficiary effects would result from a topology designed with more nodes placed closer to the sink.

3.4.2. On Demand MAC (ODMAC) [31]

ODMAC builds upon the foundation of the receiver-initiated paradigm for the realization of Energy Harvesting WSNs (EH-WSN), which are sensor networks that are powered by energy that is harvested from the surrounding environment. ODMAC uses an adaptive duty cycle mechanism based on the ENO principle [32], where the energy consumed by a node is less than or equal to the amount of energy harvested. All nodes in the network dynamically adjust the beacon and sensing duty cycle, in order to achieve and maintain an ENO-Max state[12], which is defined as an ENO state with maximum performance. This means that when the node is consuming more energy than it harvests, the duty cycles are decreased to reduce the energy consumption. In the same manner, when the energy consumed is lower than the energy harvested, the duty cycles are increased so that the node is more active. Nodes in the network have a dual role of being a receiver for forwarding tasks and sender for measuring tasks. ODMAC decouples the duty cycles of these two roles in a single node. Hence, a node has a beacon duty cycle and a sensing duty cycle. The beacon duty cycle controls the trade-off between energy consumption and end-to-end delay, while the sensing duty cycle controls the trade-off between energy consumption and measurement rate. Therefore ODMAC gives to an administrator the ability to decide the trade-offs depending on the application.

In a scenario with two or more receivers, where one receiver is on a high beacon duty cycle due to the high amount of energy it harvests, a sender has to wait for a long time listening to the channel if it wants to exchange data with another receiver. To prevent such a situation from arising, ODMAC defines a forwarding policy based on opportunity. Instead of waiting for the intended receiver to wake-up, a sender opportunistically forwards data using the first available beacon that leads towards the desired destination. Since the probability of receiving beacons from a receiver with surplus energy is high, this policy creates a more robust network, that is adaptive to changes in energy, by maintaining a balanced load in the network. Furthermore, the idle listening time of senders is reduced in the region where the receivers' coverage overlaps. Figure 10 shows an example of such an opportunistic policy, where as the state of energy of the receivers change, the nodes that are in range of both the receivers adapt to the receiver with more energy.

In addition to random backoff, ODMAC also includes a novel low-overhead collision avoidance mechanism, named *Altruistic Backoff* [33], that detects potential collisions and avoids them before the beacon transmission. Therefore, the

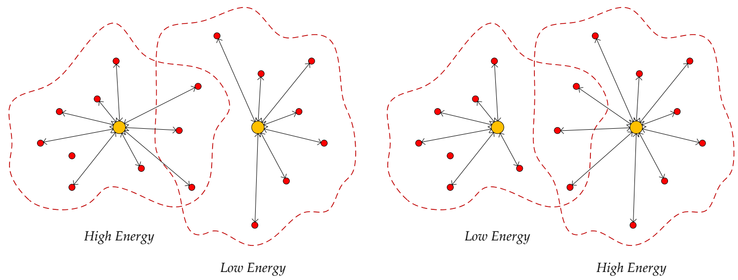

Figure 10: Opportunistic forwarding in ODMAC in a multi-sink, single-hop network.

nodes back off earlier and less energy is wasted in idle listening. The scheme works as follows. When a node wakes up it transmits a control packet, named *ABR (Altruistic Backoff Request)*, that states which beacon(s) the node is waiting for. A node that is already waiting for the same beacon and receives this packet altruistically backs off, offering the beacon to the node that woke up last. The scheme is compared with the BEB, as implemented in RI-MAC, and the results suggest that idle listening is significantly reduced.

3.4.3. Self Adapting RI-MAC (SARI-MAC) [34]

SARI-MAC self-adapts to the traffic load by adjusting the beaconing frequency to the estimated traffic. In particular, the maximum duration between two beacons is capped by the maximum link delay that is allowed by the application. Moreover, the duration between two beacons is also adapted so that the average beacon rate is equal to the average traffic rate. The later adaptation ensures that the beacon transmission frequency is large enough to serve the incoming traffic. SARI also introduces a novel collision avoidance mechanism through time slot reservation. After the beacon transmission, a contention window period follows during which the nodes pick a uniformly random slot to request for a time-slot reservation. At the end of the contention window, the receiver sends back to all the contending nodes a report with the reservations. Nodes transmit their data during their reserved time-slot, which is long enough for a data packet and the respective acknowledgment.

3.4.4. Low-latency Burst Traffic (HKMAC) [35]

HKMAC is an asynchronous duty cycle receiver-initiated MAC protocol for burst traffic in WSNs. It works as follows. Each node wakes up periodically to broadcast a beacon. When a node with queuing data packets receives the beacon from its intended receiver, it transmits data immediately. If no packet is received after broadcasting its beacon, the node sleeps to save energy. Time is divided into random periods (RP) and scheduled periods (SP). During random periods nodes wake-up without a predefined schedule and transmit their beacons. On the other hand, during scheduled period each receiver adaptively adjusts its beacon time. Thanks to this mechanism, HKMAC allows for packets to be forwarded in a pipelined manner, and manages to achieve low end-to-end delivery latency for bust traffic. Additionally, the sender is aware of the wake-up time of the receiver, hence it can schedule its transmissions to reduce idle listening. The authors show significant energy savings with respect to RI-MAC.

The way beacon adaptivity is realized is the following. Whenever a sender transmits data, it also piggybacks its next beacon time inside the packet. Upon reception of such data packet, the receiver is able to adjusts its next beacon time consequently and move from RP to SP. Should no more packets arrive due to channels errors or collisions, receivers use a T_{cycle} value (also piggybacked on data packets) to estimate the transmission time of future beacons.

3.4.5. Receiver-Initiated MAC for Energy Harvesting WSN (ERI-MAC) [36]

ERI-MAC is a typical incarnation of the receiver-initiated paradigm: message queues, beacons used as ACKs for following messages and back-off windows to mitigate collisions. One of the optimization proposed is the *packet concatenation*, which allows for small packets to be merged into a single bigger packet. Another extension of ERI-MAC is the possibility of queuing packets for a given amount of time before sending them out. The value of this timeout can be either fixed or, in case of energy harvesting WSNs, based upon the ratio between the energy consumed and the energy harvested. This is done to reach and maintain the energy neutral operation (ENO) [32].

3.5. Quality of Service

The protocols that focus on Quality of Service (QoS) provide services that prioritize the traffic according to the needs of the overlaying application.

