The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey

Pei Huang, Li Xiao, *Senior Member, IEEE,* Soroor Soltani, *Student Member, IEEE,* Matt W. Mutka, and Ning Xi, *Fellow, IEEE,*

***Abstract*—Wireless Sensor Networks (WSNs) have become a leading solution in many important applications such as intrusion detection, target tracking, industrial automation, smart building and so on. Typically, a WSN consists of a large number of small, low-cost sensor nodes that are distributed in the target area for collecting data of interest. For a WSN to provide high throughput in an energy-efficient way, designing an efficient Medium Access Control (MAC) protocol is of paramount importance because the MAC layer coordinates nodes' access to the shared wireless medium. To show the evolution of WSN MAC protocols, this article surveys the latest progresses in WSN MAC protocol designs over the period 2002-2011. In the early development stages, designers were mostly concerned with energy efficiency because sensor nodes are usually limited in power supply. Recently, new protocols are being developed to provide multi-task support and efficient delivery of bursty traffic. Therefore, research attention has turned back to throughput and delay. This article details the evolution of WSN MAC protocols in four categories: asynchronous, synchronous, frame-slotted, and multichannel. These designs are evaluated in terms of energy efficiency, data delivery performance, and overhead needed to maintain a protocol's mechanisms. With extensive analysis of the protocols many future directions are stated at the end of this survey. The performance of different classes of protocols could be substantially improved in future designs by taking into consideration the recent advances in technologies and application demands.**

***Index Terms*—Wireless sensor networks, MAC protocols, energy efficiency, time-critical, TDMA, multichannel.**

I. INTRODUCTION

ADVANCES in microelectronics led to the development of low-cost tiny sensor nodes that are equipped with sensing, processing, and communication units. A broad range of applications such as precision agriculture, environment monitoring, intrusion detection, target tracking, and etc. are facilitated through networking these sensor nodes [1]. Many valuable applications have been demonstrated in the SENSEI [2] project, which is developed by the European FP7-ICT. Since the sensor nodes are expected to operate autonomously with small batteries for a number of months or years, energy efficiency is a fundamental criterion in the design of WSN protocols. A major power consuming component of a sensor node is the radio, which is controlled by the MAC protocol. Therefore, an efficient MAC protocol increases the lifetime of a sensor network to a great extent. In addition, the MAC layer controls how nodes share the wireless medium. An efficient MAC protocol can reduce collisions and increase the achievable throughput, providing flexibility for various applications.

When developing new protocols, it is necessary to investigate all the previous studies thoroughly. There has been a tremendous amount of research on the design and implementation of MAC protocols in WSNs. Hence, surveys of WSN MAC protocols are conducted to summarize the varieties of designs and implementations. The survey in [3] reviews several early designed MAC protocols based on their medium access strategies: random access or static access. Later, the work in [4] analyzes advantages and disadvantages of some recently proposed WSN MAC protocols. A more comprehensive study in [5] classifies WSN MAC protocols into four categories: random, slots, frames, and hybrid. These categories are generally based on the similarities and differences in the medium access methods. A recently published survey [6] classifies protocols based on various problems they intend to address. In this paper, we do not just classify protocols into different categories. We further provide a clue about why a protocol is proposed for a problem and what remaining issues lead to another solution, identifying the intrinsic development flow.

In the early stages, efficient data delivery was not the first priority. Designers traded throughput and delay for energy efficiency. However, to support multi-task and efficient delivery of bursty traffic, new protocols are being developed. In this survey, we provide an overview of WSN MAC protocols from perspectives of both energy efficiency and data delivery performance with more recently proposed work. We cover the most recent progression from energy efficiency towards efficient data delivery. Based on research issues, we divide WSN MAC protocols into four branches: asynchronous, synchronous, frame-slotted, and multichannel. Compared with prior classifications, this classification makes the major problems that each branch intends to solve clearer because each branch has its own special obstacles to overcome. It is beneficial for researchers to know the major challenges they face in the design of MAC protocols in each branch. For instance, asynchronous and synchronous are related to the mechanism of duty cycling in WSNs. To save energy, duty cycling is widely adopted in WSNs. In this technique, each node alternates between active and sleep states. Two nodes can communicate only when they are both active. In synchronous MAC protocols, neighboring nodes are synchronized to wake up at the same

Manuscript received 15 July 2011; revised 26 February 2012.

P. Huang, L. Xiao, S. Soltani, and M. W. Mutka are with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824 (e-mail: {huangpe3, lxiao, soltani1}@cse.msu.edu).

N. Xi is with the Department of Electrical and Computer Engineering, Michigan State University, 2120 Engineering Building, East Lansing, MI 48824 (e-mail: xin@egr.msu.edu).

Digital Object Identifier 10.1109/SURV.2012.040412.00105

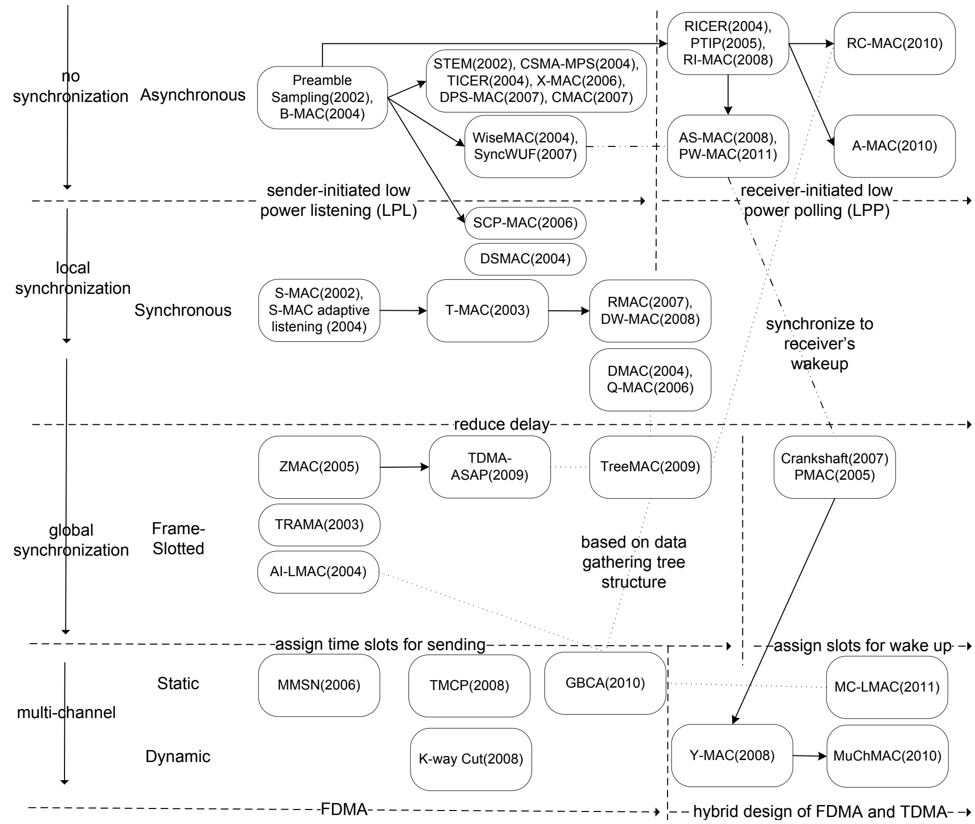


Fig. 1. Taxonomy of WSN MAC protocols.

time. Therefore, the communication is facilitated and the focus is on the delay reduction and throughput improvement. Asynchronous MAC protocols, on the other hand, focus on how to efficiently establish communication between two nodes that have different active/sleep schedules. To provide high throughput, frame-slotted mechanisms allocate time slots in a way that no two nodes within the two-hop communication neighborhood are assigned to the same slot. This addresses collision and hidden terminal problem, providing a collision-free data transmission environment. However, a major concern is that the channel utilization is low when few nodes have data to send because time slots assigned to their neighbors are wasted. Therefore, the focus of this branch is to improve channel utilization under low contention. In the final branch, multichannel is employed to further boost network capacity. Distributed channel assignment and efficient cross-channel communication are two major challenging issues in multichannel MAC protocols.

A taxonomy of WSN MAC protocols is illustrated in Fig. 1. The four horizontal branches represent our classifications. We start by introducing asynchronous MAC protocols. In the beginning, asynchronous MAC protocols use sender-initiated low power listening (LPL) to reduce cost on the receiver side. Later, receiver-initiated low power probing (LPP) is introduced to improve throughput and reduce cost on the sender side. We analyze underlying reasons behind this transition. For synchronous MAC protocols, we review the efforts that have been made to reduce delay. Synchronous MAC protocols group nodes into clusters in order to set up a common active/sleep schedule within a cluster. An inconvenience is that a node may need to maintain multiple schedules if it belongs to multiple clusters. With slightly more strict global time synchronization, frame-slotted schemes are able to address collision and hidden terminal problem by assigning a node an unique time slot within its two-hop communication neighborhood to transmit. Recently, the frame-slotted structure is also used for assigning time slots to nodes for receiving instead of transmission. We discuss benefits and problems introduced by this change. Finally, as current sensor platforms support multiple channels, multichannel MAC protocols also become a hot topic. The idea is to increase parallel transmissions by assigning nodes different channels. The future trend is to combine Time Division Multiple Access (TDMA) with Frequency Division Multiple Access (FDMA) to address cross-channel communication deficiency. Fig. 1 links MAC protocols that share similarity in utilizing some underlying structures or exhibit similar behaviors.

The remainder of this article is structured as follows. We first summarize the studies on asynchronous MAC protocols in Section II. After that, we focus on delay reduction in synchronous MAC protocols in Section III. In Section IV and Section V, we show how TDMA and FDMA can boost network capacity. In Section VI, we review how standards incorporate the various techniques. Finally, we provide conclusions and identify potential research directions in Section VII.

II. ASYNCHRONOUS MAC PROTOCOLS

In this section we cover the trend in the design of asynchronous MAC protocols, where each node chooses its active schedule autonomously. Without paying the price for synchronizing neighbors' schedules, asynchronous MAC protocols can achieve ultra-low duty cycle but have to search efficient ways to establish communication between two nodes.

In WSNs, a node spends most of the time in the low-power sleep mode and wakes up periodically to check whether there are packets for it. To indicate that there is an impending data transmission, a sender precedes its data with a preamble that is long enough to be detected by all potential receivers. This design is suitable for low traffic load applications where occasional data transmissions do not incur too much overhead and channel contention is not severe. Because the preamble transmission occupies the channel and prevents neighboring nodes from transmission, the achievable throughput is limited. To support higher throughput, asynchronous MAC protocols began to adopt receiver-initiated probing to release more room for data transmission. We will review various techniques that have been proposed to reduce the preamble length and the recent transition from sender-initiated transmission to receiver-initiated transmission.

A. Preamble Sampling

Assuming traffic is light in WSNs, the primary source of energy waste is idle listening. To reduce idle listening, duty cycling is widely adopted in WSNs. In this technique, each node alternates between active and sleep states. To let most nodes sleep as long as possible, preamble sampling is proposed to minimize the active duration of receivers. The method is to let a node periodically wake up for a short duration to sample the channel. If the channel is idle, the node goes back to sleep immediately. If channel activities are detected, the node keeps listening until the subsequent data frame is received or a timeout occurs. The method requires that the first data frame is preceded by a preamble that is at least as long as the channel sampling interval. Assuming all nodes have the same channel sampling interval, such a long preamble ensures that all potential receivers can detect the preamble and stay awake to receive the subsequent data frame. The bad consequence is that in unicast transmissions nontarget nodes are unable to find out that the packet is not destined for them until the end of the long preamble transmission. The overhearing problem thus incurs energy waste that is proportional to the node density. The advantage of preamble sampling is that the channel sampling duration is short and thus the channel sampling can be made frequent. This makes preamble sampling protocols be able to achieve satisfactory delay with ultra-low duty cycle.

The preamble sampling technique has been introduced along with the Mica wireless platform [7]. The performance of Aloha with preamble sampling and CSMA with preamble sampling have been analyzed in [8] and [9] respectively.

B. Low Power Listening (LPL)

In CSMA with preamble sampling, a transmitter performs Clear Channel Assessment (CCA) before transmitting a preamble so as to avoid collisions. A receiver also needs to perform the CCA to detect the preamble. The initial proposal in the Mica wireless platform [7] is to sample the energy on the channel and compare it to the noise floor. The channel is considered clear only if the measured channel energy is below the noise floor. Because noise has significant variance in channel energy [10], the thresholding method introduces a large number of false positives (i.e., channel is active) that lower the effective channel bandwidth because nodes cannot send, and reduce energy efficiency as nodes believe that there is an impending data transmission. Motivated by the observation that channel energy is fairly constant during packet reception, **B-MAC** (Berkeley MAC) [10] introduces an outlier detection method to improve the quality of CCA.

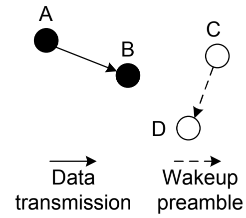


Fig. 2. Interference problem of preamble transmission.

