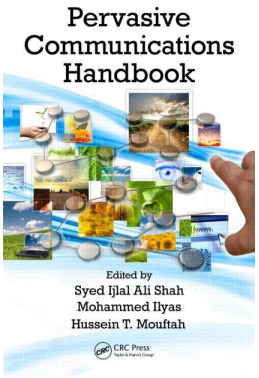

Pervasive Communications Handbook

Syed Ijlal Ali Shah, Mohammad Ilyas, Hussein T. Mouftah

Medium Access Control Protocols for Wireless Sensor Networks in a Pervasive Computing Paradigm

Publication details
https://test.routledgehandbooks.com/doi/10.1201/b11271-5
Muhammad K. Dhodhi, Syed Ijlal Shah, Marwan Fayed
Published online on: 14 Nov 2011

3

Medium Access Control Protocols for Wireless Sensor Networks in a Pervasive Computing Paradigm

Muhammad K. Dhodhi
Ross Video Limited

Syed Ijlal Shah
Freescale Semiconductor

Marwan Fayed
University of Stirling

3.1 Introduction

In his seminal work [1], Mark Weiser introduced the vision of the twenty-first century's ubiquitous and pervasive computing. In it he wrote, "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." This invisible computing is accomplished by means of "embodied virtuality," the process of drawing computers into the physical world [2]. In context-aware pervasive computing, devices must be ever-present. They are critical enough to our activities; yet, they must be taken for granted and effectively disappear into the background.

The technological advances needed to create a pervasive computing environment include four broad areas: such as hardware devices, networking, middleware, and applications [3]. Mobility, constant availability, invisibility, adaptability, and privacy are salient characteristics of pervasive computing environments that are intended to free the user from the distractions of having to interact with computers and machines.

Recent advances in VLSI circuit and fabrication technology have produced small and sophisticated devices, such as micro-sensors, to provide the necessary hardware infrastructure for creating pervasive computing environments. Nanotechnology is further helping the proliferation of tiny devices in our daily lives and systems. Not only is the size of these devices shrinking, but also the processing capacity follows the trend of their larger counterparts and doubles every 2 years. This increased processing capacity is in turn enabling the tiny devices to run complex applications and protocols.

Wireless sensor networks (WSNs), whether they are employed in the home, office, industrial plants, healthcare, or the habitat, form one of the building blocks for constructing a truly pervasive environment. The sensors in a WSN, given the processing capabilities that can be embedded into them, will have the ability to act on their own given high-level guidance from their users. They will also be able to detect and adapt to changes in their environment.

WSNs are an emergent multi-disciplinary field that impacts all the layers of the OSI network model. WSNs are characterized mainly by limited availability of energy to power these devices, lossy or challenging environments, and loosely connected networks. Adding to these challenges, sensors have limited radio range and limited storage capability [4], although, with advancement of technology, storage and processing capacities are increasing. Nevertheless, these devices fall well short of the capabilities of their larger counterparts that are available in large wired, power grid connected devices such as radar, or closed circuit TV cameras. WSNs applications, both continuous monitoring and event-driven, are impacted by the resource-constrained nature of the sensor nodes.

Medium access control, or MAC, protocols play a crucial role in the dissemination and transfer of the collected data in an efficient manner. This is because it is the MAC protocol that specifies how nodes share the wireless channel, which in turn dictates the active/sleep modes of the radio transceiver in a sensor node.

In the sections to follow, we will describe major research issues and challenges as well-as proposed solutions for the MAC layer in sensor networks.

3.2 MAC Protocols

Nodes in a sensor network collect data and communicate the observed data with one another using short haul radio links with a limited range and bandwidth. Due to limited energy supply, one of the major design goals in sensor networking is energy conservation. MAC protocols play a crucial role in the dissemination and transfer of data in an efficient manner. The MAC protocol is responsible for negotiating and obtaining access to the communication channel, over which a single node at a time may transmit. Effectively, it is the MAC that specifies how the wired or wireless channel is shared. Because of this, MAC protocols are tightly coupled to the active/sleep modes of the radio transceiver in sensor nodes. This coupling is important as it is the radio transceiver that

dominates with respect to energy consumption. For this reason, it is advantageous to power down the transceiver whenever possible. This is often referred to as sleep-mode. On the other hand, the receiver of the sensor nodes must be turned-on (active mode) so that neighboring nodes can communicate with each other. As a result the desire to power-down a transceiver conflicts with the need to listen for transmissions. Therefore, nodes must coordinate or compete to establish and maintain active/sleep schedules with their neighbors in order to communicate messages.

Beside energy-efficiency, MAC layer protocols need to meet several other important design goals for their applicability in the pervasive computing paradigm. The additional design goals include mobility, adaptability to changes, network size, network topology, reliability, fault-tolerance, re-configurability, flexibility, and context-awareness. Unlike wired networks such as Ethernet, increased latency, throughput, and bandwidth utilization may be secondary design goals for the MAC protocol in the WSNs [5–9]. Finally, because sensor nodes in a WSN environment collaborate with each other to perform a common task, fairness among nodes is a design criterion that requires little attention [7,10].

3.2.1 Energy Conservation

There are a number of transceiver activities that may lead to excessive or unneeded energy consumption. These include collisions of transmissions in multiple sender environments, control packet overhead, adaptation to network changes when nodes join, leave, or move [6]. Each of these activities demand idle listening, where a node waits and listens for activity on the communication channel. Therefore, the main sources of energy wastage by radio transceiver are due to idle listening, collisions, overhearing, and control packet overhead [11]. A collision occurs when two nodes try to transmit packets at the same time or a transmitted packet is corrupted due to interference. The packet must be discarded and retransmitted. The retransmissions increase energy consumption. Overhearing occurs when a node receives packets that are destined to other nodes. Control packet overhead includes sync packet, RTS-CTS. Idle listening is waiting to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, the sensor node will be in idle state for most of the time. The study by Reason and Rabaey [12] shows that idle listening accounts for approximately 90% of the energy wasted in sensor networks. Duty-cycling is largely seen as the means by which idle listening periods may be reduced, and this is the basis of many sensor network MAC protocols. Among the earliest examples designed to address these challenges are fixed duty cycling in sensor MAC (S-MAC) [6,11], and wake-up slots in time-division multiple access-wakeup (TDMA-W) [13]), discussed later in the chapter.