3.5.1. Delay Bounded MAC (CyMAC) [37]

CyMAC focuses on delay-sensitive applications and attempts to provide data delivery guarantees. This builds upon a unique feature introduced by CyMAC. In CyMAC, the beacons are dedicated for each neighboring sender. Thus, the period of each individual beacon can be independently adapted on a per-sender basis. The conduced comparison with RI-MAC suggests that CyMAC can provide delay

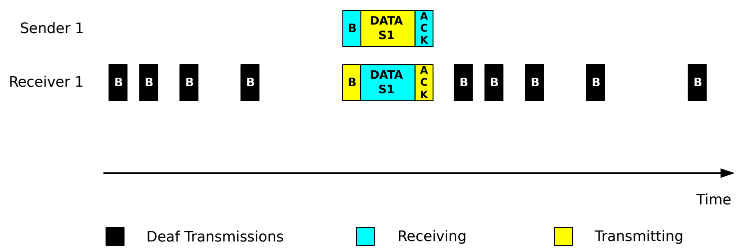

Figure 11: Traffic dependent beaconing pattern as shown in CyMAC, beacons become sparser over time whenever there is no sender to serve, and reset as soon as a new one is found.

guarantees under various traffic conditions. Except for cases of tight required delay bounds, CyMAC yields lower duty cycles than RI-MAC.

The protocol also introduces a dynamic duty cycle adaptation mechanism that aims to adjust the sleeping schedules to the given traffic conditions. Thus, when the traffic is light, sensor nodes sleep more and conserve more energy, while when the traffic is heavy, they broadcast more beacons to increase the performance. The duty cycle adaptation algorithm operates as follows. All nodes operate at a maximum duty cycle and as long as they don't serve any traffic they exponentially increase the time between two beacons. The exponential increase continues until a data packet arrives, which triggers the node to reset the duty cycle period to its minimum value. This behavior is shown in Figure 11.

3.5.2. QoS Aware Energy-Efficient MAC (QAEE-MAC) [38]

QAEE-MAC extends the receiver-initiated paradigm with a mechanism that allows priority data to be transmitted faster than normal data. Upon waking up, each sender transmits a control packet, named *Tx-beacon*, which indicates the priority of its data packet. Before the beacon transmission, the receiver wakes up and collects *Tx-beacon* packets. Then, it uses the priority information to determine to which node to transmit to. However, such support for priority packets comes at the cost of extending the idle listening time of all the involved senders.

3.5.3. Asynchronous MAC with QoS Awareness (AQ-MAC) [39]

AQ-MAC is another asynchronous protocol that promotes quality of service. It allows for two different levels of priority for a packet: *high* and *low*. Whenever a high priority packet is received, it is forwarded right away, independent of the current status of the node. On the contrary, for low priority messages the node behaves as a non-sender and stores the received packets inside an internal queue. When the transmission timeout expires, the node sends all the queued packets in a burst, using a concatenation scheme. This allows the node to save the overhead connected with having to send multiple small packets. The transmission timeout is fixed and is set according to the distance between the node and the sink. Collisions are handled in the same manner of RI-MAC, after a collision a sender will transmit a new beacon containing the value of a back-off window.

3.6. Broadcast Support

In asynchronous duty cycling sensor networks, broadcasting constitutes a challenge because nodes are not awake concurrently. For applications and protocols that require broadcasting services, MAC protocols have been enriched with mechanisms to support them.

3.6.1. Asynchronous Duty cycle Broadcasting (ADB) [40]

ADB extends RI-MAC with support for broadcasting. Similarly to unicasting, broadcasting is initiated by the receiver. Therefore, the procedure is equivalent to a series of unicast transmissions. ADB avoids transmissions over poor links, by entrusting the packet that needs to be broadcasted to other nodes. The sender tracks the procedure by maintaining two lists of neighboring nodes (those who received the broadcasted packet and those who are assigned to other nodes) and goes to sleep once all its neighbors are marked in either of these lists. Consider the example that a sender S wants to broadcast a frame to $R1$ and $R2$ and assume that the quality of the link between S and $R2$ is poor, while the link between $R1$ and $R2$ is good. After the transmission of the packet from S to $R1$, the receiver $R1$ takes the responsibility of forwarding the packet to $R2$. The coordination of the procedure, which includes the information of which nodes are pending and the quality of the respective links, is achieved by control data that is piggy-backed on the beacons and data frames.

3.6.2. Yet Another MAC (YA-MAC) [41]

In YA-MAC, the nodes go through an initialization phase in which all nodes are on 100% duty cycles. During this phase, they determine their neighborhood and agree on some protocol parameters. One of these parameters is the broadcast time interval, which defines the period of a broadcast slot. All nodes wake-up during the broadcast slot, which makes normal broadcasting feasible. The nodes are loosely synchronized. In particular, the Synchronization Error Tolerance Window (SETW) defines a guard time interval that protects the system from minor clock drifts. If the synchronization falls below a desired level, nodes are triggered to enter a 100% duty cycle phase during which synchronization is re-established.

Lastly, YA-MAC uses the amount of neighboring nodes, as it is determined in the initialization phase, to select the contention window for collision avoidance.

3.6.3. Receiver Wake-up Broadcast (RWB) [42]

RWB extends RW-MAC [26] with broadcast support. Similarly to ADB [40], a broadcast transmission consists of a series of unicast transmissions. The key difference to ADB is that packets are not delegated to other nodes. Instead, RWB uses the wake-up prediction mechanism of RW-MAC to optimize the performance. Moreover, the individual unicast transmissions that compose a broadcast transmission can be optionally acknowledged to optimize the delivery ratio.

3.7. Multi-Channel Extensions

Exploiting multiple channels increases the capacity of a link. Hence, it can lead to higher throughput, fewer collisions and shorter delays in networks with relatively high traffic. On the other hand, overlapping WSNs that are using multi-channel MAC protocols, interfere with each other, as they cannot be tuned to different orthogonal channels. Wireless sensor nodes are also typically limited by a single radio unit. As a result, MAC protocols cannot operate at multiple channels concurrently in order to transmit and receive in parallel. A series of multi-channel MAC protocols that are using the receiver-initiated paradigm are surveyed next.