Each node first takes signal strength samples at times when the channel is supposed to be clear (e.g., immediately after transmitting a frame). These samples are entered into a FIFO queue and the median of the queue is used as the input of an exponentially weighted moving average function for noise floor estimate. Then, when a node needs to perform the CCA, it searches for outliers in the received signal such that the channel energy is significantly below the noise floor. If an outlier is detected during channel sampling, B-MAC declares the channel is clear because a valid transmission could never have an outlier significantly below the noise floor. If no outlier is found within five samples, B-MAC declares the channel is busy. With the more accurate CCA, B-MAC names its preamble sampling as Low Power Listening (LPL) because fewer false positives lead to lower duty cycle.

C. Decouple Data Transmission and Preamble Sampling

In preamble sampling, a node turns its radio on when it has data to send. It is unaware of neighboring nodes' activities. Due to the hidden terminal problem, even with the CCA algorithm, its preamble transmission may collide with ongoing data transmissions of neighboring nodes. An example is shown in Fig. 2, the preamble transmission of node C may collide with the ongoing data transmission between A and B. **STEM** [11] thus separates data transmission channel from wake-up channel by using two radios.

STEM provides two preamble variants: STEM-T (STEM Tone) and STEM-B (STEM Beacon). STEM-T uses a bit stream as the preamble. A node can detect the preamble by comparing measured channel energy with the noise floor. Essentially, STEM-T is similar to traditional preamble sampling technique except for moving data transmission to a separate channel. In STEM-B, a series of beacon packets are used as the preamble. Each beacon contains the MAC address of both the sender and the target receiver. A node thus can determine whether it is the target receiver of an impending data transmission. Compared with traditional preamble sampling protocols, nontarget nodes can go back to sleep earlier in STEM-B. In addition, inter-beacon intervals are inserted to let the target receiver be able to reply with an acknowledgment (ACK). In this way, the sender can start data transmission earlier, resulting in a reduction in setup latency. It is possible that collisions between beacons occur. A node then cannot determine whether it is the target receiver or not. It turns its data radio on without sending an acknowledgment back. Since the sender cannot get any acknowledgment, it has to transmit the beacon stream for a sufficient amount of time (approximately equal to the channel sampling interval) before it sends the data frame on the data channel. If a node wakes up due to beacon collisions and it does not receive any data after some time, it times out and goes back to sleep. Obviously, when there is no beacon collision, STEM-B exhibits its merit of shortening preamble length, leading to lower setup latency and better energy efficiency. However, if data transmission is rare, STEM-B is less energy efficient than STEM-T because the channel sampling period must be longer than the inter-beacon interval in STEM-B while detecting a wake-up tone in STEM-T takes much less time.

D. Reduce Preamble Length by Packetization

Apparently, it is beneficial to divide a long preamble into a series of short packets that take some useful information. If the destination address is included, nontarget nodes can immediately go back to sleep when they receive a short preamble packet. This improves energy efficiency. **ENBMAC** [12] uses the method to address the aforementioned overhearing problem. If the timing information about when the data transmission will begin is also included, the target receiver can return to sleep once it decodes the timing information and wake up later to receive the data as shown in Fig. 3. Neighboring senders that fail on channel contention can go back to sleep as they know when the current transmission will end. Many protocols such as **MFP** [13], **B-MAC+** [14], **SpeckMAC** [15], **DPS-MAC (Divided Preamble Sampling-MAC)** [16], and **SyncWUF** [17] share the similarity although they utilize different information for different problems. Note that although they divide a long preamble into chunks, there is no gap between chunks. We classify them as **Continuous Preamble Sampling** protocols.

For unicast transmissions, it is desirable to terminate the preamble transmission once the target receiver is awakened. The similar idea of STEM-B has been adopted in single channel designs: a gap between two short packets is deliberately inserted to allow the target receiver to reply with an early ACK. We call it **Strobed Preamble Sampling**. This design is partially pushed by hardware advance toward packetizing radios. Instead of transmitting a raw bit stream, a packetizing radio takes the input as the payload of a packet and inserts its own preamble, header information, and CRC. When a packet is received, the radio passes the payload of the packet to the microprocessor if the packet passes the CRC check. While this radio technology reduces the burden on the microprocessor, it introduces a challenge to LPL protocols that assume a wake-up preamble of a continuous bit stream. The reason is that the TX FIFO is limited in low cost radios and a long preamble can only be mimicked by sending a packet repeatedly. A gap exists between two transmissions and the outlier detection based CCA searches for outliers to determine channel state. The gap between two transmissions may introduce false negatives (i.e., channel is not active) that increase setup latency. Therefore, a node must sample the channel long enough to accurately determine the channel state.

To be compatible with packetizing radios, it is justified to uses a series of short packets containing useful information as the wake-up preamble and utilize the gap to accommodate an early ACK as shown in Fig. 3. When there is no collision, the design addresses the overhearing problem and cuts back the preamble length. Protocols that share the similar design include **CSMA-MPS** [18], **TICER** [19], **X-MAC** [20], **MH-MAC** [21], **DPS-MAC (Dual Preamble Sampling MAC)** [22], and **CMAC** [23]. DPS-MAC [22] and MH-MAC [21] also include timing information for broadcast messages, allowing receivers to go back to sleep and wake up at the beginning of the data transmission. Strobed preamble sampling, however, incurs more energy waste in channel sampling because the channel sampling duration must be larger than the gap between two short packets. Some protocols such as SpeckMAC-D [15] and **MX-MAC** [24] repeat an actual data packet as the preamble. The channel sampling duration is even longer. As shown in Fig. 3, if a node detects channel activity when it wakes up, it must stay awake until it decodes the destination address information from the next short preamble packet. Using data packet as the short preamble packet increases the idle listening period.

In continuous preamble sampling, the channel sampling duration is short and thus the channel sampling can be made frequent. This reduces per-hop latency while keeping a low duty cycle. Strobed preamble sampling has the merit of cutting back preamble length. To combine the two features together, DPS-MAC (Dual Preamble Sampling MAC) [22] and CMAC [23] introduce a dual preamble sampling approach. Nodes periodically check the channel state by polling the channel with the basic LPL method. Because the polling may fall into the gap between two short packets, a second channel polling is initiated if the channel is found free in the first polling. To ensure that channel activity can be detected, two conditions must be met: (1) the interval between two instants of polling is larger than the gap between two short preamble packets; (2) the time duration of a short preamble packet transmission is larger than the interval between two instants of polling. Although this method reduces idle listening of strobed preamble sampling, it is sensitive to the switching time of radio. Since the time duration of a short preamble packet transmission must be larger than the interval between two instants of polling, switching time affects the size of a short preamble packet.

In addition to dual channel sampling, CMAC [23] also introduces anycast to reduce per-hop latency. Instead of letting nontarget nodes go back to sleep to reduce overhead of overhearing, CMAC allows them to reply if they can make progress (defined by geographical distance) towards the destination. To alleviate collisions among multiple candidates, a contention-based mechanism is adopted to let the best one reply first. This results in a slightly larger gap between two

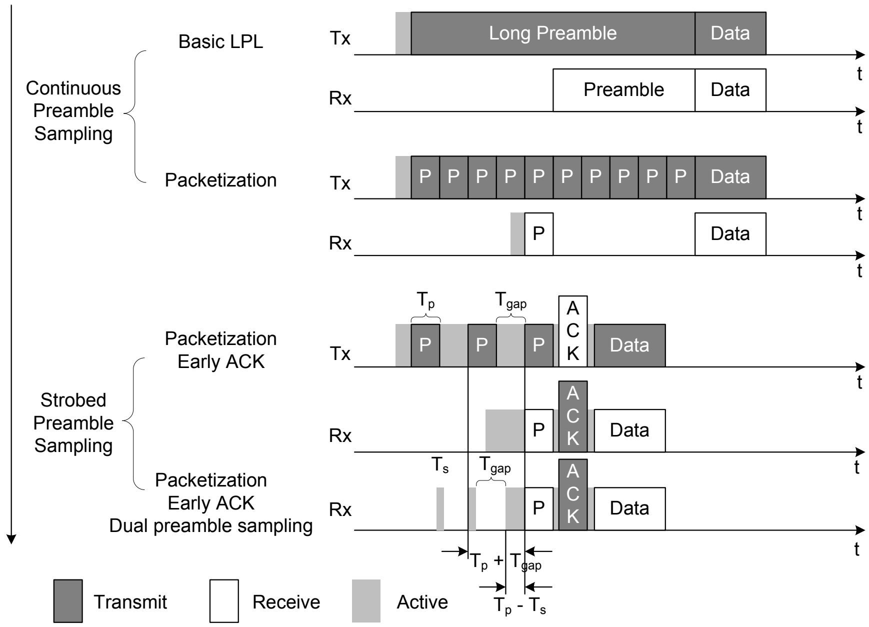


Fig. 3. Evolution of preamble sampling.

short packets. As shown in Fig. 3, if the channel is detected as busy in the second polling, a node can nap for at most T_{gap} if it does not want to miss the next short preamble packet for address decoding. The maximum idle listening period is $(T_{gap} + T_p) - (T_s + T_{gap}) = T_p - T_s$ where T_{gap} is the time period of the gap, T_p is the time period of a short preamble packet, and T_s is the channel sampling period. Since T_p must be greater than T_{gap}, the larger gap used in CMAC introduces slightly higher idle listening overhead. In addition, the anycast incurs higher overhead compared with unicast due to collisions among multiple candidates. Therefore, CMAC switches a node to unicast if the best next hop is found or no better next hop can be found after a duty cycle length.

E. Reduce Preamble Length by Schedule Learning

Although strobed preamble sampling reduces the preamble length on average, the preamble transmission still occupies the channel until the target receiver wakes up. The preamble transmission prevents neighboring nodes from transmission, leading to low channel capacity. In addition, transmitters waste energy in preamble transmission and nontarget nodes are awakened unnecessarily, causing energy deficiency at both the sender and the receiver.

For an infrastructure network, **WiseMAC** [25] lets access points stay awake to learn sensor nodes' wake-up schedules. Access points are not power constrained and each access point serves a number of sensor nodes. If the access point does not know the channel sampling schedule of a sensor node, it sends a long preamble. When a node receives a packet, it include its channel sampling schedule in the ACK. With the up-to-date channel sampling schedules of sensor nodes, the access point only needs to start preamble transmission slightly before the target receiver wakes up. The duration of the preamble is computed to guarantee that the senor node will not miss the preamble at a worst case clock drift offset. The shortened preamble alleviates the overhearing problem and increases the channel capacity. Some protocols utilize a similar idea with some improvements. For example, in **SyncWUF** [17], if the receiver's schedule is known and the computed preamble length is short enough, a simple wake-up tone is used same as in WiseMAC. However, if the receiver's schedule is unknown or out-of-state or it is a broadcast message, a long preamble is divided into chunks that contain timing information to reduce unnecessary waiting time. In **CSMA-MPS** [18], strobed preamble sampling is combined with schedule learning. Because CSMA-MPS allows for early ACK, a node may not need to finish a preamble of the worst case length. Although schedule learning squeezes more room for data transmission, the cost of maintaining up-to-date schedules of neighbors pushes some studies to shift the responsibility of establishing communication from the sender side to the receiver side.

F. Improve Throughput with Receiver Initiated Transmission

Fig. 4 illustrates the disadvantage of preamble transmission. When there are contending flows (e.g., A sends to B and C sends to D), the preamble transmission prevents neighboring nodes from delivering their packets. When nodes C and D wake up, they go back to sleep immediately after they find out that they are not the target receiver of current transmission. This leads to low channel capacity and high delay when there are contending flows.

To avoid occupying the channel before a pair of nodes are ready to exchange data, a number of protocols muffle the sender and rely on the target receiver's beacon message to initiate data transmission. As shown in Fig. 5, a node broadcasts a beacon to announce that it is ready for receiving when it wakes up. Because a node stays awake if it has data to send, it can hear the target receiver's beacon. Upon receiving the beacon from the target receiver, the sender starts

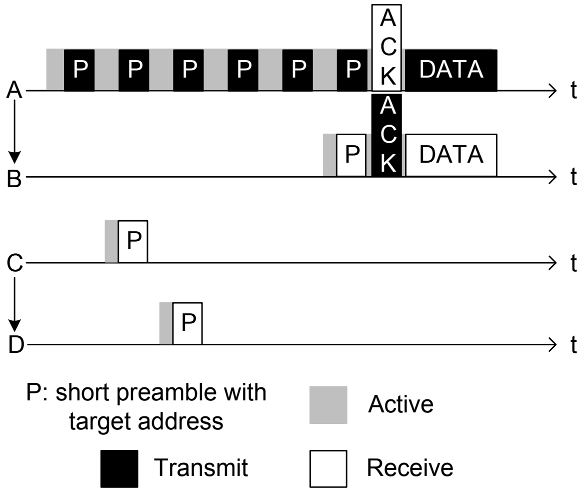


Fig. 4. In strobed preamble sampling, short preambles occupy the wireless medium until the data packet is delivered.