3.2.2 Mobility and Adaptability to Network Changes

Most of the MAC protocols are designed for relatively static sensor nodes. Mobility and network changes are ignored. Mobility is a natural requirement in the devices used in pervasive environments. The architecture of the WSNs used in pervasive computing applications must handle the physical mobility of sensor nodes for better coverage. Adaptability to network topology changes, node join (add), and node failures (drop) must be handled gracefully.

3.2.3 Scalability

Pervasive computing environments will likely face a proliferation of users, applications, networked devices, and their interactions on a scale never experienced before [3]. Handling large-scale deployment of sensor nodes over a wide area is crucial.

3.2.4 Reliability/Fault-Tolerance

A number of WSN applications require reliable data delivery while delays can be tolerated. Wireless sensor nodes often operate in inhospitable environments are prone to failures. The ability of a sensor network to operate successfully in the wake of failures (i.e., fault tolerance) will make the goal of data reliability achievable.

3.2.5 Reconfigurability/Flexibility

Versatile and flexible protocols such as Berkeley MAC (B-MAC) [14], where by tuning different parameters can provide application-specific trade-offs. A reusable suite of tools that offer compatibility with a wide variety of sensor and system technologies is needed.

3.2.6 Context Awareness

Context-aware applications require MAC protocols that can dynamically adapt to context and should be able to translate context information into suitable MAC parameters. For example, context-aware body sensor networks [15] are critical for the development of pervasive healthcare monitoring applications.

3.2.7 Categories of MAC Protocols

Despite the differences between MAC protocols for WSNs, they can nevertheless be grouped into four different categories as shown in Figure 3.1. It must be noted that, historically, MAC protocols were grouped into two broad categories: contention-based protocols and schedule-based protocols. We have extended this categorization to include new families of hybrid and consensus-based protocols. This categorization helps in identifying the key attributes of each group and provides a critical review of their usefulness in addressing the design constraints described above.

3.2.7.1 Contention-Based MAC Protocols

Most contention-based protocols are randomized techniques that allow asynchronous access to the wireless medium via collision avoidance mechanisms. Collisions (sometimes described as contentions) cause corruption of data frames. In order to recover from corrupted frames, nodes must retransmit. Carrier Sense Multiple Access with Detection (CSMA/CD) is the classical contention-based MAC protocol in wired networks generally referred to as Ethernet. In CSMA, a node, upon detecting a collision, backs off for a random period of time before retransmission. Other contention-based protocols substitute for the detect-and-retransmit approach. CSMA suffers from the

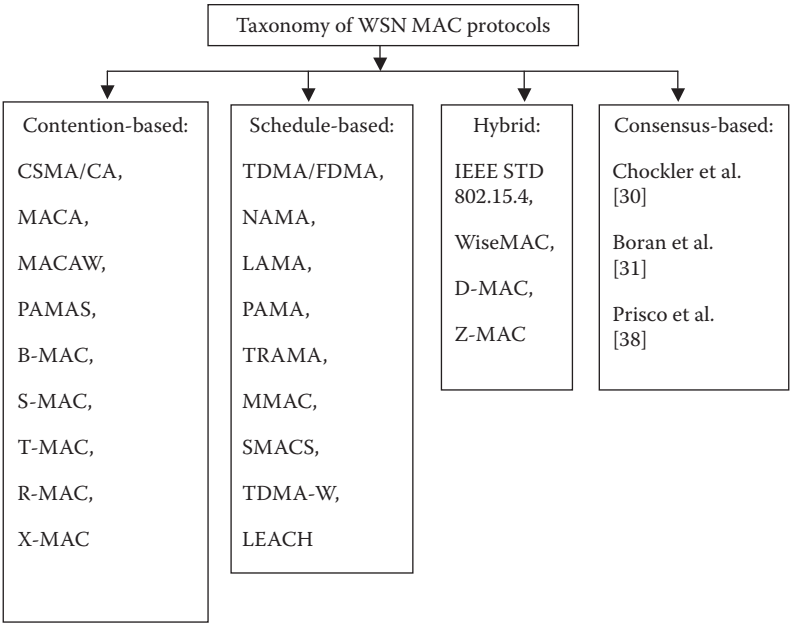

FIGURE 3.1 Taxonomy of MAC layer protocols.

hidden terminal problem [5], and the CSMA/Collision Avoidance (CSMA/CA) was developed to solve the hidden terminal problem. For example, CSMA/CA prevents collisions before they happen by listening before talking. Before transmission of the data packets, CSMA/CA first senses the channel by exchanging control packets' request to send (RTS) and clear to send (CTS). In case of collision, the CSMA/CA protocol requires the sender to back-off, where the back-off period is determined by a uniform distribution. The process is repeated once the back-off period expires. Once obtained, a node occupies the channel until it completes its transmission. Contention-based MAC can be further sub-divided into two groups: contention-based synchronous MAC protocols, such as S-MAC [11] and timeout MAC (T-MAC) [16], and asynchronous MAC protocols, such as B-MAC [17], X-MAC [18], and receiver-initiated MAC (RI-MAC) [19].

These protocols can easily adapt to network topology changes that occur when nodes join or leave. This family of protocols is simple to implement and requires less coordination between nodes. In addition, there is no need for a central coordinating node to exist in the network. However, this simplicity comes at a cost: contention-based protocols suffer from lower throughput and higher delay when compared to other protocols. They may also use more energy as compared to other protocols due to longer idle listening periods, overhearing, and collision recovery.