3.7.1. Duty Cycle Multi-channel MAC (DCM) [43]

DCM defines three types of channels, namely a single Control Channel (CC), a series of Data channels (DC) and a single Broadcast Channel (BC). Normal unicast communication is executed as follows (Figure 12). A sender that wants to transmit is actively listening to the CC for incoming beacons, named *Announcements (ANC)*. When ready to receive, the receiver transmits an ANC on the CC. The ANC frame includes the number of a DC which is selected by the receiver randomly. The authors claim that due to duty cycling and the single-radio limitation, random channel selection is a better choice to information-based selection. Right after the ANC transmission, the receiver is switching to the selected DC and listening for a Ready To Send (RTS) frame. Right after the reception of the expected ANC, the sender also switches to the announced channel. The communication then follows a typical RTS - CTS - DATA - ACK communication. Random backoff is also included for avoiding collisions between multiple nodes that received the same ANC. If a node finds the CC or the specified DC busy for multiple times, it assumes that the network is congested and goes back to sleep.

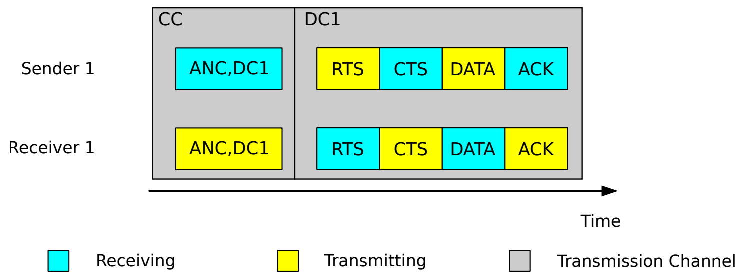

Figure 12: Multi-channel approach in DCM. A control channel (*CC*) is used to choose a specific data channel (*DCx*) where the communication will be carried through.

Moreover, a sender that is not receiving an ANC for a predetermined period of time is transmitting an ANC in CC in order to avoid deadlocks.

DCM also provides multi-channel broadcast support via the BC. Whenever a sender wants to broadcast a frame, it switches to the broadcast channel and after a CCA it transmits the broadcast frame for M consecutive time intervals. Every node, no matter how the duty cycles are configured, has to switch to the BC and check for possible incoming broadcast data once every $M - 1$ time intervals. The value of M can control the trade-off between energy efficiency and broadcast latency.

Asynchronous Receiver-initiated Multi-channel MAC (ARM) [44] constitutes a follow-up publication by DCM's main authors and it extends its analysis and evaluation. However, there are no significant changes in the core of the protocol. ARM operates similarly to DCM.

3.7.2. Efficient Multi-channel MAC (EM-MAC) [45]

EM-MAC is a multi-channel MAC protocol that does not use a common control channel as the channel numbers and wake-up schedules are not explicitly exchanged. Instead, every node generates a channel number and a time for the next wake-up event using a shared pseudo-random number generator. Every node is able to predict the next wake-up event of any other node just by knowing the *prediction state*. The prediction state includes the information of the random seed, a previous wake-up time, a multiplier a and a constant c. A node that does not have the prediction state of a given receiver, listens for a beacon on the first channel, which contains the corresponding information. Additionally, each node maintains the status of each channel by counting the failed CCAs. If the status metric exceeds a certain threshold the channel is blacklisted and is not used. If the pseudo-random number generator chooses a blacklisted channel, the node stays on the previous channels. Blacklisted channels are advertised using a bitmap on the beacons.

The rest of the protocol operation is based on the receiver-initiated paradigm. Different from RI-MAC, EM-MAC puts sender to sleep if the collision resolution mechanism does not resolve the collision before the receiver goes back to sleep. The ability to predict the next wake-up through the pseudo-random generator, allows the node to sleep and save energy in the meantime.

3.8. Hybrid approach

The protocols listed here feature a more hybrid approach when compared to the previous ones. The key elements of the receiver-initiated world are usually present in some form, however they coexist with mechanics typical of sender-initiated or synchronous schemes.

3.8.1. RI-MAC with Scheduled Wake-up Instants (SWI-RI-MAC) [46]

In SWI-RI-MAC, the authors propose to enhance the RI paradigm so that all nodes wake-up at approximately the same instant. In each wake-up period only one node transmits a beacon. The key challenge, i.e. the coordination of the nodes, is addressed through distributed learning for collision-free operation, where a node randomly picks a wake-up period in the schedule and keeps transmitting at the same instant in the following cycles if its transmission is successful. More specifically the protocol used is the *Learning Zero Collision* [47] in which a node keeps track of the free slots in the schedule; after a collision it changes to one of such free slots with probability $(1 - \gamma)$ and remains in the same one with probability γ, where γ is a system parameter.

According to the authors, the synchronization of the wake-up events amongst the nodes can be obtained by including specific information within the beacons and by allowing nodes to wake-up during a guard time before the expected beacon reception to account for clock drift.

The main advantage of SWI-RI-MAC is that broadcast can be implemented with small effort since all the nodes are synchronized. Furthermore, energy savings can be achieved thanks to the inherently low number of collisions achieved by synchronous schemes.

3.8.2. Synchronous Receiver-Initiated MAC (SRI-MAC) [48]

SRI-MAC is a synchronous duty cycle protocol with receiver-initiated data transmission. The typical message exchange within SRI-MAC can be seen in Figure 13 is the following: first a receiver sends out a beacon announcing that it

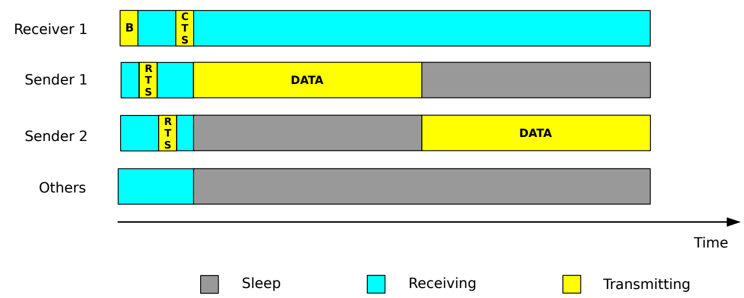

Figure 13: Mechanics of SRI-MAC, the order of the nodes is announced within the *CTS* packet, and uninterested nodes can go to sleep.

is awake and ready to receive data. This packet contains the *id* of the receiver and the *Duration Allocation Period* (DAP), which depends on the number of receiver neighbors. This values is used as a common factor to generate back-off values for collision avoidance. Upon receiving a beacon, each sender sends out a RTS packet containing the *id* of the node, the *id* of the intended receiver and the data size. At this point the receiver transmits a CTS packet which is used to assign a time slot to each sender that has previously registered itself through a RTS. At this point the communication period starts and each sender wakes up according to the predetermined order. In case where there is no sender for a particular receiver, its beacon will go unanswered (no CTS will be sent), and after time equal to the DAP specified in the beacon the channel will be considered idle by other potential receivers.