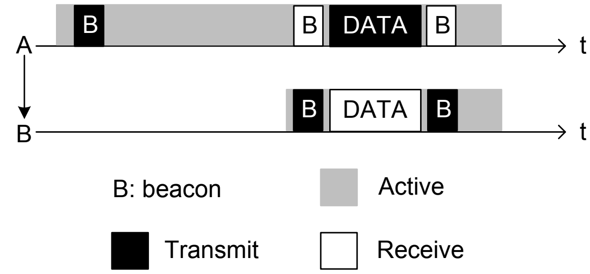


Fig. 5. In low power probing, a sender does not occupy the wireless medium until the receiver is ready to receive.

to transmit data packets to the receiver. Multiple nodes may have data for the receiver. An efficient collision resolving mechanism is thus needed. If a node does not receive anything for a certain amount of time after it broadcasts a beacon, it goes back to sleep. The advantage of the receiver-initiated Low Power Probing (LPP) design is that the channel is vacant for use before the target receiver is ready to receive. The idea of shifting communication initiation from the sender side to the receiver side is early presented in **RICER** [19] and employed in **PTIP** [26] for infrastructure networks, **Koala** [27], **AS-MAC** [28], **RI-MAC** [29], and **A-MAC** [30] for general sensor networks. When a node broadcasts a beacon, more than one of its neighbors may respond with a data packet. RI-MAC resolves collisions by reusing the beacon messages and allows back-to-back data transmission.

Considering that the traffic is light most of the time in a sensor network, RI-MAC lets a node broadcast a base beacon message with no backoff field when it wakes up. As a result, neighboring nodes can start data transmission to the node immediately. This is optimized for light traffic load scenarios where there is only one sender most of the time. Under heavy traffic load, collisions may occur and the receiver broadcasts another beacon message that includes a backoff window field. Neighboring nodes will back off randomly according to the random backoff window size indicated in the new beacon message. Whenever a collision is detected after a beacon, the receiver increases the backoff window size in a Binary Exponential Backoff (BEB) way until the maximum window size is reached. Because neighboring nodes' contentions are synchronized by each beacon, collision is still the major factor that bounds the throughput. In addition, overhearing a transmission may incorrectly trigger the recovery mechanism, leading to more redundant beacons and then more collisions. To reduce collisions, RC-MAC [31] utilizes the tree structure formed during data gathering to schedule the transmission of multiple nodes. Another improvement upon RI-MAC is to devise a prediction mechanism for estimating neighbors' wake-up time as introduced in PW-MAC [32]. Recently, **A-MAC** [30] shows that by enabling auto-acknowledgement to the beacon message, a node is able to quickly determine whether there is a packet for it.

G. Reduce Collision in Receiver Initiated Transmission

Since senders are required to wake up before receivers, **RC-MAC** [31] coordinates multiple senders' transmissions by piggybacking a scheduling message, which specifies the node that can transmit upon receiving the message, to an ACK. The method relies on that a data gathering tree exists or is formed during route discovery. On the tree structure, a parent receives packets from multiple children and forwards packets to its own parent. A parent acknowledges its children with an ACK message that indicates which child is the next sender. Therefore, the scheduled child can transmit immediately to reduce latency and sibling nodes will back off to avoid collisions. By learning the bandwidth demands of children, a parent adjusts the channel access opportunities in accordance with their demands so that fairness is ensured. RC-MAC also explicitly interrupts the scheduling of a parent-children unit so as to ensure that no unit can occupy the channel exclusively. The scheduling round of a unit is dynamically adjusted according to the parent's remaining buffer size, which balances the channel access opportunities among units.

The local scheduling method reduces collisions and increases throughput in a basic parent-children unit. The contention among units, however, significantly affects the performance. RC-MAC relies on explicit back-off to reduce interruption of scheduling caused by contention among units. Coexistence of units in a dense network, however, still remains a key issue. Multichannel design is a promising solution for addressing interference among units.

H. Estimate the Wake-up Time of Receiver

Although LPP increases channel utilization, only the receiver benefits from the receiver-initiated design. The sender has to stay awake until the data packets are delivered. To reduce energy consumption of senders, **PW-MAC** [32] introduces a method to predict the target receiver's wake-up time so that a sender only needs to wake up slightly before the target receiver. The motivation is same as that of WiseMAC [25], but PW-MAC allows nodes to independently generate pseudo-random schedules, which avoids persistent collisions caused by two similar wake-up schedules. The collision can also be avoided by distributing nodes' schedules as introduced in **AS-MAC** [28], but PW-MAC allows to use variable wake-up intervals.

Since a pseudo-random function generates the same sequence of pseudo-random numbers for the same parameters, PW-MAC utilizes the beacon message used in RI-MAC to relay these parameters. By learning neighbors' pseudo-random function parameters, a node can easily calculate the future wake-up time of any neighboring node. Clock drift, hardware and operating system latency may cause prediction errors. A sender in PW-MAC thus advances its wake-up by a time period that is adjusted according to the clock drift rate. In addition, if a sender finds that the prediction error (defined as the difference between the estimated wake-up time of the target receiver and the actual wake-up time of the receiver) is greater than a threshold, the sender requests an update of the prediction state. This ensures that the prediction error is within the sender wake-up advance time so that a sender will not miss the receiver's beacon.

Currently, the pseudo-random function parameters are fixed for each node. If these parameters need to be adjusted according to some optimization mechanisms, maintaining information consistence among neighbors would be a problem. In addition, constant calculation of neighbors' schedules introduces unnecessary computing overhead, which causes additional energy consumption and introduces delay for other system operations. A periodic wake-up schedule in AS-MAC [28] is easier to follow but nodes' schedules must be distributed to avoid constant collisions.

I. Summary

In this section, we have surveyed various ways to establish communication between two nodes in asynchronous MAC protocols. The research starts with using long preamble to wake nodes up. The preamble sampling method is first improved with a better outlier detection based CCA algorithm and has been suggested to decouple data transmission from preamble sampling. Soon four approaches have been proposed to reduce cost at receivers and senders as shown in Fig. 1. The first solution is to include useful information in preamble. This addresses the overhearing problem but still inherits the drawback of consuming channel bandwidth for preamble transmission. The other three methods include schedule learning, synchronized polling, and receiver-initiated LPP. All of the three methods reduce the amount of time a pair of nodes occupy the wireless medium before they actually exchange data. In the following section we will discuss the synchronized polling, which shares similarity with schedule learning in that nodes are synchronized to a common schedule. However, contention in synchronized polling is more severe than that in schedule learning because all nodes that are synchronized to a common schedule share the wireless medium in synchronized polling whereas only several senders are synchronized to the target receiver's schedule in schedule learning. A good combination in asynchronous MAC protocols would be receiver-initiated LPP with auto-ACK to the beacon and distributed wake-up schedule, but how to ensure that the beacon will not be missed while minimizing the idle listening is platform dependent. Once nodes can reach each other efficiently, more efforts can be put on throughput improvement and delay reduction.

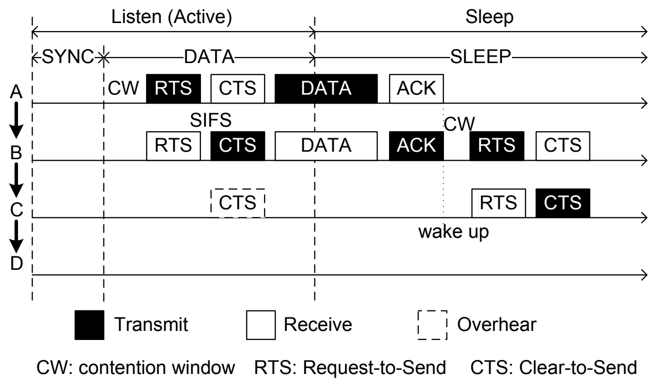


Fig. 6. The adaptive listening in S-MAC.

III. Synchronous MAC protocols

In this section we summarize the recent developments in synchronous MAC protocols. Synchronizing active time of neighboring nodes is a natural solution to establish communication between two nodes. However, it brings the cost of additional synchronization overhead.

In synchronous MAC protocols, a node listens to the channel for a certain amount of time. If it does not hear any schedule from other nodes, it determines its next wake-up time and broadcasts its schedule. This makes the node become a synchronizer. If a node receives a schedule from a neighbor before choosing its own schedule, it follows the received schedule, which makes it a follower. Generally, nodes form clusters of several hops. All nodes in a cluster are synchronized to a synchronizer that is one or a few hops away. If a node receives a different schedule after it sets its own schedule (i.e., either has broadcasted its own schedule or followed a previously received schedule), it adopts both so that it can be a bridge between two clusters. In other words, the node wakes up at times of both its neighboring cluster and its own cluster.

As schedule learning is introduced to asynchronous MAC protocols, the boundary between synchronous MAC and asynchronous MAC gradually becomes fuzzy. A main difference is that only senders wake up at the target receiver's polling/probing time in prediction integrated asynchronous MAC protocols while a cluster of nodes wake up at the same time in synchronous MAC protocols. The common active period is not reserved for a single receiver. It is shared by all nodes in the cluster and they must contend for channel access. In other words, synchronous MAC protocols require local time synchronization while a node in asynchronous MAC protocols chooses its schedule independently without awareness of its neighbors' schedules.

Synchronous MAC protocols do not face the problem of establishing communication between nodes as asynchronous MAC protocols. Therefore, most designs focus on improving throughput and reducing delay. They are more appropriate for applications of periodic traffic where the wake-up schedule is easy to determine.

A. Adaptive Listening

A classical synchronous MAC protocol is **S-MAC** [33], where nodes are organized into clusters by local time synchronization. Each cluster has an independent schedule composed of three periods: SYNC, DATA, and SLEEP. All nodes of the same cluster wake up at the beginning of the SYNC period to synchronize clocks with each other. Then, nodes with packets to send contend for exchange of Request-to-Send (RTS) and Clear-to-Send (CTS) frames in the DATA period. Nodes that are not involved in communication, return to sleep at the start of the SLEEP period; other nodes return to sleep after they finish transmission of data packets and acknowledgement (ACK) frames. It is noticeable that in each cycle a packet can only be forwarded by one hop. Although adaptive listening [34] is later introduced into S-MAC to overcome this deficiency, the improvement is limited. A packet can only be forwarded by at most 2 hops per cycle.

S-MAC with adaptive listening [34] is illustrated in Fig. 6. Nodes can only hear their immediate neighbors (e.g., node B can only hear node A and node C). Because node C can overhear the CTS sent by node B, it goes back to sleep at the beginning of the SLEEP period but wakes up at the end of the current transmission. Node B can therefore immediately forward the data packet to node C instead of waiting for the next cycle. The cost is that not only the target receiver but all nodes that overhear the RTS and the CTS will wake up. The benefit is that S-MAC with adaptive listening can relay a packet by 2 hops per cycle. However, the delivery cannot go beyond this because the next hop (i.e., node D) will not wake up since it cannot overhear the CTS from its two-hop neighbor (i.e., node B) and does not know when the current transmission will end. The long end-to-end delay makes S-MAC inappropriate for applications that require strict timing constraints.

B. Future Request-to-Send

Instead of fixing the length of the active period, **T-MAC** [35] makes a node stay awake until no activation event has occurred for a certain amount of time. This design aims to achieve optimal active periods under various traffic loads. However, overhearing is introduced because a node has to stay awake while it is not involved in data transmission. T-MAC justifies overhearing with improvement on the throughput.

Another contribution of T-MAC is the idea of Future Request-to-Send (FRTS) packet, which allows a node to notify its target receiver that it cannot access the medium at the current time. The mechanism has the potential to extend the data delivery to up to 3 hops per cycle. As shown in Fig. 7, when node C loses contention and overhears a CTS packet, it sends a FRTS packet to its target receiver D. The FRTS packet contains the length of the current data transmission from node A to B, hence the target receiver D can learn its wake-up time. However, nodes that are further downstream cannot overhear the FRTS and thus will not wake up. The mechanism thus cannot be performed iteratively and a packet can only be forwarded by at most 3 hops per cycle. During the time it postpones its data transmission for the FRTS, node A must transmit a Data-Send (DS) packet of the same size of FRTS to prevent any neighboring node from taking the channel. The collision of FRTS and DS at node B is not a problem because they are useless for node B. The potential collision of multiple FRTS packets, however, is not discussed.

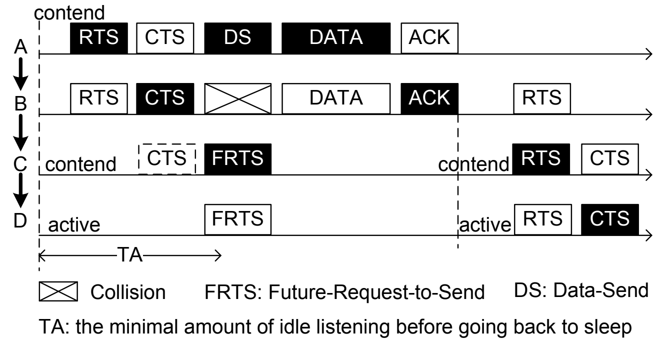


Fig. 7. The FRTS mechanism in T-MAC.

C. Shifting Data Transmission to the Sleep Period

Inspired by the FRTS mechanism, **RMAC** [36] presents a novel approach to reduce latency in multi-hop forwarding. Same as S-MAC, time is divided into repeated cycles and each is further divided into three periods: SYNC, DATA, and SLEEP. Instead of exchanging data during the DATA period, a control frame called Pioneer Frame (PION) is forwarded by multiple hops. The control frame informs nodes on a routing path of when to wake up in the SLEEP period. A PION not only serves as a RTS frame to request communication, but also confirms a request like a CTS frame. RMAC and DW-MAC [37] are similar as shown in Fig. 8. The differences are that the Scheduling Frame (SCH) in DW-MAC is called PION in RMAC, and the time T_1^s is 0 in RMAC, which implies that the first hop starts data transmission when the SLEEP period starts. The drawback of RMAC is that two hidden terminals may always cause collisions of data transmission although they succeed in scheduling through PIONs. For example, node C in Fig. 8 may initiate a PION later than node A. Although they can succeed in PION transmission, they start data transmission at the same time when the SLEEP period starts as they are both the first hop. They will collide with each other and downstream nodes will wake up unnecessarily if the expected data packets cannot arrive due to collisions at previous hops.