We now discuss the more popular contention-based protocols used in WSNs (such as the S-MAC and the T-MAC). These protocols regulate idle listening, over-hearing, and data transmission phases with sleep and wake up periods in such a way so as to reduce the overall energy consumption and improve throughput and delay.

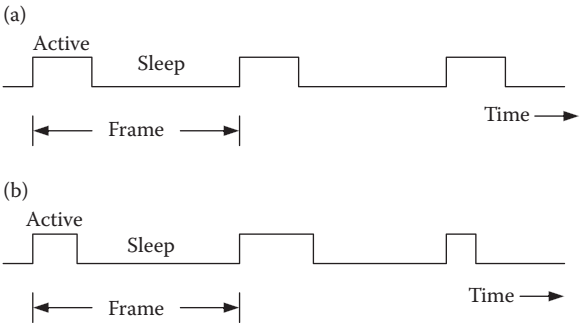

FIGURE 3.2 S-MAC vs. T-MAC frame. (a) S-MAC frame with constant active (sleep) time. (b) T-MAC frame with dynamic active time.

3.2.7.1.1 Sensor MAC

Sensor MAC (S-MAC) is one of the first contention-based MAC protocol [6,7,11] designed for multi-hop WSNs. In order to improve energy efficiency, S-MAC implements a periodic listen (or active) and sleep cycle. This scheme divides time frame into a short constant listen period, followed by a long sleep period as shown in Figure 3.2a. Typically the duty cycle (i.e., the ratio between the listen time and the full listen/sleep cycle time) consists of only 1–10% [11]. All sensor nodes choose an active/sleep schedule for themselves such that nodes listen, exchange data, and control packets (i.e., SYNC, RTS/CTS, ACK) during active periods, and turn off their radios during the sleep periods. Nodes share their schedules with neighbors so that they may communicate. Despite this sharing, it is not always possible to synchronize schedules between neighbors. One reason for this is clock shift. Therefore, to provide clock synchronization, each node periodically broadcast its schedule in a SYNC packet.

In order to avoid collisions, S-MAC uses RTS/CTS handshake procedure similar to the IEEE 802.11 distributed coordination function operation [20]. Another important feature of the S-MAC is the use of a message-passing mechanism to reduce the contention latency of long messages. Long messages are fragmented and transmitted in bursts. Only one RTS/CTS pair is used to reserve the medium for all the fragments, while a separate acknowledgement (ACK) is used per fragment. This scheme will allow only a particular fragment to be retransmitted if required, thus avoiding the cost of retransmission of the full message. S-MAC also allows the formation of virtual clusters of nodes that have common sleep schedules.

We may summarize the S-MAC family of protocols according to the following benefits and drawbacks.

Pros:

- S-MAC tries to reduce energy wastage due to collision by using RTS/CTS.
- The energy waste caused by idle listening is reduced by sleep schedules.
- Its implementation is simple relative to other protocols.

Cons:

- Broadcast data packets do not use RTS/CTS which increases collision probability.
- Due to predefined fixed duty ratios, the efficiency of the algorithm under variable traffic load degrades.
- The fixed duty cycle is the biggest drawback of S-MAC. If the listening period is long, too much energy would be wasted by idle listening as the sensor will be awake even if there is no traffic. On the other hand, if the listening period is short, contention probability is high and energy would be wasted by retransmission efforts.
- With increase in the WSN density, maintaining neighbor's schedules is an additional overhead.

3.2.7.1.2 Timeout MAC

T-MAC protocol proposed by [16] improves the S-MAC protocol by using a novel adaptive listen/sleep duty cycle, demonstrated using Figure 3.2b. T-MAC also uses fixed frames similar to S-MAC; however, T-MAC allows for a variable duration of the active listening period. In T-MAC, nodes dynamically time-out and end the active period, thereby turning off their radio if no activation event has occurred. The duration, T_a, of the active period is given by Equation 3.1. An event may be reception of data packet or start of listen/sleep frame time. For variable traffic, the energy consumption in T-MAC is less than the S-MAC because of the duty cycle adjustment through fine-grained time-outs. It is also better in handling traffic fluctuations. However, the trade-off for increased energy conservation in T-MAC when compared to S-MAC is latency.

$$T_a = 1.5(C + R + T) \tag{3.1}$$

where C is the length of the contention interval, R is the length of the RTS packet transmission time, T is the turn-around time (i.e., the time between the end of the RTS packet and the beginning of the CTS packet).

3.2.7.1.3 Berkeley MAC

Polastre et al. [14] proposed a CSMA-based versatile MAC protocol, called B-MAC, to meet the challenges of a wide range of WSN applications. Instead of a single monolithic protocol for general purpose workloads, B-MAC is an adaptive, reconfigurable light weight protocol that implements only the core functionality of an MAC layer. It employs low power listening (LPL) [21] to reduce duty cycle and to minimize idle listening. It provides well-defined flexible interfaces that allow different applications to implement their own MAC. B-MAC employs clear channel assessment technique and back-offs for channel arbitration for improving the channel utilization, and link layer ACKs for reliability. Another important feature of the B-MAC is that it uses asynchronous duty-cycle approach. That is, B-MAC does require neighboring nodes to synchronize their active-sleep schedules. Each node can independently operate on its own duty-cycle schedule.

LPL scheme introduced by Hill and Culler [21] is an asynchronous wake-up of sleeping radios for CSMA-based protocols. LPL is similar to preamble sampling

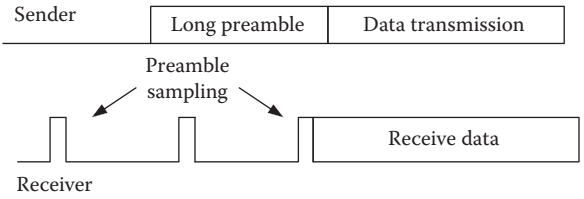

FIGURE 3.3 Low power listening in B-MAC.

scheme presented by El-Hoiydi [22] for ALOHA. In both of these schemes, prior to data transmission, a sender node transmits a long preamble (i.e., a wake-up signal) that lasts longer than the sleep period of receiver nodes and the receiver periodically wakes up according to its schedule and senses the channel as shown in Figure 3.3. If a preamble is detected, the node will remain awake to receive data, otherwise the node will return to sleep mode. The extended preamble of the LPL scheme ensures that the receiver will wake-up at least once during the preamble. The preamble length is provided as a parameter to the upper layer and it depends on the application developer.