Energy savings can be achieved through this approach because each sender has to wake-up only during the allocation period and during the assigned slot of the communication period. Furthermore, nodes that are not taking part to the communication only have to participate to initial information period where the receiver announces itself, and then can stay asleep for the reminder of the time slot.

3.8.3. Asymmetric Links (Asym-MAC) [49]

Asym-MAC is a hybrid protocol that dynamically switches between two modes of operation: sender-initiated and receiver-initiated, depending on the asymmetry of the link. The motivation given by the authors is that receiver-initiated MAC protocols perform poorly when asymmetric links are present. The default mode of operation of Asym-MAC is the so called *R-mode* which operates using the classic RI scheme. Whenever a sender fails to receive a beacon from the intended receiver for more than τ times, it switches to the *T-mode* and starts sending pream-

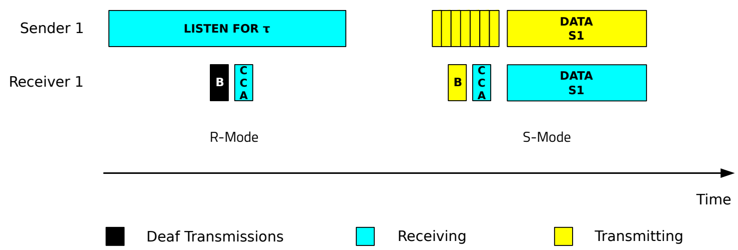

Figure 14: Mechanics of Asym-MAC, switching from R-Mode to S-Mode. Whenever a beacon is not received for more than τ, the sender switches mode and starts transmitting preambles.

bles. There is no explicit communication of this change, rather the receiver node performs a CCA for a small period of time at the end of each beacon transmission in order to check whether or not there is an ongoing preamble transmission. Should this be the case, the receiver node also switches to the *T-mode*. Collisions can happen between beacons from a receiver in *R-mode* and preambles form a sender in *T-mode*. To address this, after a collision has happened, the receiver performs additional CCAs to increase the chance of receiving preambles.

Finally, the asymmetry of a link is measured through the analysis of the *Packet Reception Ratio* (PRR): given a fixed beaconing period it is possible to determine the number of packet sent and the theoretical number of packet that should have been sent if all the beacons were received.

The mechanics of Asym-MAC are shown in Figure 14.

3.9. Security

TinySec [50] is a security suite for WSNs that provides important services such as data integrity and confidentiality at link level. TinySec is fully compatible with the receiver-initiated paradigm. However, receiver-initiated MAC protocols are keen to beacon replay attacks. A replay attack is defined as an attack against a protocol where previously exchanged messages are reused in order to fool legitimate participants into thinking that the current run of the protocol is valid and exchanged data is fresh [51]. Beacons contain the identity of their creator which is the main piece of information needed to determine whether or not a specific beacon can be used by a potential sender, according to the overlying routing algorithm. By replaying beacons, it is possible to deploy a series of other attacks.

Receiver Authentication Protocol (RAP) [52]: RAP is a challenge-response authentication protocol that aims at authenticating the receiver, i.e. the beacon transmitter, in a receiver-initiated data transmission. It has two modes of operation, namely *detection* and *prevention* mode. The detection mode (RAP-D) is

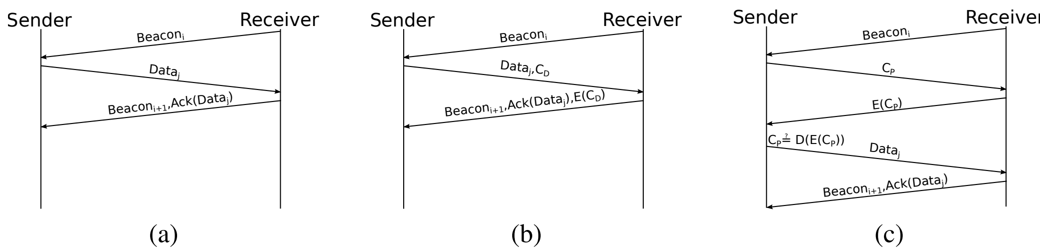

Figure 15: A typical receiver-initiated protocol (a), the inexpensive detection phase of RAP-D (b), and the prevention phase of RAP-P (c).

a low overhead scheme and aims at detecting an intruder that replays beacons without preventing it from doing so. The prevention mode (RAP-P), on the other hand, is a more costly scheme that prevents the attack altogether. The difference between the two modes is the timing of the challenge-response message exchange. In RAP-P, the challenge-response message exchange takes place before the data transmission. Thus, the sender transmits the data packet only if the receiver is authenticated. The low overhead nature of RAP-D, on the other hand, is maintained by piggybacking the challenge and its response on top of the frames normally exchanged in the MAC protocol. In other words, the authentication of the receiver takes place after the data transmission. Having energy efficiency as a primary system priority, the idea is that a node normally operates at the low overhead detection mode and switches to the expensive prevention mode only if necessary.

4. Real World Deployment

MAC protocols in wireless sensor networks are hardly standardized, the only exceptions being *Bluetooth Low Energy*, *IEEE 802.15.4* and the protocols based upon that. This is probably one of the main reasons why real world deployment are hard to come by, industries shy away from non-standardized platforms and most of the known implementations are either research projects or military applications which can both justify the use of highly specialized components. Receiver-initiated protocols are the last addition to the scene, and are far from being standardized. Nevertheless, there exist some real world applications (mostly still under development) that make use of RI protocols. Here we will present some of them in a short, non-exhaustive list, in order to point out what is possible to achieve with real implementations.

4.1. Environmental Monitoring [53]

Environmental monitoring is a typical application for wireless sensor networks. The goal is to collect data from the environment such as temperature, barometric pressure or rainfall levels, over a fairly long period of time in order to build historical records. The reason because WSNs are a good match for this kind of applications is because they are for the most part inexpensive, easy to deploy and require very low maintenance. However, in order to cover large geographical areas, sophisticated routing protocols must be implemented and forwarding the packets toward a common collection point quickly becomes a bottleneck from an energy standpoint.

An alternative to multi-hop network is the one used in the work under discussion [53], which consists of having mobile collectors that travel around the area of interest directly communicating to nodes. This approach solves the routing problem but has to deal with other non-trivial problems such as geo-locating each node and computing the best path to visit them.