DW-MAC [37] thus introduces a one-to-one mapping function to this design to ensure collision-free data transmission in the SLEEP period. In the example illustrated in Fig. 8, node B calculates its wake up time T_1^S as

$$\frac{T_1^D}{T_{DATA}} = \frac{T_1^S}{T_{SLEEP}}$$
$$\Rightarrow T_1^S = T_1^D \cdot \frac{T_{SLEEP}}{T_{DATA}}$$

In addition,

$$\frac{T_3^D}{T_{DATA}} = \frac{T_3^S}{T_{SLEEP}} \tag{1}$$

Equation 1 implies that if node A can receive a confirmation SCH from node B, the time length of T_3^S is assured to be collision-free because node B can receive a SCH correctly during T_3^D. They are one-to-one scaled based on the ratio between T_{SLEEP} and T_{DATA}. Therefore, successfully reserved data transmission will never collide at a target receiver in the SLEEP period. If the SCH sent to node B is collided, node B will not respond a confirmation SCH and thus node

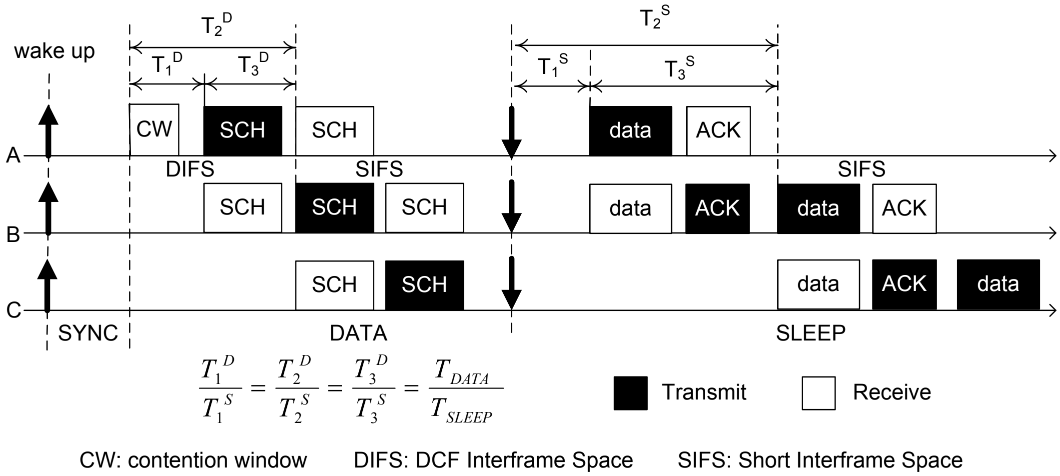


Fig. 8. Multi-hop forwarding of a unicast packet in DW-MAC.

A and B will not reserve a time for data transmission during T_3^S. However, if the confirmation SCH is collided at node A, node A will not wake up to send during T_3^S but node B will wake up to receive. The difficulty is that the sender cannot distinguish the collision of a request SCH and the collision of a confirmation SCH. It can only conservatively regard that the reservation failed. Consequently, all downstream nodes will wake up unnecessarily to receive the expected data packet that will not arrive due to the false alarm.

The cost of waking up unnecessarily may be tolerable, but any failure of a sequential SCH reservation is a waste of time and it significantly limits the number of packets that can be delivered in a cycle. A sequence of time slots are reserved for a data packet but no data will be transmitted, whereas many other data packets have no time to reserve time slots for transmission because the DATA period is limited. In a complex multiple flow scenario, the performance of throughput, delay, and power consumption is significantly compromised due to the incorrect channel reservation.

D. Staggered Schedule

For specific applications, some underlying structures can be utilized. **DMAC** [38] targets at data gathering applications where data are delivered from multiple sources to a sink. Based on the naturally formed data gathering tree, the active/sleep schedules of nodes are staggered so that packets can flow continuously toward the sink.

As shown in Fig. 9, each node skews its wake-up time $d\mu$ ahead of the sink's schedule in accordance with its depth d on the data gathering tree. The variable μ represents the length of the time that is needed for one packet transmission and reception. The ordered offsets of schedules ensure sequential transmission and lead to low delay. However, because nodes of the same depth have the same offset, they contend for sending to their receivers. When one node wins channel access, its neighboring nodes of the same level lose their chance of transmission. In order to increase the number of active slots, a data prediction mechanism and a More-to-Send (MTS) notification are introduced. In Fig. 9, if node B wins channel access opportunity to send to C, node C needs to add another RECV slot 3μ later than the current RECV slot to check whether node A has data to send to it. The length of 3μ is selected to ensure that the previous packet has be forwarded 3 hops away. In addition, node D also needs to send a MTS packet to node E to make node E wake up periodically. This ensures that when node D wins channel access, node E is ready to receive. The overhead is increased along with the traffic load because nodes on routing paths have to wake up repeatedly for contention of sending or receiving.

Similarly, **Q-MAC** [39] also defines staggered schedule but the active periods are shifted in a way that facilitates downlink traffic, carrying queries from the sink to sources. The MAC is designed for query based sensor networks where the sink initiates data gathering by sending a query to the source. For uplink traffic, Q-MAC has two options. The simplest way is to let each node remain active from the instant of query reception to the time the queried data are forwarded. If the sink knows the route length in advance, an uplink staggered schedule can be constructed. Although Q-MAC reduces latency, it incurs long idle listening if the route length is unknown. Low-power wireless communication is vulnerable to interference and thus routes in WSNs are usually not fixed. Therefore, the energy efficiency of Q-MAC is questionable when the route length is hard to determine.

E. Adaptive Duty Cycle

The basic preamble sampling method requires a long preamble before data, following the design of synchronous MAC protocols, SCP-MAC [40] synchronizes neighbors' channel polling so that only a short preamble is required to wake the receiver up. Because a cluster of nodes share the same schedule, contention is more severe compared with schedule learning in asynchronous MAC protocols. However, SCP-MAC incurs less schedule maintenance overhead as nodes do not need to maintain multiple schedules.

To reduce multi-hop latency, SCP-MAC develops an adaptive channel polling mechanism to add additional polling slots along the path. As shown in Fig. 10, when node B receives a packet at the first regular polling slot, it adds n high frequency polling slots to receive additional data packets from node A. If

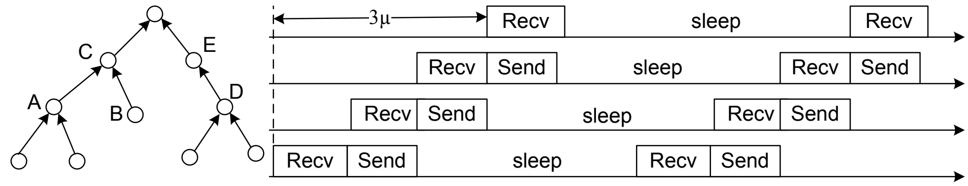


Fig. 9. DMAC in a data gathering tree.

any of the additional polling slots was successfully used, node B extends the adaptive polling with another n slots; otherwise, it returns to the regular polling frequency. In order to quickly shift nodes on a routing path to adaptive polling, node A intentionally pauses its transmission at the second regular polling slot, allowing node B to send to C, but nodes A and B have to contend for sending in the following adaptive slots. The procedure repeats so that a new node can enter the high frequency polling at each regular polling slot. For adaptive polling slots, n is set to 3 considering that each node contends with its previous and next hop. However, it is possible that a node loses contention for all of the 3 slots and returns to the regular low frequency polling, which will interrupt the continuous forwarding. In addition, inter-flow contentions may also degrade the adaptive polling performance. **DSMAC** [41] is another WSN MAC protocol that changes duty cycle dynamically. A node doubles its duty cycle when it detects the increase of its neighbors' bandwidth demands and the new duty cycle is broadcasted along with the SYNC packet. These methods reduce delay when the traffic load increases.

F. Summary

In this section, we reviewed the efforts that have been made to reduce multi-hop forwarding delay in synchronous MAC protocols. With the help of data gathering tree structure, schedules can be staggered to create a continuous flow to the sink. However, nodes wake up unnecessarily if their upstream nodes lose contention in channel access. Note that multiple flows converge at the sink and thus the contention becomes more and more severe near the sink. Without the traffic pattern constraint, most protocols try to notify as many nodes as possible on active routes of incoming data as soon as possible. One trend is to shift data transmission from the DATA period to the SLEEP period and use the DATA period for notification and scheduling. While this paradigm minimizes delivery latency for individual packets, some packets may experience long delay as they cannot be scheduled for transmission within limited DATA period. Therefore, more efficient scheduling methods are needed. Without a subtle scheduling design, the delay can be reduced by dynamically increasing the duty cycles of nodes on the active routing paths. The performance improvement, however, comes with the cost of increased energy consumption.

IV. FRAME-SLOTTED MAC PROTOCOLS

Frame-slotted schemes derive from Time Division Multiple Access (TDMA). The advantage of TDMA is that the throughput is high with maximized channel utilization under high contention. TDMA can be used in synchronous MAC protocols in each cluster. However, if the active periods of two clusters overlap, the collision-free data transmission is no longer guaranteed. Therefore, although local time synchronization is more scalable, TDMA is usually defined with slightly more strict global time synchronization. Although TDMA requires global time synchronization, the adoption of frame-slotted schemes will not introduce additional overhead in applications that require time synchronization to provide spatial-temporal correlation between reports generated by multiple sensors. For example, in the SENSEI project [2] developed by European FP7-ICT, heterogeneous wireless sensor and actuator networks have to be integrated into a common framework. In order to obtain information from various sensors located in the physical world, the spatial and temporal results from the sensors should be synchronized. Frame-slotted MAC protocols would be a promising candidate for these kinds of applications.

Frame-slotted MAC protocols are also popular in small scale networks. A special case of WSN is the wireless body area network (WBAN). A WBAN consists of a number of sensors that are either connected with a person's body or are small enough to be implanted. Sensors transmit data at a relatively wide range of data rates from 1 Kbps to 1Mbps for body temperature, electrocardiography (ECG, heart), electromyography (EMG, muscular contractions), electroencephalography (EEG, brain), movement, and etc. The WBANs have relatively constant network structures and fixed sensor functions. Therefore, many recently proposed MAC protocols are TDMA-based [42] [43] [44] [45]. In these networks the synchronization procedure can be simplified due to the hierarchical structure. Master nodes that have more power act as coordinators and this removes the need of idle listening for other nodes [46].

A. Increase Channel Utilization by Slot Stealing

A major drawback of TDMA is its low channel utilization when only few nodes have data to send because a node can transmit only in its assigned time slots. A sensor network, however, generates no or few data packets most of the time. To improve channel utilization of TDMA under low contention, **Z-MAC** [47] incorporates Carrier Sense Multiple Access (CSMA) into TDMA.

When a WSN is deployed, Z-MAC applies DRAND [48] to do time slot assignment. DRAND ensures that no two nodes within the two-hop communication neighborhood are assigned to the same slot. To further increase the channel utilization, a time frame rule is designed to adapt the local frame size to a node's local neighborhood size. To combine CSMA with TDMA, a node is allowed to contend for sending if a slot is not used by the owner. Consequently, Z-MAC performs as CSMA under low contention and possesses high channel utilization

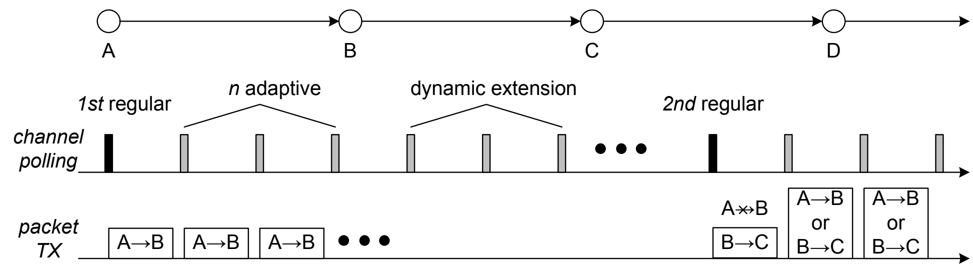


Fig. 10. Adaptive channel polling in SCP-MAC.

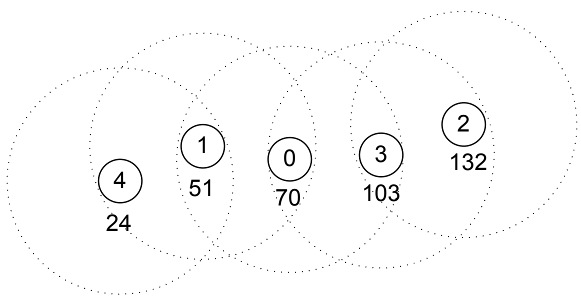


Fig. 11. Low channel utilization problem of TRAMA. The number under each node is its priority for a particular time slot t.

as TDMA under high contention. However, the hybrid design faces challenges in a dense network. When some nodes have data to send, they have to contend for slots that are assigned to their neighbors who have no traffic and these contentions are synchronized in each slot. In every slot, a sender has to wait for a certain amount of time to ensure that the slot is abandoned by the owner. Each receiver also has to stay awake to check whether it is the target receiver. As a result, the slot stealing method introduces nontrivial additional energy consumption.