B-MAC's flexibility results in better packet delivery rates, throughput, latency, and energy consumption as compared to S-MAC. That is, the application can adjust the sleep schedule to adapt to changing traffic loads. This results in higher performance from B-MAC because with the application-specific knowledge one can turn on and off these low-level features to minimize the overhead of the MAC protocol.

The authors in [14] also present an analytical model for monitoring application in order to calculate and set B-MAC parameters that optimize power consumption. Experiments have shown that B-MAC has better performance in terms of latency, throughput, and energy consumption as compared to S-MAC.

Pros:

- B-MAC provides on the fly reconfiguration and provides bidirectional interfaces by which the user can choose optimal parameters for a given application.
- The sender and receiver can be completely decoupled in their duty cycles. Avoids synchronization overhead.
- No need to share schedules with neighbors.
- B-MAC is currently the default MAC protocol in TinyOS.

Cons:

- Long preamble consumes a significant amount of channel time. Therefore, it is not efficient in case of contending traffic flows [18].
- Lop power listening approach suffer from overhearing at receiver nodes that are not target of a given data packet.
- As a consequence, B-MAC does not suit for highly varying traffic rates.
- The flexibility of B-MAC means the application programmer needs to know how to optimize the parameter. Therefore, it makes the application to be more complicated.

3.2.7.1.4 X-MAC

X-MAC proposed by Buettner et al. [17] is an asynchronous duty-cycled energy-efficient MAC protocol for WSNs that employs a series of short preamble prior to data transmission. X-MAC embeds the ID of target nodes into preambles and adds gaps between preambles to wait for ACK from the target node. On the one hand, by using a series of short preamble instead of a single long preamble X-MAC avoids the overhearing problem and on the other hand, it retains the advantages of LPL scheme. Similarly, by embedding the target address in each short preamble, X-MAC allows irrelevant nodes to go to sleep immediately and if the node is an intended recipient, it sends early ACK *to* the sender and remains awake for the subsequent data. When the sender gets the ACK, it stops preamble transmission and starts transmitting the DATA frame. In this way, X-MAC saves energy by avoiding overhearing while reducing latency almost by half on average [18].

Authors in [17] have shown through implementation that X-MAC. results in significant savings in terms of both energy and latency. X-MAC also uses traffic-specific adaptive duty cycle algorithm to accommodate varying traffic loads in the network.

Pros:

- Because of asynchronous duty-cycled approach, it avoids synchronization overhead.
- No need to share schedule information with neighbors.
- Active period can be significantly shorter than that of synchronized methods.
- Avoids overhearing due to short preamble. Energy efficient for under light traffic loads.

Cons:

- Series of short preamble before the data packet consume a significant amount of channel time. Therefore, it is not efficient in the case of contending traffic flows.

3.2.7.1.5 Receiver-Initiated MAC

RI-MAC proposed by Sun et al. [18] is an energy-efficient and high-performance asynchronous duty cycle MAC protocol for a wide range of traffic loads. RI-MAC separates itself from other contention-based schemes by implementing receiver-initiated data transmission.

In RI-MAC, each node wakes up based on its own schedule and broadcasts a beacon to notify neighbors it is awake. That is, the receiver sends the invitation beacon and stays awake for some time to wait for packets from a sender before going back to sleep if a sender wants to send a data frame, it jumps to the listening state and waits for receiver's beacon. Upon receiving the beacon, it begins data transmission immediately.

In LPL-based asynchronous protocols such as B-MAC and X-MAC, the preamble transmission by a sender may occupy the wireless channel for much longer than the data transmission. In the worst case, this could prevent all neighboring nodes with pending data from transmitting their data. In WSN paradigms, where multiple sensor nodes detect the same event and send their data to the sink node, all nodes could experience significant delay. In contrast, in RI-MAC, instead of a long preamble only a short beacon is transmitted before the data transmission. This minimizes unnecessary channel occupancy. In RI-MAC, collisions are detected by the receiver, if a collision occurs; the

receiver sends a new beacon specifying a contention window. In the case of consecutive collisions, the length of contention window is increased.

Sun et al. [18] have demonstrated the performance evaluation of the RI-MAC protocol through *ns-2* simulation and through measurements of an RI-MAC implementation in TinyOS. RI-MAC achieves higher throughput, packet delivery ratio, and power efficiency under a wide range of traffic loads as compared to the prior asynchronous duty cycling approaches such as X-MAC.

Pros:

- High-power efficiency under a wide range of traffic loads.
- Asynchronous duty-cycled means sender and receiver schedules are decoupled.
- More suitable in pull-mode. That is, where the need to receive data dominates over the need to transmit.

Cons:

- Idle listening may be encountered because the sender usually has to be awake for a long time before the receiver sends beacon.

3.2.7.2 Schedule-Based MAC Protocols

Schedule-based MAC protocols such as TDMA, frequency division multiple access, and code-division multiple access are widely used in traditional wireless networks. Schedule-based protocols avoid collisions by dividing the wireless channel into subchannels (slots) based on time, frequency, or orthogonal codes. The slots are further organized into time frames. In every frame, each sensor node is allocated a slot to occupy the medium and conduct its operation. This provides a mechanism for two nodes to communicate with each other in a uniquely selected slot. These protocols can use either centralized or distributed coordination schemes so that the access to a particular channel is limited to only one sender node at a given slot. Thus, scheduled protocols tend to avoid collisions and idle listening by scheduling the data transmission in advance, but waste energy in other ways. These protocols require strict time synchronizations and also need to exchange their schedules with neighbors periodically [10,13,22].