To provide more details, autonomous mobile robots (called *data mules*) are equipped with RF transceivers and move through the network to download data from the sensing nodes. In order to perform the data exchange, the RI-based LPP mechanism of [13] is used. A data mule will approach a node, wait for a beacon, acknowledge it and issue a command to begin the data download. This allows each node to select its own duty cycle. Furthermore, if the mule passage happens at a predefined time, the node can synchronize and avoid sending unanswered beacons.

The original paper presents different exploration strategies based on approximations of the Traveling Salesperson Problem, and also a proof deployment.

4.2. Passenger Flow in Airport Terminals [54]

The second application that we are going to describe is a *Digital Boarding Assistance*, or *DigiBa*, that is meant to help passengers with the boarding process. Upon their arrival at the airport, passengers are lent an inexpensive device equipped with a small screen, some interface buttons and a radio transceiver. This device is then used for several tasks. First of all it can be used to relay messages to a group of users (e.g. to inform the passengers of a specific flight that the gate number has changed) or to a single passenger (e.g. individual calls for late passengers). Furthermore, important messages can be accompanied by an acoustic signal and must be explicitly acknowledged by the user. On the other way individual passengers can use the DigiBa to request information on specific topics, and be informed when a related event triggers. Lastly the DigiBa can be used for navigation within the airport, guiding passengers to different locations such as their gate or to specific points of interest such as restaurants, shops and currency exchange. This project faces many interesting challenges like achieving low-power bidirectional communication, indoor localization and providing navigation assistance using simple patterns and easy to recognize architectural elements.

Focusing on the communication side of the system, a 2.4 GHz technology such as Bluetooth or WLAN is suggested, together with a number of base stations scattered throughout the airport. Additionally, what the authors propose is to use two different solutions for the up-link channel and the down-link channel. Namely CSMA is proposed for the up-link, this is because the base stations are always active and each DigiBa can transmit whenever it has to, the only concern is to avoid collisions with other devices. On the other hand, for the down-link, RI-MAC [1] is suggested as the protocol to use. Devices can send a beacon when they are awake, so that base stations in range can answer back with messages. This avoids the use of preambles from the base station and helps minimizing the transmission delay.

4.3. Hybrid Application in Metering Systems [55]

The third use case that we are going to discuss is related metering systems and smart grid. Here each meter is considered a node of the network and it can communicate with a central power station. The transmission model taken into account consists of messages sent by the meters to the central station every thirty minutes, with an acceptable delay of up to five minutes. Communication in the opposite direction is also required for maintenance purposes.

In order to keep the number of required central stations low, meters are allowed to talk amongst them and forward messages towards the central stations. The protocol of choice for this work is IRDT [19] which allows intermittent operation of the nodes and minimizes latency. Nevertheless, this approach facilitates saturation of the nodes in the neighborhood of a central station. When many messages have to be delivered toward the same location it becomes harder and harder to find an available receiver. To avoid this, the authors use a hybrid approach and make use of a sender-initiated approach for the nodes surrounding a central station.

4.4. Thermal Energy Harvesting [56]

The final use case discussed here is related to energy harvesting applications. Specifically, power (heat) meters attached to household radiators are used as cost allocators for the measured consumption. The collection of this information happens automatically and wirelessly through advanced metering infrastructures. Similarly to the previous use case a bi-directional communication is required between meters and collection gateway nodes. In these way consumption details can be collected and, possibly, analyzed in order to offer on-the-fly adjustments which can improve the efficiency of the system and lower consumption.

The approach used by the authors is to have the cost allocator devices harvest thermal energy from the radiator itself, using such energy to power the device, which would otherwise be battery operated. This drastically reduces the amount of power available to each node, which has direct consequences on the transmission model of the whole system. The solution used to address this problem is to use receiver-initiated MAC protocols. Specifically the protocol of choice is ODMAC [31] which is specifically designed to operate with energy harvesting networks and allows each node to autonomously and independently select its own duty cycle according to the available energy reservoir.

5. Reflection

All the protocols surveyed in Section 3 define mechanisms and features that can be added to the basic paradigm to optimize its performance. It should be noted that such features can be used in different combinations beyond the definition of each individual protocol. Depending on the properties of a specific application a network engineer can combine features, introduced by different protocols, to optimize the overall performance of the system. We see this survey as a tool for enabling developers and researchers to more easily choose such features and their relative implementation by having a quick reference of what has been already done in literature, how it has been done and what can be achieved.

Sensor networks are mainly characterized by the limited resources of its nodes. A holistic network design is vital for the efficient use of the limited resources. The MAC protocol, as a fundamental part of the networking stack, should be configured with respect to the topological structure of the network, the power source of the nodes and the characteristics and requirements of the running application.

A key design decision is between static and adaptive duty cycles, as many of the presented features are not compatible with both. Adaptive duty cycles are expected to be beneficial only in dynamic network conditions, as they would introduce overhead otherwise. The energy profile of the nodes, which is the combination of the energy input and energy consumption profile, plays a key role.

When the energy profile of the nodes of the system is unbalanced, static duty cycles would introduce bottlenecks in the network. A balanced energy consumption profile implies a carefully designed static topology and stable traffic generation, in such a way that the duties of all nodes are balanced. A balanced energy input profile implies that the nodes are powered by batteries with similar energy resources. In this case, significant energy can be saved by predicting the upcoming wake-up using a backup prediction scheme that assumes static duty cycles, like WideMAC [24], Pseudo [25] and PW-MAC [27]. If there are no other networks deployed in the same area, multiple channels can further increase the performance (EM-MAC [45]).

In the opposite case, e.g. dynamic topologies, applications with bursty traffic or nodes that are powered by unpredictable sources of energy harvested from the environment, a dynamic duty cycle approach is recommended. In addition to using the specific adaptive duty cycle features when relevant (Stair [30], CyMAC [37], ODMAC [31], SARI-MAC [34] and HKMAC [35]), idle listening can be reduced either by predicting the next wake-up using the approach of RW-MAC [26], by using multiple receivers as described by the opportunistic forwarding mechanism of ODMAC [31] or by using the duty cycled listening approach of EE-RI-MAC [20]. Moreover, if the use of multiple channels is possible, the approach described in DCM [43] can be adopted.