Another work that studies the slot stealing problem is **TDMA-ASAP** [49]. It utilizes the data gathering tree structure to reduce collisions in slot stealing. The advantage of TDMA-ASAP is that when a node abandons a slot, only nodes that have data for the same parent can steal the slot and thus other nodes do not need to wake up to check whether they have data to receive. However, the overhead for detecting whether a slot is abandoned still exists.

B. Increase Channel Utilization by Adaptive Assignment

TRAMA [50] increases channel utilization of TDMA in another way. It switches between random access period and scheduled access period. In the random access period, nodes that have data to send will claim slots for use. The protocol avoids assigning time slots to nodes of no data to send and thus increases the channel utilization from a new perspective.

To determine the ownership of a slot, it adopts the neighborhood-aware contention resolution (NCR) algorithm of NAMA [51], which defines a node u's priority at time slot t as a hash of the concatenation of u and t:

$$priority(u, t) = hash(u \oplus t).$$

For any particular time slot, the node with the highest priority within the two-hop communication neighborhood wins the slot. Although the time slot assignment incurs no communication overhead, the spatial reuse of time slots is low because nodes' priorities may be sequential. As shown in Fig. 11, node 2 is the node with the highest priority among node 0's two-hop neighborhood for a particular time slot, and node 3 is the winner of the slot in node 1's view. Consequently, only node 2 can occupy the slot, leading to low spatial reuse of slots.

TRAMA also devises an adaptive election algorithm to reuse slots that are discarded by the owners. When a winner of a slot does not have data to send, it scans its one-hop neighbor set. If a one-hop neighbor has the highest priority among its own two-hop neighbors, the one-hop neighbor is added in the Possible Transmitter Set (PTS). Note that the node does not know the entire two-hop neighborhood of the one-hop neighbor and thus collision-free data transmission cannot be guaranteed. Finally, the node of the highest priority in the PTS is authorized to send. The procedure introduces additional computing and control overhead, which sacrifices energy efficiency and introduces system delay. In addition, packets experience long delay as downstream nodes do not know that they will have data to send and thus will not reserve time slots for data transmission.

AI-LMAC [52] adaptively assigns time slots to nodes when the sink initiates a query for data. AI-LMAC assumes that a parent-child relationship exists between all the nodes in the network and the root is the highest parent in the hierarchy. Since the parent knows the proportion of data that will be contributed by each of its immediate children, it sends a message to each of its children to advise the ideal number of slots that they should take up. A node then searches for free slots within its two-hop communication neighborhood and takes up any available slot to meet its bandwidth demand. However, if the data rates of sources are not fixed, the sink cannot accurately estimate the number of slots that each of its children should take up. In addition, there is no mechanism to ensure that a node can grab enough slots to meet the desired bandwidth demand.

C. Maximize Throughput at the Sink

The time slot assignment of most TDMA protocols is based on graph coloring, which aims at maximizing spatial reuse of slots. For data gathering applications, multiple flows converge at the sink. Fair assignment of slots among nodes cannot result in throughput maximization at the sink. Therefore, **TreeMAC** [53] devises a time slot assignment algorithm that is well tuned for throughput maximization at the sink by utilizing the data gathering tree structure.

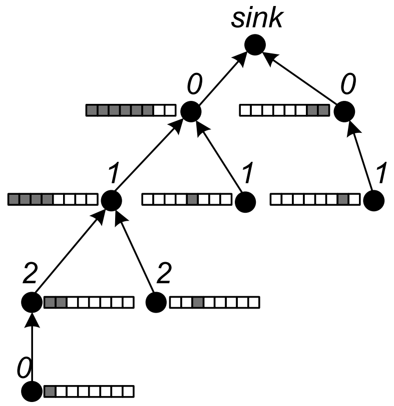


Fig. 12. In TreeMAC, nodes of the same depth on the tree get the same transmission slot in different frames. The number above each node denotes the node's transmission slot, and the shaded squares illustrate the frames assigned to the node.

The basic idea of TreeMac is to eliminate horizontal two-hop interference by frame assignment and vertical two-hop interference by slot assignment. Time is divided into cycles and each cycle is divided into N frames of 3 slots. As shown in Fig. 12, nodes of the same depth on the tree are assigned different frames for transmission. In a frame, the transmission slot is calculated as $(\ell_u - 1) \bmod 3$ where ℓ_u is the depth of the node. Using three slots, a node can avoid contention with its previous and next hop. Moreover, the number of frames assigned to a node is proportional to its traffic load. Therefore, fairness is ensured in terms of flow instead of individual nodes. Since a node may not have data to send, some slots are wasted. In addition, a node needs time to join the tree as it adopts CSMA and has a lower channel access priority than those already on the tree. The protocol thus prefers a relatively stable tree structure.

D. Reduce Duty Cycle by Switching Sending Slots to Receiving Slots

TDMA schemes specify time slots that a node can transmit. If a node receives data from multiple nodes, it has to wake up multiple times no matter whether there is a packet for it. Some protocols thus use the frame-slotted structure for specifying time slots in which a node should wake up for listening. The benefit is that only nodes that have data for the owner of the slot need to wake up. Since time slots are assigned to receivers, collision-free data transmission is no longer guaranteed. This type of MAC protocols adopt frame-slotted structure of TDMA but are functionally more similar to prediction integrated asynchronous MAC protocols (e.g., WiseMAC [25], AS-MAC [28], and PW-MAC [32]).

In **Crankshaft** [54], time is divided into frames and each frame is further divided into slots. Each node listens for one unicast slot in every frame. A key feature of Crankshaft is that several broadcast slots are added to the end of unicast slots. Nodes can contend for broadcast in the broadcast slots. Crankshaft assigns unicast slots to nodes based on node ID modulo frame size. The simple assignment saves effort for learning schedules of neighbors, but two nodes may own the same unicast slot. To make the two nodes be able to communicate, nodes are allowed to send in their assigned unicast slots. Therefore, the receiver of a slot also contends with nodes that intend to send data to it. To reduce contention, a node that loses contention in current frame retry transmission with a probability of 70% in the next frame.

The time slot assignment can be more elaborate as in **PMAC** [55], which takes traffic load into consideration. In PMAC, a node is encouraged to increase its sleep time exponentially until the upper bound is reached if it does not have data to send. The sleep/active schedule of a node is represented by a pattern consists of n-sleep/1-awake slots. Nodes' patterns are exchanged in the end of each frame. The period is set long enough such that all nodes can broadcast their patterns. Based on received neighbors' patterns, a node can determine whether the target receiver will wake up at a particular time slot or not. The adaptivity to traffic load, however, comes with the overhead of exchanging patterns every time frame. In addition, contention is severe as multiple receivers exist in a time slot. A node in PMAC reverts to a pattern of repeated '1' once it has data to send. This design enables a node to quickly respond to a new flow, but it also increases energy consumption unnecessarily if only several packets need to be delivered. The node has to wake up at every slot in the next frame as its pattern is reset to repeated '1'.

E. Summary

In this section, we have introduced two hybrid designs of CSMA and TDMA for improving channel utilization: slot stealing and adaptive time slot assignment. Slot stealing introduces additional overhead for detecting whether a slot is abandoned. In addition, each node must wake up in every slot to check whether there is a packet for it. This design sacrifices energy efficiency and deviates away from the goal of low power operation in WSNs. Adaptive slot assignment incurs less overhead but also yields lower spatial reuse of slots. A better distributed slot assignment algorithm is highly demanded. The target of time slot assignment can be quite diverse as we have seen. Instead of maximizing the channel utilization, we can try to maximize the throughput at the sink. In addition, time slots can be assigned for receiving instead of sending. The benefit is that if a node has no data for the owner of the slot, it does not need to wake up and hence saves energy. However, the challenges in this type of MAC protocols are same as prediction integrated asynchronous MAC protocols, which include collision resolving and deterministic delay guarantee.

V. MULTICHANNEL MAC PROTOCOLS

Recently parallel data transmission also attracts intense attention as many WSN platforms emerged with multichannel support. The radio bandwidth in WSNs is limited and thus it is desirable to devise multichannel MAC protocols to handle bursty traffic or provide multi-task support. The next generation of networks highlighted by the SENSEI [2] project focus on efficient utilization of spectrum and integrating various networks into an advanced global network. Multichannel WSN MAC protocols are helpful in this field.

In multichannel MAC protocol designs, two questions need to be answered: how to allocate channels and how to achieve cross-channel communication. The solution must also consider energy efficiency and cost.

A. Address Cross-channel Communication

MMSN [56] points out that multichannel MAC protocols designed for general wireless networks are not suitable for WSNs because of two reasons. First, they use some more powerful radios such as radios that have multichannel sensing ability. Second, the control overhead of channel negotiation is acceptable considering that the data packet size is much larger. As an example, MMAC [57] assumes that time is synchronized and divided into fixed-length beacon intervals. Each beacon interval is further divided into an ATIM (Ad Hoc Traffic Indication Messages) window and a communication window. Nodes negotiate which channel to use for data transmission on the same default channel in the ATIM window, and then switch to the reserved channels for communication in the communication window. Later, TMMAC [58] introduces dynamic ATIM window adjustment rules to increase the flexibility of the design. Because data packet size is usually small in WSNs, at most 128 bytes for CC2420 radio [59], pairwise channel negotiation for each packet introduces high control overhead. Authors of MMSN thus proposed to assign channels to nodes in a static way. The idea is to let a node learn its neighbors' channel choices and then select a channel that is not chosen by any of its neighbors within the two-hop communication neighborhood. The idea is similar to the time slot assignment for receivers, thereby the channel assignment cannot guarantee collision-free data transmission.

To allow two nodes of different channel choices to communicate, MMSN introduces toggle snooping and toggle transmission. If a node has data to send, it must listen on its own channel (f_{self}) for potential data reception and listen on its target receiver's channel (f_{dest}) for clear channel assessment. MMSN thus lets the node snoop on the two channels alternately. If the node does not sense any signal on both channels, the node transmits a preamble alternately on the two channels. The toggle transmission not only prevents other nodes from sending data to the transmitter but also informs any node that intends to transmit on f_{dest} to back off. The toggle snooping period (T_{TS}) is twice as long as the toggle transmission period (T_{TT}). The setting ensures that when a node sends out a preamble using toggle transmission, any other node that is toggle snooping can detect the transmission within a maximum delay of T_{TS} if they have a shared channel. Considering that the cost of channel switching is nontrivial [60], nodes that use the toggle snooping/transmission frequently may exhaust their energy earlier than other nodes.

B. Channel Assignment based on Metric Optimization

Le et al. [60] claim that frequency synthesizer needs time to stabilize and thus communication in the same channel incurs less overhead compared with inter-channel communication. Therefore, they propose to group nodes that communicate frequently into the same channel and separate nodes that do not communicate much into different channels. The idea can be modeled as finding a minimum K**-way cut** that minimizes inter-channel communication in the graph given by the network topology. To reduce the complexity of solving the problem, a heuristic method is introduced in [60]. The idea is to move some nodes to another channel when a channel becomes overloaded.

The first step is to define a metric that measures the effect of a crowded channel. All channels are organized as a ladder and all nodes start at the lowest channel. Each node periodically broadcasts the total number of times it successfully and unsuccessfully acquires the channel, s and f respectively. Node i estimates the probability that any of its neighbors can successfully access the channel as $\alpha_i = (\sum_j s_j)/\sum_j(s_j+f_j)$. If the value is less than a threshold, node i considers a switch from its current channel to the next higher channel with a probability that depends on the channel quality difference and how closely a node resembles a sink. A node that acts like a sink has a low-cost outgoing link because it sends less traffic and this makes it more appropriate to cut. The method assumes that some intermediate nodes can do aggregation so that they send less than they receive; otherwise the probability of initiating channel switch will be zero for all intermediate nodes. When a node switches, nodes that communicate heavily with it also switch. If a transmitter wants to send data to a receiver on a different channel, the transmitter simply switches to the receiver's channel. As a result, the transmitter may miss some packets destined for it.

Similarly, **TMCP** [61] partitions a sensor network to K vertex-disjoint trees, each of which is assigned a channel. Different from K-way cut problem that aims at minimizing inter-channel communication, TMCP tries to minimize the maximum intra-tree interference value among all trees. TMCP defines the interference value of a tree as the maximum interference value among all non-leaf nodes and the interference value of a node is defined as the number of nodes within the node's interference range. The definition is not very accurate since a node's neighboring nodes are potential but not actual interferers. Two nodes with the same number of neighboring nodes can experience different levels of interference depending on their neighbors' activities. Therefore, the metric does not reflect the actual interference intensity.

Comparing with node-based multichannel protocols such as MMSN [56], link-based multichannel assignments require fewer number of orthogonal channels and incur fewer channel switches. Currently, they only divide nodes into K groups given K available channels. To further increase parallel transmissions, spatial reuse of channels can be considered.