While these protocols avoid collision, are more energy efficient than contention-based protocols, and offer higher throughputs, they tend to be poorly suited in environments that are un-predictable and dynamic.

3.2.7.2.1 Time Division Multiple Access

In TDMA protocols, time is divided into slotted frames and each node is assigned a particular time slot per frame for transmission. These protocols are collision and retransmission free, because each node has a dedicated time slot in which only that particular node transmits. These protocols save energy by keeping their radio transceivers off most of the time and turning them on only for short period at their scheduled time slot when they are either transmitting or receiving packets. Hence, in TDMA, sensor nodes are also able to eliminate the idle listening problem which is a significant source of energy wastage in the case of contention-based MAC protocols.

TDMA requires centralized or cluster-based approach. Many variations of the TDMA protocol have been reported [13,23]. TDMA protocols often require 2-hop neighbor information to establish schedules. For example, low-energy adaptive clustering hierarchy (LEACH) proposed by Heinzelman et al. [24] is a well-known routing protocol for static WSNs. LEACH organizes nodes into cluster hierarchies, and applies TDMA within each cluster. The major drawbacks to naïve TDMA approaches such as LEACH are a fixed throughput that is impossible to increase beyond utilization of all available slots. Network topology is also relatively fixed since a given network topology is used to establish a collision-free arrangement and tight synchronization among nodes. Both knowledge of topology and strict synchronization require large overheads and/or expensive hardware and hence render TDMA solutions less attractive in large-scale sensor deployment. It means scalability is a big issue.

3.2.7.2.2 TDMA-Wakeup

TDMA-W proposed by Chen and Khokhar [13] is an energy-efficient distributed schedule-based MAC protocol for WSNs environment which introduced the concept of a wakeup slot. The wireless channel is organized as a set of TDMA-W frames of T_{frame} seconds. Each node is assigned two slots per T_{frame}; one slot for data transmission called a send slot (*s-slot*) and another slot for wakeup session (*w-slot*). A node always listens during its *w-slot* and transmits data, if any, in the s-slot. A node utilizes its assigned slot only when it is sending or receiving information; otherwise, its receiver and transmitter are turned off. The wake-up packets are used to activate a sleeping node in order to accelerate their response.

The TDMA-W protocol operates in two phases; a self-organization (setup) phase and a channel access phase. In the network setup (self-organization) phase as described in [13], each node repeatedly executes a 7-step self-organization scheme that will uniquely determine *s-slot* number for all the nodes and their corresponding neighbors. During the setup phase, nodes broadcast their node-ids, *s-lot* number one-hop neighbors'-ids and neighbor s-lot numbers to each other. At the end of the setup phase, each node also selects any unused *s-slot* beyond its two-hop neighbors as its *w-slots* and broadcast it. The *w-slot* can be shared with neighbors. The *w-slot* may not be a unique, whereas *s-lots* are unique in two-hop range.

In the channel access phase, each sensor node maintains a pair of counters per neighbor (an outgoing packet counter and an incoming packet counter). These counters are used to determine the status of a particular neighbor, whether the corresponding neighbor is in active or sleep mode. Whenever a source node has a data packet to be delivered to a given destination node, it determines the status of the destination node and if it is active, the data packet will be sent out during the *s-slot*. If the destination node is in sleep mode, the source node sends a control packet (including the source ID) in the wake-up slot of the particular destination node prior to the data packet. The intended destination node will turn on its receiver in the *s-slot* corresponding to the source node ID. This effectively creates a pairing between source and destination within a slot.

Nodes in TDMA-W send control packets during scheduled data slots, and that is how the neighbors' allocation information is disseminated. Based on its neighbors' allocation information, a node is able to select a collision-free transmission slot which will be used

for subsequent data and control packet transmissions. It handles both the synchronized and non-synchronized networks.

Pros:

- Energy is saved in TDMA-W using transmission slots and avoiding contention.
- Special wake-up packets may be to expedite a sleeping receiver's response.

Cons:

- Absolute slot identifications must be exchanged via control packets.
- TDMA-W is dependent on strict global framing and time synchronization.

It should be noted that the benefits in energy consumption involve predictability rather than savings. This is because in each frame a node must remain active during its wake-up slot even if it is not transmitting or receiving during that frame. This implies that a node must expend one full data slot worth of energy in each frame, irrespective of its communication activities. This drawback is addressed by the traffic adaptive medium access (TRAMA) protocol, discussed in the following section.

3.2.7.2.3 Traffic Adaptive Medium Access

TRAMA proposed by Rajendran et al. [23] is an energy-efficient, collision-free traffic-adaptive medium access protocol for static multi-hop WSNs. It is designed to be a dynamic time-division scheme in which the size of both the frames and the allocation of slots within the frames are determined dynamically based on the traffic flow.

A distributed election scheme uses information about the traffic at each node to determine which node can transmit in each time slot. TRAMA does not assign time slots to those nodes that have no traffic to send and, therefore, nodes can keep on sleeping instead of waking-up. This reduces idle listening and overhearing to zero. Energy consumption is reduced by ensuring collision-free transmissions and by providing facilities that allow nodes to remain in a low-power mode whenever they are not transmitting or receiving.

TRAMA consists of three major components: (a) the neighbor protocol that collects information about neighboring nodes, (b) the schedule exchange protocol that is used to exchange two-hop neighbor information and their schedules, and (c) the adaptive election algorithm (AEA) that determines the sender and receiver nodes for the current time slot using the neighborhood as well as schedule information.

In TRAMA, time frame is divided into a contention-free *scheduled access* period and a contention-based *random access* period, as shown in Figure 3.4a adopted from [10]. The overall frame time is fixed. Data transmission is performed in the scheduled *access* slot and the neighbor information is collected in the *random access* slot. Each node uses an AEA to determine the slot which can be used to transmit packets. The schedule access period is then used to announce the schedule and perform the actual data transmission.