Independent of how the duty cycling is organized, the beacon acknowledgement proposed by A-MAC [21] mitigates the cost of beaconing. In case any form of wake-up prediction mechanism is used, this information can be used to optimize the transmission buffer as the frame reordering feature of RP-MAC [28] defines. If, on the other hand, no wake-up prediction mechanism is used, the operation cycles of REA-MAC [22] reduce the idle listening, while the opportunistic cooperation, proposed by OC-MAC [17], and the altruistic backoff of ODMAC [31] handle collisions in a way that also mitigates idle listening. Otherwise, Binary Exponential Backoff (BEB), as described in RI-MAC [1] or RC-MAC [18] can be used. Such methods constitute active collision avoidance mechanisms. In cases of very low traffic, random access via random beacon selection (e.g. IRDT [19]), would sufficiently handle collisions without the additional overhead.

The rest of the features provide services for the application or protocols at a higher level and, therefore, should only be used if these services are needed and the network is capable of handling the additional overhead. The approach of QAEE-MAC [38], CyMAC [37] and AQ-MAC [39] can be used for traffic differentiation and applications with priority requirements. Asym-MAC [49] can provide a solution in deployments where links appear to be highly asymmetric and SRI-MAC [48] can provide a solution in dense deployments that can benefits from elements of synchronization. TinySec [50] and RAP [52] can be used for applications with security requirements. For broadcast support, the approach of RWB [42] and WI-RI-MAC [46] can be used along a wake-up prediction mechanism, while ADB [40] or YA-MAC [41] can be used otherwise.

Tables 2 and 3 present all the surveyed protocols in a more compact way. More specifically, Table 2 provides a top-down approach, where each protocol is described and characterized in terms of its implemented features. On the other hand, Table 3 uses a complementary bottom-up organization, showing what technique is used to address each challenge and by which protocols it is implemented.

Protocol name	**Features summary**
A-MAC	Idle listening minimization, Collision avoidance
ADB	Broadcast
AsymMAC	Collision Avoidance, Hybrid: mode change
AQ-MAC	QoS, Frame reordering, Collision avoidance
CyMAC	Adaptive D/C, QoS
DCM	Multiple channels, Broadcast
EE-RI-MAC	Idle listening minimization
ERI-MAC	Adaptive D/C, Collision avoidance
EM-MAC	Wake-up prediction, Multiple channels
FERI-MAC	Wake-up prediction, Collision avoidance
HKMAC	Wake-up prediction, Idle listening minimization
IRDT	Collision avoidance
OC-MAC	Idle listening minimization, Cross-layer interaction, Collision avoidance
ODMAC	Adaptive D/C, Idle listening minimization, Cross-layer interaction, Collision avoidance
PW-MAC	Wake-up prediction
Pseudo	Wake-up prediction
QAEE-MAC	QoS, Idle listening minimization, Cross-layer interaction, Collision avoidance
RAP	Cross-layer interaction, Security
RC-MAC	Collision avoidance
REA-MAC	Cross-layer interaction Idle listening minimization

RI-MAC	Collision avoidance
RICER	Wake-up prediction, Cross-layer interaction
RP-MAC	Frame reordering, Collision avoidance
RW-MAC	Wake-up prediction Idle listening minimization, Collision avoidance
RWB	Broadcast
SARI-MAC	Adaptive D/C, Cross-layer interaction, Collision avoidance
Stair	Adaptive D/C
SRI-MAC	Collision avoidance, Idle listening minimization, Hybrid: synchronization
SWI-RI-MAC	Collision avoidance, Broadcast, Hybrid: synchronization
WideMAC	Wake-up prediction, Idle listening minimization, Collision avoidance
YA-MAC	Collision avoidance, Broadcast

Table 2: The above table provides the list of features that each protocol implements. It provides a $Protocol \rightarrow Features$ classification.

6. Conclusions

In this paper we have surveyed a number of receiver-initiated MAC protocols for WSNs classifying them according to their different properties. The main goal of this work is to provide the interested researcher with enough insight into each protocol so that further review of the relevant literature can be carried out autonomously.

Wireless sensor networks continue to be a hot topic and new and interesting ideas are being proposed constantly. As briefly discussed in Section 5, the point of view shared by the authors is that given the highly specialized applications for which WSNs are deployed, there is no such thing as *the* perfect solution. On the contrary, a specific technique could be very good in one scenario and disastrous in the next one. Alongside this, a strong integration and a tight interaction between the different components of a protocol, again dictated by the needs introduced by the overlying application, are key for the achievement of a successful solution. Under these assumptions, we think a survey is a good tool to list all the existing ideas, providing with a starting point from where required functionality can be mixed and matched in order to craft a solution that perfectly suits the need of the current application.

Challenge	Technique	Protocols
Idle listening	Wake-up prediction	EM-MAC, FERI-MAC, HKMAC, Pseudo, PR-MAC, PW-MAC, RICER, RW-MAC, SRI-MAC, WideMAC
	Beacon acknowledgment	A-MAC
	Duty cycle of listening	EE-RI-MAC, QAEE-MAC
	Cross-layer interaction	OC-MAC, ODMAC, REA-MAC, RICER
	Beacon period adaptation	FERI-MAC, HKMAC, IRDT, SARI-MAC
	Indirect	IRDT, ODMAC
Collision avoidance	Random backoff	A-MAC, AQ-MAC, DCM, ERI-MAC, EM-MAC, IRDT, OC-MAC, ODMAC, RC-MAC, RI-MAC, RICER, SRI-MAC, QAEE-MAC, WideMAC, YA-MAC
	Cooperation	FERI-MAC, OC-MAC, ODMAC
	Data aggregation	AQ-MAC, ERI-MAC, IRDT
	Beacon period adaptation	IRDT, SARI-MAC
	Time-slot reservation	SARI-MAC, SWI-RI-MAC
	Staggering	RW-MAC
	Multi-channel extensions	DCM, EM-MAC
	CCA Extension	AsymMac
Adaptive duty cycling	Traffic based	CyMAC, SARI-MAC
	Energy based	ERI-MAC, ODMAC
	Distance based	Stair
Quality of service	Frame reordering	AQ-MAC, CyMac, QAEE-MAC
Broadcast	Synchronization	DCM, SWI-RI-MAC, YA-MAC
	Multiple unicasts	ADB, RWB
Asymmetric Links	Mode change	AsymMac
Security	Authentication	RAP

Table 3: A specific challenge might be addressed by each protocol in a different way. The above table provides a $Challenge \rightarrow Protocols$ approach, a complementary view to Table 2.

References

[1] Y. Sun, O. Gurewitz, D. B. Johnson, RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks, in: Proc. 6th ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys), ACM, 2008, pp. 1–14.