GBCA [62] proves that finding a channel assignment that minimizes the total interference in a network is NP-hard. Therefore, it models a channel assignment game to solve the problem with a suboptimal result. Since routing in WSNs is generally static and of tree/forest structure, it models the parent of each Parent-Children Set (PCS) as a player. These players negotiate channel usage base on the Best Response (BR) dynamic, trying to maximize their payoff in each iteration. It proves that there exists a Nash Equilibrium (NE) in the game and the solution converges to the NE in polynomial time. However, same as TMCP [61], the interference is measured based on network topology without considering current traffic

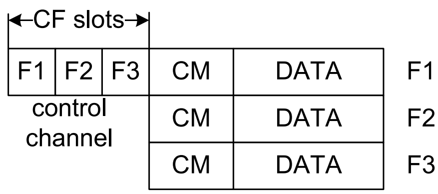


Fig. 13. Time slot structure in MC-LMAC.

pattern. As a result, it may not reach the best solution for current interference condition. To adapt to dynamic traffic, the game theoretic approach may incur excessive overhead.

C. TDMA/FDMA for Sending

Toggle snooping and toggle transmission introduce additional overhead for each packet transmission. To avoid frequent channel switching before data transmission, channel assignment can be combined with time slot assignment. In **MC-LMAC** [63], time is slotted and several consecutive slots are organized as a frame. When a node joins the network, it listens for neighbors' occupied slots in frames, which use a bit 0/1 to indicate whether a slot is occupied. The slot occupancy is associated with frequencies. A node chooses a slot/channel pair that is not occupied by any of its neighbors within its two-hop communication neighborhood. A node may choose the same slot that is used by its neighbors on a different channel. To ensure that two direct neighbors can communicate, a node must exclude slots that are used by its direct neighbors, but a node can share the same slot with its two-hop neighbors. To reduce the possibility that a node is addressed by multiple senders, MC-LMAC utilizes the data gathering tree structure. It lets a node choose the slot that is not used by any other children of the same parent with a higher priority. The control overhead of broadcasting slot occupancy on all channels is slightly high. Same as traditional TDMA, the channel utilization under low contention is low and this leaves room for further improvement.

The aforementioned slot/channel selection allows parallel data transmission, which is achieved by letting all nodes listen on the control channel first in each slot. Each slot has a Common Frequency (CF) period for notifying receivers of data reception. As shown in Fig. 13, a CF period is composed of k small slots where k is equal to the total number of available channels. Multiple nodes may control the current time slot, but each of them is on a different channel (i.e., each chooses an unique channel/slot pair). Therefore, each sender of the current slot can use the CF slot that is corresponding to its controlled channel number to broadcast its target receiver's ID. After the CF period, each pair of sender and receiver switch to the sender's controlled channel for data transmission. Note that the control channel can be reused for data transmission. The Control Message (CM) period is reserved for control information transmission (e.g., the acknowledgement to the sender and the occupied slots vectors). The protocol is suitable for scenarios of high demands on the medium where the overhead is compensated by parallel data transmissions.

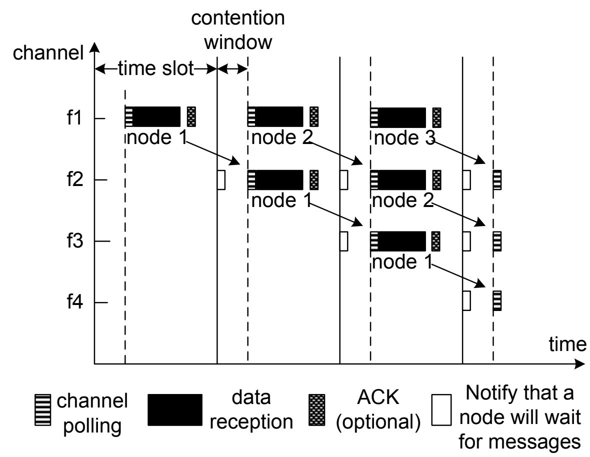


Fig. 14. Channel hopping mechanism in Y-MAC.

D. TDMA/FDMA for Receiving

A node may have different amount of data to send in different periods. To adapt to dynamic traffic, it is better to allocate network resources accordingly. Some protocols adopt the frequency hopping technique.

Y-MAC [64] is a TDMA-based dynamic channel selection scheme, where time slots are assigned to nodes for receiving. Recall that the benefit is fewer nodes wake up at each slot compared with scheduling senders, but multiple senders have to contend for sending. In Y-MAC, each time frame consists of a broadcast period and a unicast period. Both of them are further divided into small slots. To support broadcast, all nodes listen on the base channel during the broadcast period. Nodes that have broadcast packets contend for sending in each broadcast slot. If there is no broadcast traffic, the radio can be turned off for energy saving.

The slots of the unicast period are assigned in the same graph-coloring way that each node owns an exclusive slot within its two-hop communication neighborhood. To make use of slots owned by neighbors, Y-MAC lets a node hop to the next channel to receive another packet in the next slot if it receives a packet on the current channel. As shown in Fig. 14, the procedure starts with the base channel in a node's assigned slot. Any node must listen on the base channel in its assigned slot. If a node receives a packet in its assigned slot, it hops to the next channel in the next slot for potential data reception. Although the next slot is owned by one of its neighbors, its data reception will not interfere with its neighbor's data reception because they are on different channels. However, if two receivers own the same slot, they will always hop to the same channel and their senders may keep interfering with each other. In addition, Y-MAC does not discuss when a node should stop receiving and change to sending, and which channel can be used when all channels have been visited. The channel hopping method thus needs further investigation.

MuChMAC [65] is another hybrid design of TDMA and FDMA. Each node independently chooses its receiving channel for each slot. To make it easy to learn the hopping sequence of a neighbor, the hopping sequence is produced by a pseudo-random generator using the ID and the slot number as the input. To support broadcast, a broadcast slot is inserted

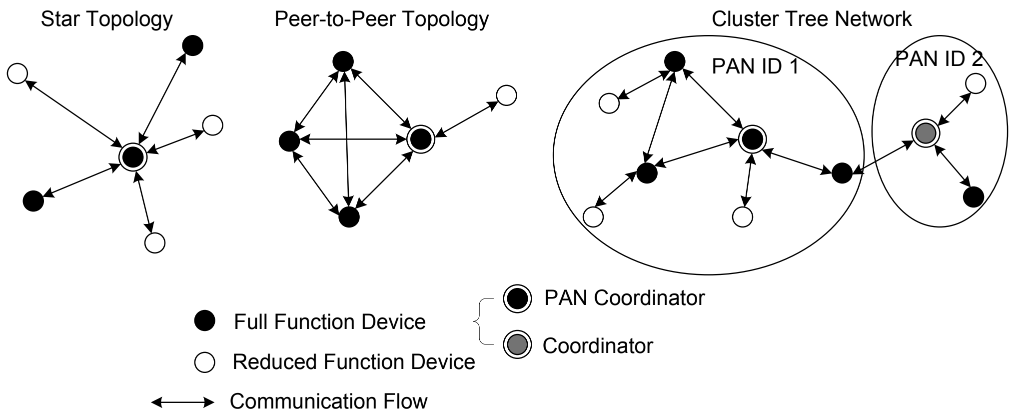


Fig. 15. IEEE 802.15.4 topologies and their use.

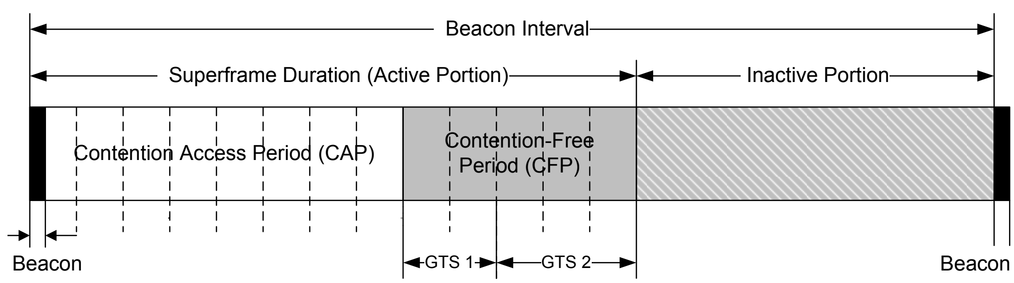


Fig. 16. An example of the IEEE 802.15.4 superframe structure.

every u unicast slots. The channel numbers of these broadcast slots also follow a pseudo-random hopping sequence and the sequence is identical for all nodes.

Because nodes choose channels randomly for each slot, two nodes may choose the same channel for a particular slot. If they are within the interference range of each other, the amount of parallel traffic is reduced. To address the problem, MuChMAC divides a time slot into several subslots. A node will choose one of the subslots to wake up. Obviously, the chance that two nodes choose the same channel/subslot pair is low, but many subslots are wasted. Therefore, how to assign channel/slot to nodes efficiently remains an open issue.

E. Summary

Static channel assignments may be optimized for a certain topology, but considering that interference condition changes in accordance with traffic, it is better to flexibly allocate channels among nodes. Dynamic channel allocation, however, is a tough task same as the adaptive time slot assignment. In addition to channel assignment, cross-channel communication is another challenge in multichannel MAC protocols. Currently, toggle snooping/transmission may draw energy quickly. An alternative way is to combine TDMA with FDMA. A good example of scheduling senders is introduced in MC-LMAC [63], but overhead is slightly high and channel/slot utilization is low under low contention. If slots and channels are assigned for receiving as in Y-MAC [64] and MuChMAC [65], contentions among senders need to be addressed if two receivers happen to hop to the same channel.

VI. Standards

Along with academic researches, standards are developed to speed up utilization of WSNs. We briefly discuss some industrial progresses in this section.

A. IEEE 802.15.4 and ZigBee

The **IEEE 802.15.4** standard [66] is proposed for Low-Rate Wireless Personal Area Networks (LR-WPANs). The key features of low cost, low power consumption, and low data rate typically fit the requirements of WSNs. Two different device types are defined in IEEE 802.15.4: a full-function device (FFD) and a reduced-function device (RFD). A FFD is a device that can serve as a PAN coordinator or a coordinator while a RFD is intended for extremely simple applications and implemented with minimal resources and memory capacity.

The standard supports two topologies: the star topology and the peer-to-peer topology as shown in Fig. 15. In the star topology, all communications must go through the PAN coordinator which is the central controller of the WPAN. In the peer-to-peer topology, any device can communicate with any other devices if they are in the communication range of one another, but still the PAN coordinator must be present to choose PAN identifier and manage the network. More complex network structures can be constructed out of the peer-to-peer topology. An example is the multicluster tree where the PAN coordinator can instruct a device to become the coordinator of a new cluster adjacent to the first one. Other devices gradually connect to the network and extend the coverage area.

Two modes of operation exist in IEEE 802.15.4, namely the beacon-enabled mode and the nonbeacon-enabled mode. In the nonbeacon-enabled star topology, the PAN coordinator stays awake. Associated devices send to the PAN coordinator or request data from the PAN coordinator in a contention-based way, applying the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In the nonbeacon-enabled peer-to-peer topology, the devices wishing to communicate have to keep their radio on constantly or some synchronization mechanisms have to be employed. How to achieve synchronization in the peer-to-peer topology, however, is beyond the scope of the standard. To facilitate low-power operation, a superframe structure is defined in beacon-enabled mode. The coordinator broadcasts beacons periodically and all associated devices are synchronized by the beacons. As shown in Fig. 16, the superframe has an active and an inactive portion. The active portion is divided into the contention access period (CAP) and the contention-free period (CFP). All contention-based transactions must be completed before the CFP begins, which is reserved for low-latency applications or applications requiring specific data bandwidth. A device can request a guaranteed time slot (GTS) in the CAP. A coordinator can allocate up to seven GTSs, and a GTS is allowed to occupy more than one slot period. The optional superframe structure allows the coordinator to switch to the low-power sleep mode periodically and provide the data with quality of service (QoS).

In multihop networks, the beacon-enabled mode is not that straightforward. Beacons of one coordinator may collide with beacons from other coordinators or data frames from nodes that are associated with other coordinators. The beacon collisions are intolerable because beacons provide synchronization and define the superframe structure (i.e., CAP, CFP, and inactive period). In addition, if two coordinators' superframes overlap with each other, the CFP is no longer collision-free. These problems motivate researchers to work on beacon scheduling [67] [68] [69] [70] [71]. Generally, there are two methods. The first approach is superframe duration scheduling (SDS) where each coordinator transmits its superframe during the inactive period of its neighbors in the two-hop communication neighborhood as illustrated in Fig. 17(a). A long inactive period used in this approach may introduce significant delay in data transmission. The second approach is to create a beacon-only-period (BOP) at the start of the superframe where each coordinator selects a free time-slot to transmit its own beacon and thus avoid collisions (Fig. 17(b)). Because the active periods overlap with each other, how to allocate GTSs efficiently is still under research [72].

The IEEE 802.15.4 standard defines the physical layer (PHY) and the medium access control (MAC) sublayer specifications. **ZigBee** [73] builds up the network layer and the application layer on top of the IEEE 802.15.4 physical and data link layers, thus defining a full protocol stack for LR-WPANs. In industrial applications, comparison [74] shows that Zigbee is not suitable because it cannot meet the stringent industrial requirements, especially the deterministic delay and high reliability. Some industrial wireless standards thus highlight their features of strict latency and high reliability for factory automation.