The main advantage of TRAMA over S-MAC [6,11] is improvement in channel utilization. The tradeoff is longer delay and higher energy consumption in comparison with S-MAC protocol.

The main issue with TRAMA is its complexity and the assumption that nodes are synchronized network-wide.

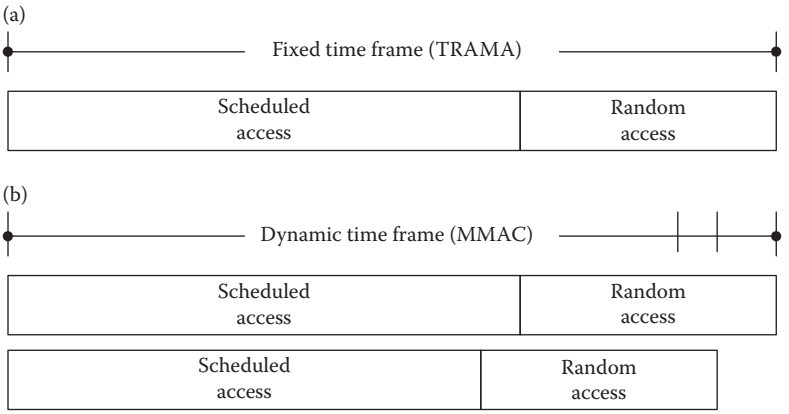

FIGURE 3.4 Fixed vs. dynamic frame. (a) Fixed time frame as used in TRAMA. (b) Dynamic Time frame used in MMAC. (Adapted from Ali, M. and Uzmi, Z., *Int J Sensor Netw* 2006; 1(3/4): 134–142.)

3.2.7.2.4 Mobility Adaptive MAC

MMAC proposed by Ali et al. [10] is a collision-free distributed MAC protocol for mobile WSNs that are a variant of TRAMA [23]. MMAC has two different access periods similar to TRAMA, a random access period for schedule reservation and a scheduled access period for transmitting data packets, as shown in Figure 3.4b.

MMAC is motivated by the idea that, in mobile environments, the use of a fixed-length static frame causes performance degradation because mobile nodes may be forced to wait for unacceptable lengths of times before being able to communicate, thereby rendering schedules obsolete. To alleviate this problem, MMAC uses a dynamic frame time that is inversely proportional to the level of mobility. This is possible in MMAC only if nodes are aware of their locations so that mobility patterns of nodes may be predicted. The time frame length is dynamically adjusted according to the mobility patterns, thus making it suitable for handling both the low mobility (e.g., topology changes, node joins, and node failures) and the high mobility (e.g., concurrent node joins and failures, and physical mobility of nodes).

MMAC uses a probabilistic autoregressive model to predict the mobility of two-hop neighbors. It then adjusts the time frame and random access time according to the mobility of those nodes. The result is a collision-free schedule based on estimates of traffic flow, mobility patterns.

Pros:

- A time-division scheme that allows for node mobility.

Cons:

- Location information is assumed to exist.
- Node localization may consume a lot of energy.
- MMAC is the highly complex and computation intensive scheduling algorithm for determining the transmitter for each slot in a frame.

- In case of high mobility, packet schedule causes a higher control packet overhead, and results in a higher duty cycle.

3.2.7.3 Hybrid MAC Protocol

We have seen that both collision detection and time-division schemes have their drawbacks. Hybrids approaches incorporate the advantages of both contention-based and schedule-based schemes, but tend to avoid their disadvantages. IEEE 802.15.4 [25], wireless sensor MAC (WiseMAC) [26], and Z-MAC [27] are examples of hybrid MAC protocols.

3.2.7.3.1 Wireless Sensor MAC

The WiseMAC protocol [26,28] builds on spatial TDMA and CSMA first proposed in [29], using a preamble sampling protocol. In the original work, nodes have two communication channels such as TDMA, which is used for accessing data channel, and CSMA, which is used for accessing control channel. WiseMAC reduces the number of data channels back down to one.

To reduce the idle listening, it uses non-persistent CSMA with preamble sampling technique.

In WiseMAC, the sampling schedules of the direct neighbors are shared by piggybacking them into ACK frames. A node receiving a data frame includes in the following ACK frame the remaining time until its next sampling time. The length of the wakeup preamble is minimized by exploiting the knowledge of sampling of the schedules of its direct neighbors.

With this knowledge, the sender of the next data frame estimates when the target receiver will wake up next, and starts transmitting its preamble just before then. All nodes in the network perform channel sampling periodically with independent sampling offsets.

This scheme allows the reduction in the energy consumption caused by unnecessarily long preambles by the sender as well as the receive power consumption. It also brings a drastic reduction in the energy wasted due to overhearing.

WiseMAC requires no set-up signaling, no network-wide time synchronization, and is adaptive to the traffic load. It provides ultra-low average power consumption in low traffic conditions and high energy efficiency in high traffic conditions.

While WiseMAC solves many of the problems associated with low power communications, it does not provide a mechanism by which nodes can adapt to changing traffic patterns.

3.2.7.3.2 Zebra-MAC

Z-MAC [27] is a hybrid CSMA/TDMA MAC protocol for WSNs.

As we have learned, CSMA is ideal for low traffic loads while it suffers from traffic contention under high traffic. On the other hand, TDMA avoids contentions and provides high channel utilizations under high traffic. In TDMA, nodes can only transmit during scheduled slots and under low contentions slots are wasted and channel utilization is reduced. The Z-MAC protocol combines the advantages of both TDMA and CSMA while offsetting their weaknesses. This is achieved by utilizing CSMA as a baseline proto-

col while switching to TDMA under high traffic contentions. Therefore, Z-MAC achieves high channel utilization and low-latency under low contention as in CSMA and it achieves high channel utilization under high contention and reduces collision among two-hop neighbors at a low cost similar to TDMA. Therefore, Z-MAC can dynamically adjust the behavior of MAC between CSMA and TDMA depending on the traffic contention and provide high performance at both ends of the utilization spectrum.