[2] A. Bachir, M. Dohler, T. Watteyne, K. Leung, MAC Essentials for Wireless Sensor Networks, IEEE Commun. Surveys Tutorials 12 (2010) 222–248.

[3] W. Ye, J. Heidemann, D. Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Networks, in: Proc. 21st Ann. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM), volume 3, IEEE, 2002, pp. 1567–1576.

[4] T. V. Dam, K. Langendoen, An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, in: Proc. 1st ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys), ACM, 2003, pp. 171–180.

[5] P. Lin, C. Qiao, X. Wang, Medium Access Control With A Dynamic Duty Cycle For Sensor Networks, in: Proc. IEEE Wireless Commun. and Networking Conf. (WCNC), volume 3, IEEE, 2004, pp. 1534–1539.

[6] A. El-Hoiydi, J.-D. Decotignie, WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks, in: Proc. 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS), 2004, pp. 18–31.

[7] J. Polastre, J. Hill, D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks, in: Proc. 2nd ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys), ACM, 2004, pp. 95–107.

[8] M. Buettner, G. V. Yee, E. Anderson, R. Han, X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks, in: Proc. 4th ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys), ACM, 2006, pp. 307–320.

[9] C. Cano, B. Bellalta, A. Sfairopoulou, M. Oliver, Low energy operation in WSNs: A survey of preamble sampling MAC protocols, Computer Networks 55 (2011) 3351–3363.

[10] E.-Y. A. Lin, J. M. Rabaey, A. Wolisz, Power-efficient rendez-vous schemes for dense wireless sensor networks, in: Proc. IEEE Int. Conf. on Commun. (ICC), volume 7, IEEE, 2004, pp. 3769–3776.

[11] X. Fafoutis, N. Dragoni, Analytical Comparison of MAC Schemes for Energy Harvesting-Wireless Sensor Networks, in: Proc. Int. Workshop on Algorithms and Concepts for Networked Sensing Systems Powered by Energy Harvesters (EnHaNSS), IEEE, 2012, pp. 1–6.

[12] C. M. Vigorito, D. Ganesan, A. G. Barto, Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks, in: Proceedings 4th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), IEEE, 2007, pp. 21–30.

[13] R. Musaloiu-E., C.-J. M. Liang, A. Terzis, Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks, in: Proc. of the 7th Int. Conf. on Inform. Process. in Sensor Networks, IEEE Computer Society, 2008, pp. 421–432.

[14] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, Tinyos: An operating system for sensor networks, in: Ambient Intelligence, Springer Berlin Heidelberg, 2005, pp. 115–148.

[15] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, D. Culler, Exploiting The Capture Effect For Collision Detection And Recovery, in: Proc 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II), IEEE, 2005, pp. 45–52.

[16] IEEE, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standards Association, 2012.

[17] X. Wang, Q. Zhang, Opportunistic Cooperation in Low Duty Cycle Wireless Sensor Networks, in: Proc. IEEE Int. Conf. on Commun. (ICC), IEEE, 2010, pp. 1–5.

[18] P. Huang, C. Wang, L. Xiao, H. Chen, RC-MAC: A Receiver-Centric Medium Access Control Protocol for Wireless Sensor Networks, in: Proc. 18th Int. Workshop on Quality of Service (IWQoS), IEEE, 2010, pp. 1–9.

[19] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, Energy-Efficient Receiver-Driven Wireless Mesh Sensor Networks, MDPI Sensors 11 (2010) 111–137.

[20] Y.-T. Yong, C.-O. Chow, J. Kanesan, H. Ishii, EE-RI-MAC: An energy-efficient receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks, International Journal of Physical Sciences 6 (2011) 2633–2643.

[21] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, A. Terzis, A-MAC: A versatile and efficient receiver-initiated link layer for low-power wireless, ACM Trans. Sensor Networks 8 (2012) 30:1–30:29.

[22] H.-w. Tang, J.-n. Cao, C.-x. Sun, K. Lu, REA-MAC: A low latency routing-enhanced asynchronous duty-cycle MAC protocol for wireless sensor networks, Journal of Central South University 20 (2013) 678–687.

[23] X. Fafoutis, A. D. Mauro, N. Dragoni, Sustainable Medium Access Control: Implementation and Evaluation of ODMAC, in: Proc. 4th Workshop on Energy Efficiency in Wireless Networks and Wireless Networks for Energy Efficiency (E2Nets), IEEE, 2013, pp. 407–412.

[24] J. Rousselot, A. El-Hoiydi, J.-D. Decotignie, WideMac: a Low Power and Routing Friendly MAC Protocol for Ultra Wideband Sensor Networks, in: Proc. IEEE Int. Conf. on Ultra-Wideband (ICUWB), volume 3, IEEE, 2008, pp. 105–108.

[25] H. Lee, J. Hong, S. Yang, I. Jang, H. Yoon, A Pseudo-Random Asynchronous Duty Cycle MAC Protocol in Wireless Sensor Networks, IEEE Commun. Lett. 14 (2010) 136–138.

[26] D. Yang, Y. Qiu, S. Li, Z. Li, RW-MAC: An asynchronous receiver-initiated ultra low power MAC protocol for Wireless Sensor Networks, in: Proc. IET Int. Conf. on Wireless Sensor Network (IET-WSN), IET, 2010, pp. 393–398.

[27] L. Tang, Y. Sun, O. Gurewitz, D. B. Johnson, PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks, in: Proc. 30th Ann. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM), IEEE, 2011, pp. 1305–1313.

[28] Q. Hu, Q. Tian, Z. Tang, RP-MAC: A Passive MAC Protocol with Frame Reordering for Wireless Sensor Networks, Int. J. of Wireless Inform. Networks 20 (2013) 74–80.

[29] L. Pu, Y. Luo, Z. Peng, H. Mo, J.-H. Cui, Traffic estimation based receiver initiated mac for underwater acoustic networks, in: Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems, ACM, New York, NY, USA, 2013, pp. 1–5.

[30] T. Wada, I.-T. Lin, I. Sasase, Asynchronous Receiver-Initiated MAC Protocol with the Stair-Like Sleep in Wireless Sensor Networks, in: Proc. 22nd Ann. IEEE Int. Symp. on Personal Indoor and Mobile Radio Commun. (PIMRC), IEEE, 2011, pp. 854–858.

[31] X. Fafoutis, N. Dragoni, ODMAC: An On-Demand MAC Protocol for Energy Harvesting - Wireless Sensor Networks, in: Proc. 8th ACM Symp. on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), ACM, 2011, pp. 49–56.