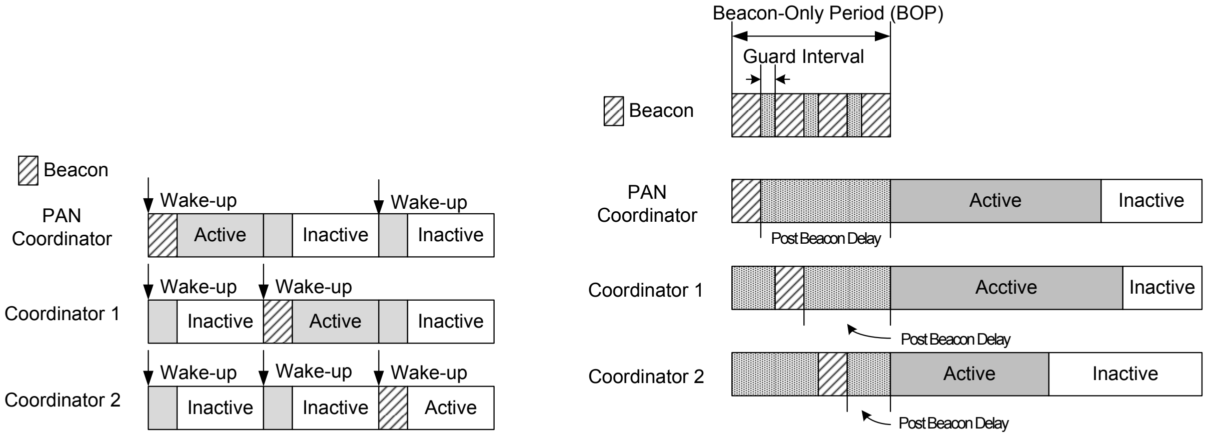
B. WirelessHART and ISA-100.11a

WirelessHART [75], released by the HART Communication Foundation in September 2007, is the first open wireless communication standard for process control and related applications. Soon **ISA-100.11a** [76] was approved by the ISA Standards and Practices Board as an official ISA standard in September 2009. In general, ISA-100.11a is more flexible by providing more configurable parameters than WirelessHART. For example, the time slot size is fixed at 10 ms in WirelessHART while in ISA100.11a it is configurable on a per-superframe base. Both standards use Time Division Multiple Access (TDMA) with frequency hopping for channel access in the 2.4 GHz band. The combination of direct sequence spread spectrum (DSSS) and frequency-hopping spread spectrum (FHSS) makes WirelessHART and ISA-100.11a more robust to interference in harsh industrial environments. IEEE 802.15.4 defines 16 channels (i.e., channels 11-26) in the 2.4 GHz band. WirelessHART uses channels 11-25 because channel 26 is not legal to use in some countries. In ISA-100.11a, channel 26 is optional.

WirelessHART adopts a slotted hopping scheme as shown in Fig. 18(a) where the channel is changed in each slot while ISA100.11a defines three channel hopping schemes including slotted hopping, slow hopping, and hybrid hopping. As shown in Fig. 18(b), a device stays on a channel for a collection of contiguous time slots in slow hopping. This supports devices with imprecise timing settings or devices that want to join the network. A node can mainly scan these channels for the advertisement of a target network. The slow hopping also serves as a way to improve support for event-based traffic. Usually a group of devices share a slow hopping period in a contention-based way, that is, transmissions in a slow hopping period is CSMA/CA based. When an event triggers the need for a device to immediately transmit a data packet or an alarm, the device does not need to wait for the next time slot that is assigned to it, thereby reducing the latency. However, slow hopping increasing devices' energy consumption as they have to listen to the channel for possible incoming packets. Therefore, hybrid hopping combines slotted hopping and slow hopping (Fig. 18(c)), in which slotted hopping accommodates periodical messages and slow hopping improves support for less predictable messages. In WirelessHART, the channel hop patterns is controlled by the network manager. In ISA100.11a, five preprogrammed hopping patterns are defined as below.

- Pattern 1: 19, 12, 20, 24, 16, 23, 18, 25, 14, 21, 11, 15, 22, 17, 13 (, 26)
- Pattern 2: pattern 1 in reverse
- Pattern 3: 15, 20, 25 (intended for slow-hopping channels)
- Pattern 4: pattern 3 in reverse
- Pattern 5: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 (, 26).

Pattern 1 ensures that any retransmission will not be performed on the same IEEE 802.11 channel so as to avoid constant interference from IEEE 802.11 devices. In addition, both WirelessHART and ISA-100.11a support blacklisting that eliminates channels of significant interference. The frequency diversity together with the time diversity make WirelessHART



(a) Superframe Duration Scheduling (SDS)

(b) Beacon-only-period Scheduling (BOP)

Fig. 17. Beacon scheduling approaches.

and ISA-100.11a favorable in industrial applications. A systematic comparison between WirelessHART and ISA-100.11a can be found in [77]. The **IEEE 802.15.4e** is also under development for considering the real-time aspects [78]. Surveys on WSN technologies for industrial automation are also available in [79], [80].

VII. Conclusion and Future Work

In this article we have surveyed the recent developments in the design of WSN MAC protocols. We classify WSN MAC protocols into four categories: asynchronous, synchronous, frame-slotted, and multichannel. The classification aims at identifying the research trend of WSN MAC protocols based on techniques they employ. We show that throughput and delay metrics are also being considered besides energy efficiency in WSN MAC protocols when time synchronization and multichannel are gradually supported.

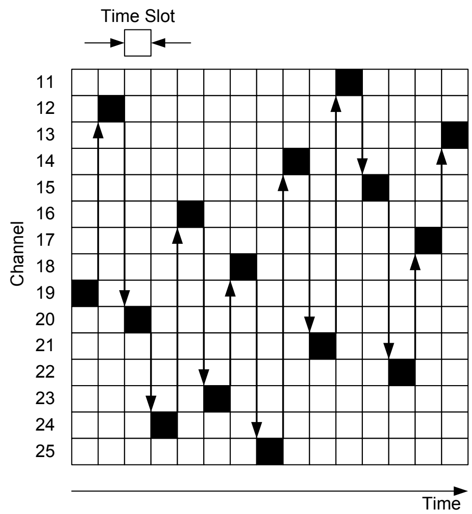
Energy saving by putting nodes into low-power sleep mode periodically is a fundament mechanism in WSN MAC protocols. Asynchronous MAC protocols have integrated the prediction mechanism to estimate the best wake-up time for sending. Recently, the responsibility of establishing communication is gradually shifted from the sender side to the receiver side. Although the transition is reasonable since only when the receiver is ready to receive the sending is effective, quick and proper response to link breaks and target receivers' leave is a potential issue. It is also challenging to minimize the idle listening of senders while guaranteeing that the sender will not miss the beacon of the receiver. In other words, although LPP-based MAC protocols provide higher throughput, the energy efficiency is generally lower than that of LPL-based MAC protocols in light traffic load scenarios because channel sampling duration is short in LPL-based MAC protocols but idle listening duration for a beacon is slightly longer. Which one is more appropriate is application dependent. In asynchronous MAC protocols, nodes' random and independent schedules result in a long initial delay for event report. These are the potential research directions in asynchronous MAC protocols.

Synchronous MAC protocols synchronize a cluster of nodes to a common schedule. The trend in this branch is to move data delivery from the common active period to the sleep period. The common active period is used for arranging data transmission in the sleep period. Current scheduling methods are good at delay reduction for individual packets but are very vulnerable to interference. It is highly demanded to design a more robust scheduling scheme that could efficiently utilize the sleep period for data transmission. Synchronous MAC protocols may gradually evolve to traffic-adaptive TDMA schemes, which utilize the time in the sleep period wisely by updating traffic condition during the common active period. In fact, we have seen such a paradigm in TRAMA [50], but the channel utilization is low due to the inefficient slot assignment and the delay is high because nodes that fail to claim slots in the random period cannot access the channel in the entire cycle. Moreover, the downstream nodes are not notified in advance and thus they will not reserve time slots for use. The end-to-end delay is high. More future work can be done under this framework.

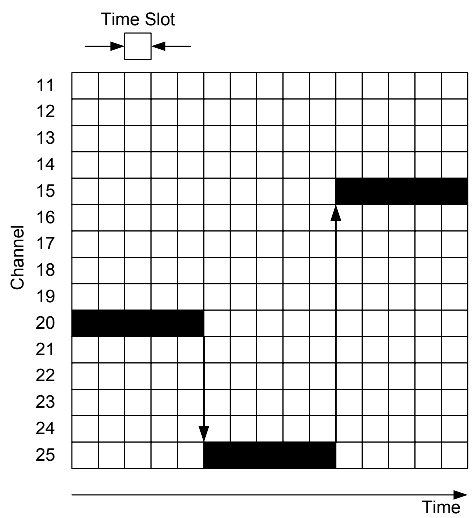
In TDMA, slot stealing helps to improve the channel utilization under low contention, but it incurs high energy consumption as all nodes have to wake up to receive potential data in a slot. Assigning time slots only to nodes that are on active routes can increase channel utilization without incurring too much overhead, but there are several issues that require further study. First, a mechanism is needed to rapidly inform nodes of incoming traffic. Second, a lightweight time slot assignment algorithm is needed for dynamic time slot assignment among nodes on the active routing paths. Third, how to utilize slots that are claimed by a node but left unused is of interest.

Scheduling senders in TDMA wakes all potential receivers up. Although scheduling receivers in TDMA leads to fewer operations of turning radio on, the strategy deviates away from the original goal of TDMA (i.e., collision-free data transmission). It is thus necessary to address contentions among multiple senders efficiently.

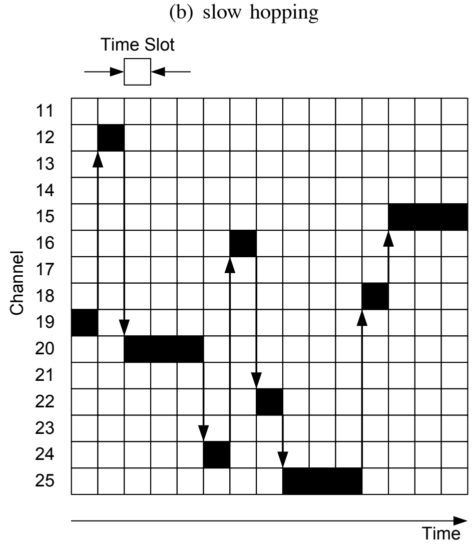
To further boost network throughput, multichannel MAC protocols became a hot topic. One challenge for future research is the design of dynamic channel allocation algorithms that adapt to the dynamic traffic of sensor networks. The interference analysis of prior work is based on network



(a) slotted hopping



(b) slow hopping



(c) hybrid hopping

Fig. 18. Frequency-hopping patterns.

topology, and thus the solutions may not be the best fit for the current interference condition. It is desirable to allocate network resources flexibility according to the traffic but it is also challenging to design such a dynamic channel allocation algorithm with low overhead.

Another challenge in multichannel MAC protocols is cross-channel communication. For cost consideration, a sensor node is usually equipped with a half-duplex radio transceiver and the transceiver is unable to listen on multiple channels at the same time. This implies that a node has to switch back and forth between two channels to check whether it has data to receive and whether the target receiver's channel is clear. A node is also responsible for letting its neighbors know that it is going to send and thus is unable to receive. These operations increase the overhead for a packet transmission and frequent channel switch makes a node consume energy faster than others. Therefore, how to perform cross-channel communication efficiently is worth further study.

The channel hopping used in industry standards is robust to interference. However, the channel hopping strategies and time slot assignment are under control of a central unit (i.e., network/system manager). To extend the network size, reliable distributed TDMA/FDMA algorithms are highly desired.

Finally, most previous studies have only considered orthogonal channels. However, it is possible to utilize partially overlapped channels. Some studies [81] [82] have shown that the overall throughput can be improved by using non-orthogonal channels even though the throughput on each individual channel may slightly drop. Although the introduction of non-orthogonal channels has the potential to further boost network capacity, how to utilize them wisely is an open issue. As it becomes harder to model channel interference when co-channel interference is introduced, it is more challenging to design channel allocation algorithms for sensor networks, especially when dynamic channel allocation is desired.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.

[2] "SENSEI (Integrating the Physical with the Digital World of the Network of the Future)," http://www.sensei-project.eu.

[3] P. Naik and K. M. Sivalingam, *A survey of MAC protocols for sensor networks*. Kluwer Academic Publishers, 2004, pp. 93–107.

[4] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, 2006.

[5] K. G. Langendoen, *Medium access control in wireless sensor networks*. Nova Science Publishers, Inc., 2008, pp. 535–560.

[6] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC essentials for wireless sensor networks," *IEEE Commun. Surveys Tutorials*, vol. 12, no. 2, pp. 222 –248, 2010.

[7] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, November/December 2002.

[8] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in *Proc. ICC*, 2002.

[9] ——, "Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks," in *Proc. 7th International Symposium on Computers and Communications (ISCC)*, 2002, pp. 685–692.

[10] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. SenSys*, 2004.

[11] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Trans. Mobile Computing*, vol. 1, no. 1, pp. 70–80, January-March 2002.