Authors [27] have demonstrated that Z-MAC performs better than B-MAC under medium to high contentions. However, its performance is inferior to B-MAC under low contention.

3.2.7.4 Consensus-Based MAC Protocol

A relatively unexplored approach rests in the family of consensus-based MAC protocols. These protocols are primarily concerned with achieving agreement on a particular value or state within the wireless nodes. This feature also provides a level of fault-tolerance absent in the other MAC approaches. This may be highly desirable, given that some WSNs can be categorized by rugged terrain, high node failure rates, battery depletion, or hibernation where an agreed upon state by the nodes in the network may get lost. A loss of state can result in temporary disruption of connectivity in the network. Hence, fault tolerance is necessary to maintain a consistent state that all active nodes in the network can act on. Studies in [30,31] show that even sophisticated MAC protocols can result in excessive message loss of up to 50%.

There has been extensive research done on fault tolerance in wired networks with reliable or eventually reliable communications channels. The challenges in wireless networks are more complex as messages can be lost due to collision, node failures, and signal fading. This is only accentuated by environments which are more dynamic, allowing nodes to enter and exit from the sensor community. Developing a consensus algorithm that addresses all of these concerns is non-trivial. In recent years, several papers have been published to address the issue of consensus in WSNs. We will briefly discuss a couple of algorithms on fault tolerance and consensus in WSNs.

Most algorithms [32–34] that work on consensus define the agreement over a set of processes, where each process may propose a value. A leader then decides which value to accept and asks for approval of that value from the processes. Once the leader receives the acceptance from the processes, it then declares the value to be the final value for that round.

Different algorithms stipulate slightly different conditions, but all of them in general follow the process defined above. In Chockler et al. [32], for example, the number of nodes in the system is not known *a priori*; however, they do assume a synchronous system with bounded delays and a message loss detection scheme. In Borran [33], the consensus algorithm based on the Paxos algorithm [35,36] requires that the number of nodes in the system are known *a priori*, so that when the leader sends out the proposed value, and receives responses back from the other sensors in the network, it knows when to stop and move on to the next phase. The leader moves on to the next stage when it receives responses from a majority of the active nodes. The algorithm described in [37], besides using the three basic steps described above, also requires a failure detection scheme.

In WSN, because of the conditions that these sensors operate in, fault tolerance and consensus in the system are vital to improve the efficiency of the system.

3.3 Conclusions and Future Directions

In this chapter, we have discussed the issues and challenges associated with MAC protocols for WSNs.

Sensors have always been a part of our lives. However, with advancements in process technology, it is possible to miniaturize sensor nodes to the point where they can be embedded inconspicuously in most things we use. The inevitable proliferation of sensors is enabling pervasive communication and computing, where sensor nodes collect data and process it before relaying it to the host node where decisions can be made on how and whether to act on the information received.

Sensor-based pervasive computing also requires sensor networks to be energy-efficient, scalable, fault tolerant, and cost-effective. WSNs are composed of tiny sensors which are connected to one another, where each node is constraint by energy, storage, and processing capacity, despite the advances made in process technology and VLSI design. WSNs impact all layers of the OSI model and in particular the MAC layer. Several different categories of MAC protocols have been suggested in literature addressing different problems and applications. These MAC protocols can be categorized into four different families. These are: Contention; Scheduled; Hybrid; and Consensus based protocols.

Contention-based algorithms are simple to implement; they require very little coordination between different nodes in the network. The existence of a central coordinating node is not a requirement. This family of protocols is also robust to failures and changes in the network topology. But this flexibility comes at a cost: these protocols may use higher energy; have lower throughput; and have higher transmission delay. Contention-based protocols are well suited to environments which are dynamic; that is, they are constantly changing due to either node failures or mobility and are characterized by low message activity.

Schedule-based protocols, on the other hand, offer higher network efficiency, and lower delay. However, these protocols require a central node to arbitrate the schedule and assign time slots. Most of the protocols in this category are based on the TDMA protocol. The schedule-based protocols are well suited for sensor environments that require lower message delay, have high message traffic, and have a stable topology. They are also much more complicated to use, in part because of the added complexity of time-synchronization.

Hybrid protocols, as the name suggests, are a mixture of both schedule- and contention-based protocols. Protocols like the one suggested in [26] use the contention-based protocol for downstream information transfer and the schedule-based protocols for upstream communication. Having two separate protocols may be reasonable when they are implemented in software, but implementing two different MACs in hardware adds to the cost of the MAC.

Finally, the fourth category that we discussed belongs to consensus-based protocols. Consensus-based protocols are becoming important as pervasiveness of sensors in our

environment and daily lives increases. Wireless sensor nodes often operate in inhospitable environments, are prone to failure, and therefore need a protocol that either implements in the MAC layer or utilizes the functions of the underlying MAC protocol to achieve consensus. Consensus is a growing requirement in WSN networks when all nodes in the network need to have a consistent view of the state of the network.

As WSN evolve, MAC protocols will also evolve to accommodate the needs of the applications and the environment. Since the applications and the environment that sensors support and operate in are varied, it is difficult to foresee a single dominant category. Consensus-based protocols will gain more traction as fault tolerance requirements increase.

A simple and lightweight modular MAC that can dynamically adaptable to the context, handles mobility, fault-tolerant, flexible, and easily extensible to accommodate new services will be a step forward toward achieving the vision of ubiquitous pervasive computing.