[32] A. Kansal, J. Hsu, S. Zahedi, M. B. Srivastava, Power Management in Energy Harvesting Sensor Networks, ACM Trans. on Embedded Computing Syst. (TECS) 6 (2007) 32.

[33] X. Fafoutis, C. Orfanidis, N. Dragoni, Altruistic Backoff: Collision Avoidance for Receiver-Initiated MAC Protocols for Wireless Sensor Networks. Manuscript submitted for publication, Int. J. of Distributed Sensor Networks (2013).

[34] Q. Lampin, D. Barthel, I. Auge-Blum, F. Valois, SARI-MAC: The Self Adapting Receiver Initiated MAC protocol for Wireless Sensor Networks, in: Proc. 8th IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Commun. (WiMob), IEEE, 2012, pp. 12–18.

[35] H. Tang, C. Sun, Y. Liu, B. Fan, Low-latency asynchronous duty-cycle mac protocol for burst traffic in wireless sensor networks, in: Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, 2013, pp. 412–417.

[36] K. Nguyen, V.-H. Nguyen, D.-D. Le, Y. Ji, D. Duong, S. Yamada, A receiver-initiated mac protocol for energy harvesting sensor networks, in:

Y.-S. Jeong, Y.-H. Park, C.-H. R. Hsu, J. J. J. H. Park (Eds.), Ubiquitous Information Technologies and Applications, volume 280 of *Lecture Notes in Electrical Engineering*, Springer Berlin Heidelberg, 2014, pp. 603–610.

[37] Y. Peng, Z. Li, D. Qiao, W. Zhang, Delay-Bounded MAC with Minimal Idle Listening for Sensor Networks, in: Proc. 30th Ann. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM), IEEE, 2011, pp. 1314–1322.

[38] S. Kim, J. Jeon, H. Park, QoS Aware Energy-Efficient (QAEE) MAC Protocol for Energy Harvesting Wireless Sensor Networks, in: Convergence and Hybrid Inform. Technology, volume 7425, Springer Berlin Heidelberg, 2012, pp. 41–48.

[39] K. Nguyen, Y. Ji, Asynchronous mac protocol with qos awareness in wireless sensor networks, in: Global Communications Conference (GLOBECOM), 2012 IEEE, 2012, pp. 555–559.

[40] Y. Sun, O. Gurewitz, S. Du, L. Tang, D. B. Johnson, ADB: An Efficient Multihop Broadcast Protocol based on Asynchronous Duty-cycling in Wireless Sensor Networks, in: Proc. 7th ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys), ACM, 2009, pp. 43–56.

[41] P. Yadav, J. A. McCann, YA-MAC: Handling Unified Unicast and Broadcast Traffic in Multi-hop Wireless Sensor Networks, in: Proc. 7th IEEE Int. Conf. and Workshops on Distributed Computing in Sensor Syst. (DCOSS), IEEE, 2011, pp. 1–9.

[42] Y. Qiu, S. Li, D. Yang, Z. Li, RWB: An Efficient Receiver-Initiated Single-Hop Broadcast Protocol for Asynchronous MAC in Wireless Sensor Networks, Recent Advances in Comput. Sci. and Inform. Eng. 127 (2012) 261–266.

[43] J. Li, D. Zhang, L. Guo, DCM: A Duty Cycle Based Multi-channel MAC Protocol for Wireless Sensor Networks, in: IET Int. Conf. on Wireless Sensor Network (IET-WSN), 2010, pp. 233–238.

[44] J. Li, D. Zhang, L. Guo, S. Ji, Y. Li, ARM: an Asynchronous Receiver-initiated Multichannel MAC Protocol with Duty Cycling for WSNs, in: Proc. 29th IEEE Int. Performance Computing and Commun. Conf. (IPCCC), IEEE, 2010, pp. 114–121.

[45] L. Tang, Y. Sun, O. Gurewitz, D. B. Johnson, EM-MAC: A Dynamic Multi-channel Energy-Efficient MAC Protocol for Wireless Sensor Networks, in: Proc. 12th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), ACM, 2011, p. 23.

[46] C. Cano, D. Malone, B. Bellalta, J. Barcelo, On the improvement of receiver-initiated mac protocols for wsns by applying scheduling, in: IEEE 14th Int. Sympsium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013, pp. 1–3.

[47] M. Fang, D. Malone, K. Duffy, D. Leith, Decentralised learning macs for collision-free access in wlans, Wireless Networks 19 (2012) 83–98.

[48] S. Boulfekhar, M. Benmohammed, Synchronous receiver initiated MAC protocol for long-lived sensor networks, Computers & Electrical Engineering 40 (2014) 504–516.

[49] M. Won, T. Park, S. H. Son, Asym-mac: A mac protocol for low-power duty-cycled wireless sensor networks with asymmetric links, Communications Letters, IEEE 18 (2014) 809–812.

[50] C. Karlof, N. Sastry, D. Wagner, Tinysec: a link layer security architecture for wireless sensor networks, in: Proc. 2nd ACM Int. Conf. on Embedded Networked Sensor Syst. (SenSys), ACM, 2004, pp. 162–175.

[51] D. E. Denning, G. M. Sacco, Timestamps in key distribution protocols, Commun. ACM 24 (1981) 533–536.

[52] A. D. Mauro, X. Fafoutis, S. Mödersheim, N. Dragoni, Detecting and Preventing Beacon Replay Attacks in Receiver-Initiated MAC Protocols for Energy Efficient WSNs, in: Proc. of the 18th Nordic Conf. on Secure IT Syst. (NordSec), 2013, pp. 1–16.

[53] O. Tekdas, V. Isler, J. H. Lim, A. Terzis, Using mobile robots to harvest data from sensor fields, Wireless Communications, IEEE 16 (2009) 22–28.

[54] B. Greßmann, H. Klimek, V. Turau, Supporting intelligent passenger flows in airport terminals, 2011.

[55] M. Sugano, Combining sender- and receiver-driven mac protocols for a large-scale metering system based on potential routing, in: Emerging Technologies for a Smarter World (CEWIT), 2013 10th International Conference and Expo on, 2013, pp. 1–6.

[56] M. D. Vithanage, X. Fafoutis, C. B. Andersen, N. Dragoni, Medium Access Control for Thermal Energy Harvesting in Advanced Metering Infrastructures, in: Proc. of IEEE Eurocon, to be published, IEEE, 2013, pp. 291–299.