[12] S. Lim, S. Kim, J. Cho, and S. An, "Medium access control with an energy-efficient algorithm for wireless sensor networks," in *Personal Wireless Communications*, ser. Lecture Notes in Computer Science, P. Cuenca and L. Orozco-Barbosa, Eds. Springer Berlin / Heidelberg, 2006, vol. 4217, pp. 334–343.

[13] A. Bachir, D. Barthel, M. Heusse, and A. Duda, "Micro-frame preamble MAC for multihop wireless sensor networks," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 7, June 2006, pp. 3365 –3370.

[14] M. Awenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the efficiency of preamble sampling protocols for wireless sensor networks," in *Proc. First Mobile Computing and Wireless Communication International Conference (MCWC)*, September 2006, pp. 117 –122.

[15] K.-J. Wong and D. K. Arvind, "SpeckMAC: low-power decentralised MAC protocols for low data rate transmissions in specknets," in *Proc. 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, 2006, pp. 71–78.

[16] S. Lim, Y. Ji, J. Cho, and S. An, "An ultra low power medium access control protocol with the divided preamble sampling," in *Ubiquitous Computing Systems*, ser. Lecture Notes in Computer Science, H. Youn, M. Kim, and H. Morikawa, Eds. Springer Berlin / Heidelberg, 2006, vol. 4239, pp. 210–224.

[17] X. Shi and G. Stromberg, "SyncWUF: An ultra low-power MAC protocol for wireless sensor networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 1, pp. 115 –125, January 2007.

[18] S. Mahlknecht and M. Bock, "CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks," in *Proc. IEEE International Workshop on Factory Communication Systems*, September 2004, pp. 73 – 80.

[19] E.-Y. Lin, J. Rabaey, and A. Wolisz, "Power-efficient rendez-vous schemes for dense wireless sensor networks," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 7, June 2004, pp. 3769 – 3776.

[20] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. SenSys*, 2006, pp. 307–320.

[21] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, "A wireless sensor MAC protocol for bursty data traffic," in *Proc. IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, September 2007, pp. 1 –5.

[22] H. Wang, X. Zhang, F. Nait-Abdesselam, and A. A. Khokhar, "DPS-MAC: An asynchronous MAC protocol for wireless sensor networks." in *Proc. 14th International Conference on High Performance Computing (HiPC)*, 2007, pp. 393–404.

[23] S. Liu, K.-W. Fan, and P. Sinha, "CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks," in *Proc. 4th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2007, pp. 11–20.

[24] C. J. Merlin and W. B. Heinzelman, "Network-aware adaptation of MAC scheduling for wireless sensor networks," in *Proc. 3rd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS Poster Session)*, June 2007.

[25] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proc. 9th International Symposium on Computers and Communications (ISCC)*, vol. 1, June-July 2004, pp. 244–251.

[26] ——, "Low power downlink MAC protocols for infrastructure wireless sensor networks," *Mob. Netw. Appl.*, vol. 10, pp. 675–690, October 2005.

[27] R. Musaloiu-E, C.-J. Liang, and A. Terzis, "Koala: Ultra-low power data retrieval in wireless sensor networks," in *Proc. IPSN*, 2008, pp. 421–432.

[28] B. Jang, J. B. Lim, and M. L. Sichitiu, "AS-MAC: An asynchronous scheduled MAC protocol for wireless sensor networks," in *Proc. MASS*, 2008.

[29] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proc. SenSys*, 2008, pp. 1–14.

[30] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. SenSys*, 2010, pp. 1–14.

[31] P. Huang, C. Wang, L. Xiao, and H. Chen, "RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks," in *Proc. IWQoS*, 2010, pp. 1–9.

[32] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proc. INFOCOM*, 2011, pp. 1305–1313.

[33] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. INFOCOM*, 2002, pp. 1567–1576.

[34] ——, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[35] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. SenSys*, 2003, pp. 171–180.

[36] S. Du, A. K. Saha, and D. B. Johnson, "RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks," in *Proc. INFOCOM*, 2007, pp. 1478–1486.

[37] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proc. MobiHoc*, 2008, pp. 53–62.

[38] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proc. IPDPS*, 2004.

[39] N. Vasanthi and S. Annadurai, "Energy efficient sleep schedule for achieving minimum latency in query based sensor networks," in *Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 2, June 2006, pp. 214–219.

[40] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proc. SenSys*, 2006, pp. 321–334.

[41] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, March 2004, pp. 1534–1539 Vol.3.

[42] Z. Chen, C. Hu, J. Liao, and S. Liu, "Protocol architecture for wireless body area network based on nRF24L01," in *Proc. IEEE International Conference on Automation and Logistics (ICAL'08)*. IEEE, 2008, pp. 3050–3054.

[43] O. Omeni, A. Wong, A. Burdett, and C. Toumazou, "Energy efficient medium access protocol for wireless medical body area sensor networks," *IEEE Trans. Biomedical Circuits and Systems*, vol. 2, no. 4, pp. 251–259, 2008.

[44] A. Milenkovic, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Computer Communications*, vol. 29, no. 13-14, pp. 2521–2533, 2006.

[45] B. Latre, B. Braem, I. Moerman, C. Blondia, E. Reusens, W. Joseph, and P. Demeester, "A low-delay protocol for multihop wireless body area networks," in *Proc. 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, 2007, pp. 1–8.

[46] S. Marinkovic, E. Popovici, C. Spagnol, S. Faul, and W. Marnane, "Energy-efficient low duty cycle MAC protocol for wireless body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 915–925, 2009.

[47] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a hybrid MAC for wireless sensor networks," in *Proc. SenSys*, 2005, pp. 90–101.

[48] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proc. MobiHoc*, 2006, pp. 190–201.

[49] S. Gobriel, D. Mosse, and R. Cleric, "TDMA-ASAP: Sensor network TDMA scheduling with adaptive slot-stealing and parallelism," in *Proc. ICDCS*, 2009, pp. 458–465.

[50] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in *Proc. SenSys*, 2003, pp. 181–192.

[51] L. Bao and J. J. Garcia-Luna-Aceves, "A new approach to channel access scheduling for Ad Hoc networks," in *Proc. MobiCom*, 2001, pp. 210–221.

[52] S. Chatterjea, L. van Hoesel, and P. Havinga, "AI-LMAC: an adaptive, information-centric and lightweight MAC protocol for wireless sensor networks," in *Proc. 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*, 2004, pp. 381–388.

[53] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen, "TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks," in *Proc. PerCom*, 2009.

[54] G. P. Halkes and K. G. Langendoen, "Crankshaft: an energy-efficient MAC-protocol for dense wireless sensor networks," in *Proc. EWSN*, 2007, pp. 228–244.

[55] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. IPDPS*, 2005.

[56] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in *Proc. INFOCOM*, 2006, pp. 1–13.

[57] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proc. MobiHoc*, 2004, pp. 222–233.
[58] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic, "TMMAC: An energy efficient multi-channel MAC protocol for ad hoc networks." in *Proc. ICC*, 2007, pp. 3554–3561.
[59] "CC2420 datasheet," http://www.ti.com.
[60] H. K. Le, D. Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *Proc. IPSN*, 2008, pp. 70–81.
[61] Y. Wu, J. A. Stankovic, T. He, J. Lu, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proc. INFOCOM*, 2008, pp. 1193–1201.
[62] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun, "Multi-channel assignment in wireless sensor networks: a game theoretic approach," in *Proc. INFOCOM*, 2010, pp. 1127–1135.
[63] O. D. Incel, L. van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: A multi-channel MAC protocol for wireless sensor networks," *Ad Hoc Netw.*, vol. 9, pp. 73–94, January 2011.
[64] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks," in *Proc. IPSN*, 2008, pp. 53–63.
[65] J. Borms, K. Steenhaut, and B. Lemmens, "Low-overhead dynamic multi-channel MAC for wireless sensor networks," in *Proc. EWSN*, 2010, pp. 81–96.
[66] *IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4-2011, September 2011.
[67] B. Villaverde, R. De Paz Alberola, S. Rea, and D. Pesch, "Experimental evaluation of beacon scheduling mechanisms for multihop IEEE 802.15.4 wireless sensor networks," in *Proc. 4th International Conference on Sensor Technologies and Applications (SENSORCOMM)*, July 2010, pp. 226–231.
[68] R. Burda and C. Wietfeld, "A distributed and autonomous beacon scheduling algorithm for IEEE 802.15.4/ZigBee networks," in *Proc. IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS)*, October 2007, pp. 1–6.
[69] H.-I. Jeon and Y. Kim, "BOP (beacon-only period) and beacon scheduling for MEU (mesh-enabled USN) devices," in *Proc. 9th International Conference on Advanced Communication Technology*, vol. 2, February 2007, pp. 1139 –1142.
[70] E.-J. Kim and H.-H. Choi, "EBBS: Energy-efficient BOP-based beacon transmission scheduling for WSNs," in *Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, September 2008, pp. 1–6.
[71] A. Koubaa, A. Cunha, and M. Alves, "A time division beacon scheduling mechanism for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks," in *Proc. 19th Euromicro Conference on Real-Time Systems (ECRTS)*, July 2007, pp. 125–135.
[72] B. C. Villaverde, S. Rea, and D. Pesch, "Guaranteeing reliable communications in mesh beacon-enabled IEEE802.15.4 WSN for industrial monitoring applications," in *Proc. 2nd International ICST Conference on Ad Hoc Networks (ADHOCNETS)*, 2010, pp. 359–370.
[73] "ZigBee Alliance," http://www.zigbee.org/.
[74] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of WirelessHART and ZigBee for industrial applications," in *Proc. IEEE International Workshop on Factory Communication Systems (WFCS)*, May 2008, pp. 85 –88.
[75] *HART Field Communication Protocol Specification, Revision 7.0*, HART Communication Foundation Std., September 2007.
[76] *Wireless Systems for Industrial Automation: Process Control and Related Applications*, Std. ISA-100.11a-2009, 2009.
[77] S. Petersen and S. Carlsen, "WirelessHART Versus ISA100.11a: The format war hits the factory floor," *IEEE Industrial Electronics Mag.*, vol. 5, no. 4, pp. 23–34, December 2011.
[78] F. Chen, R. German, and F. Dressler, "Towards IEEE 802.15.4e: A study of performance aspects," in *Proc. 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, April 2010, pp. 68–73.
[79] D. Christin, P. S. Mogre, and M. Hollick, "Survey on wireless sensor network technologies for industrial automation: The security and quality of service perspectives," *Future Internet*, vol. 2, no. 2, pp. 96–125, 2010.
[80] M. Paavola and K. Leiviska, "Wireless sensor networks in industrial automation," *Factory Automation*, 2010.
[81] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh, "Partially overlapped channels not considered harmful," in *Proc. SIGMETRICS*, 2006, pp. 63–74.
[82] X. Xu, J. Luo, and Q. Zhang, "Design of non-orthogonal multi-channel sensor networks." in *Proc. ICDCS*, 2010, pp. 358–367.

Pei Huang received the BS with distinction and MS degrees in electrical engineering from Southwest Jiaotong University, China, in 2005 and 2008, respectively. Currently, he is a research assistant with the Michigan State University Computer Science and Engineering Department, where he is pursuing his PhD degree. His research interests include the areas of wireless communication, wireless ad hoc and sensor networks, and cognitive radio networks.

Li Xiao received the BS and MS degrees in computer science from Northwestern Polytechnic University, China, and the PhD degree in computer science from the College of William and Mary in 2002. She is an associate professor of computer science and engineering at Michigan State University. Her research interests are in the areas of distributed and networking systems, overlay systems and applications, and wireless networks. She is a senior member of the IEEE.

Soroor Soltani is a PhD candidate in the Electrical and Computer Engineering Department at Michigan State University, East Lansing, MI. She graduated with BE in Electrical and Communication Engineering with distinction from Kuwait University, Kuwait in 2005. She received her MS in Electrical Engineering from the Michigan State University, MI, in 2008. Her expertise and research interests lie in wireless cognitive radio mesh networks operating in urban areas, and multimedia communication over stochastic networks. She is a student member of the IEEE.

Matt W. Mutka received the BS degree in electrical engineering from the University of Missouri-Rolla, the MS degree in electrical engineering from Stanford University, and the PhD degree in Computer Sciences from the University of Wisconsin-Madison. He is on the faculty of the Department of Computer Science and Engineering at Michigan State University, where he is currently professor and chairperson. He has been a visiting scholar at the University of Helsinki, Helsinki, Finland, and a member of technical staff at Bell Laboratories in Denver, Colorado. His current research interests include mobile computing, sensor networking, wireless networking and multimedia networking.

Ning Xi received his D.Sc. degree in Systems Science and Mathematics from Washington University in St. Louis, Missouri in December, 1993, and B.S. degree in electrical engineering from Beijing University of Aeronautics and Astronautics. He is the University Distinguished Professor and John D. Ryder professor of Electrical and Computer Engineering at Michigan State University. Currently, he serves as the Head and Chair Professor of the Department of Mechanical and Biomedical Engineering, the City University of Hong Kong. Dr. Xi is a fellow of IEEE. Currently, his research interests include robotics, manufacturing automation, micro/nano manufacturing, nano sensors and devices, and intelligent control and systems.