References

1. Weiser, M. The Computer for the 21st Century, *Scientific American*, pp. 94–104, September 1991.
2. Soylu, A., De Causmaecker, P., and Desmet, P. Context and adaptivity in context-aware pervasive computing environments: links with software engineering and ontological engineering. *J Softw*, 2009; 4(9): 992–1013.
3. Saha, D. and Mukherjee, A. Pervasive Computing—a paradigm for the 21st century. *IEEE Comput* 2003; 36(3): 25–31.
4. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. A survey on sensor networks. *IEEE Commun Mag* 2002; 40(8): 102–114.
5. Ye, W. and Heidemann, J. Medium Access Control in Wireless Sensor Networks, USC/ISI Technical Report, Tech. Rep. ISI-TR-580, 2003.
6. Ye, W., Heidemann, J., and Estrin, D. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans Netw* 2004; 12(3): 493–506.
7. Demirkol, I., Ersoy, C., and Alagoz, F. MAC protocols for wireless sensor networks: a survey. *IEEE Commun Mag* 2006; 44(4): 115–121.
8. Bachir, A., Dohler, M., Watteyne, T., and Leung, K. MAC essentials for wireless sensor networks. *IEEE Commun Surveys Tutorials* 2010; 12(2): 222–248.
9. Ali, M., Saif, U., Dunkels, A., Voigt, T., Römer, K., Langendoen, K., Polastre, J., and Zartash, U. Medium access control issues in sensor networks. *ACM SIGCOMM Comput Commun Rev* 2006; 36(2): 33–36.
10. Ali, M. and Uzmi, Z. Medium access control with mobility-adaptive mechanisms for wireless sensor networks Source. *Int J Sensor Netw* 2006; 1(3/4): 134–142.
11. Ye, W., Heidemann, J., and Estrin, D. An energy-efficient MAC protocol for wireless sensor networks. *IEEE Infocom*, 2002; 2: 1567–1576.
12. Reason, J. and Rabaey, J.M. A study of energy consumption and reliability in a multi-hop sensor network. *ACM SIGMOBILE Mobile Comput Commun Rev* 2004; 8(1): 84–97.

13. Chen, Z. and Khokhar, A. Self-organization and energy-efficient TDMA MAC protocol by wake up for wireless sensor networks, in *Proceedings of the first IEEE Conferences Sensor and Ad Hoc Communication and Networks (SECON '04)*, pp. 335–341, October 2004.
14. Polastre, J., Hill, J., and Culler, D. Versatile low power media access for wireless sensor networks, in *Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, MD, USA, 3–5 November 2004, pp. 95–107.
15. Chiti, F., Fantacci, R., Archetti, F., Messina, E., and Toscani, D. An integrated communications framework for context aware continuous monitoring with body sensor networks. *IEEE J Select Areas Commun* 2009; 27(4): 379–386.
16. van Dam, T., Langendoen, K. An adaptive energy efficient MAC protocol for wireless networks, in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems,* Los Angeles, CA, November 2003.
17. Buettner, M., Yee, G., Anderson, E., and Han, R. X-MAC: a short preamble MAC protocol for duty cycled wireless sensor networks, in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (Senses 2006)*, pp. 307–320, 2006.
18. Sun, Y., Gurewitz, O., and Johnson, D.B. RI-MAC: a receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks, in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*, November, 2008.
19. IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
20. Hill, J.L. and Culler, D.E. Mica: a wireless platform for deeply embedded networks, *IEEE Micro* 2002; 22: 12–24.
21. El-Hoiydi, A. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks, in *Proceedings of IEEE International Conference on Communications*, New York, NY, 2002, pp. 3418–3423.
22. Yadav, R. et al. A survey of MAC protocols for wireless sensor networks. *Ubiquit Comput Comput J* 2009; 4(3): 827–833.
23. Rajendran, V., Obraczka, K., and Garcia-Luna-Aceves, J.J. Energy efficient, collision-free medium access control for wireless sensor networks, in *International Conference on Embedded Networked Sensor Systems*, November 2003, pp. 181–192.
24. Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks, in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, January 2000, pp. 1–10.
25. IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2006.
26. El-Hoiydi, A. and Decotignie, J-D. WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks, in *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Lecture Notes in Computer Science, LNCS 3121*, pp. 18–31, July 2004.
27. Rhee, I., Warrier, A., Aia, M., and Min, J. Z-MAC: a hybrid MAC for wireless sensor networks, in *Proceedings of the 3rd international Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, CA, USA, November, 2005.

28. Enz, C.C., El-Hoiydi, A., Decotignie, J-D., and Peiris, V. WiseNET: an ultralow-power wireless sensor network solution. *IEEE Comput* 2004; 37(8): 62–70.
29. El-Hoiydi, A. Spatial TDMA and CSMA with preamble sampling for low power ad-hoc wireless sensor network, in *Proceedings of ISCC'02, Seventh International Symposium on Computers and Communications*, pp. 685–692, July 2002.
30. Woo, A., Tong, T., and Culler, D. Taming the underlying challenges of multihop routing in sensor networks, in *The First ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, pp. 14–27, 2003.
31. Zhao, J. and Govindan, R. Understanding packet delivery performance in dense wireless sensor networks, in *The First ACM Conference on Embedded Networked Sensor Systems (SENSYS)*, pp. 1–13, 2003.
32. Chockler, G., Demirbas, M., Gilbert, S., Newport, C., and Nolte, T. Consensus and collision detectors in wireless ad hoc networks, in *Proceedings of ACM Symposium on Principles of Distributed Computing (PODC'05)*, Las Vegas, NV, USA, 17–20 July 2005.
33. Borran, F. et al. Extending paxos/last voting with an adequate communication layer for wireless ad hoc networks, in *Symposium on Reliable Distributed Systems*, 2008.
34. Chockler, G., Demirbas, M., Gilbert, S., Lynch, N., Newport, C., and Nolte, T. Consensus and collision detectors in radio networks. *Distrib Comput* 2008; 21(1): 55–84.
35. Lamport, L. The part-time parliament. *ACM Trans Comput Syst* 1998; 16(2): 133–169.
36. Lamport, L. Paxos made simple. *ACM SIGACT News* 2001; 32(4): 18–25.
37. Chandra, T.D., Hadzilacos, V., and Toueg, S. The weakest failure detector for solving consensus. *J ACM* 1996; 43(4): 685–722.
38. Prisco, R.D. et al. Revisiting the PAXOS algorithm. *Theor Comput Sci* 2000; 243(2): 35–91.