Medium Access Control in Wireless Sensor Networks

Kurtis Kredo II [a] Prasant Mohapatra [b,*]

[a] *Electrical and Computer Engineering Department, University of California, Davis*

[b] *Computer Science Department, University of California, Davis*

Abstract

Limited energy, computational, and communication resources complicate protocol design within sensor networks and prevent the application of many techniques used within other networks. Constraints on sensor node cost further restrict which technologies sensor networks may utilize. Despite much attention in recent years, researchers have yet to achieve the goal of long term, independent operation of sensor network deployments under these constraints. One research direction considers the energy expended performing communication functionality. Medium access protocols provide the greatest influence over communication mechanisms and provide the most direct influence over the utilization of the transceiver, the largest energy consumer in most sensor nodes. We present a discussion of medium access control concepts in relation to sensor networks and examine previous wireless medium access control protocols to illustrate how they do not match the requirements and characteristics of sensor networks. We then present several protocols recently proposed in the literature specifically for sensor networks.

Key words: Sensor Network, Medium Access Control, Survey

1 Introduction

Sensor networks [1] consist of small, inexpensive, resource constrained devices that communicate wirelessly in a multihop network. Each device, called a sensor node, collaborates with other devices in the network to perform some

* Corresponding author.
Email addresses: kbkredo@ucdavis.edu (Kurtis Kredo II), prasant@cs.ucdavis.edu (Prasant Mohapatra).

operation for the end user, such as environmental monitoring or target tracking. End users typically desire to deploy sensor nodes randomly throughout the target area in large numbers—hundreds to thousands of sensor nodes; however, some special cases may require the precise deployment of a smaller network. Large sensor network deployments require sensor nodes of marginal cost to keep the overall cost within reasonable bounds, but requiring low cost places a limit on the technologies each sensor node may utilize. Therefore, each sensor node often has a simple processor and limited memory resources. Producing simple, small, and inexpensive devices also limits the energy resources available for sensor node operation. Replacing or renewing energy resources after deployment becomes infeasible or too costly in most cases, so the protocols and applications must make judicious use of the finite energy resources. Some sensor nodes may have the capability to scavenge energy from their environment [2], such as with a solar cell, but adding such capabilities increases the sensor node cost, complicates network deployment, and current commercial devices consume too much energy to survive on ambient energy sources in most environments. Sensor nodes communicate by forming a multihop network to forward messages to the destination, which may collect data for later retrieval by the end user or transfer the data over a dedicated communications link. Sensor nodes avoid direct communication with a distant destination due to the high transmission power requirements for reliably sending messages across the deployment area, which may cover a large geographical area. Despite using multihop communication to reduce energy requirements for communication, the wireless transceiver often consumes the largest amount of energy—per time period of use—within a sensor node and, thus, provides the greatest potential for energy savings. Beyond improving the radio design, an efficient medium access control (MAC) protocol possesses the greatest capability to decrease the energy consumption of the transceiver since it directly controls transceiver operation.

A MAC protocol provides slightly different functionality depending on the network, device capability, and upper layer requirements, but several functions exist in most MAC protocols. In general, a MAC protocol provides [3]:

- Framing – Define the frame format and perform data encapsulation and decapsulation for communication between devices.
- Medium Access – Control which devices participate in communication at any time. Medium access becomes a main function of wireless MAC protocols since broadcasts easily cause data corruption through collisions.
- Reliability – Ensure successful transmission between devices. Most commonly accomplished through acknowledgement (ACK) messages and retransmissions when necessary.
- Flow Control – Prevent frame loss through overloaded recipient buffers.
- Error Control – Use error detection or error correction codes to control the amount of errors present in frames delivered to upper layers.

Most work on sensor network MAC protocols has focused on medium access techniques since the transceiver consumes a significant amount of energy and the MAC protocol has the most direct control over its utilization. Limited energy resources provide the primary constraint for sensor network protocol design, so proposed MAC protocols primarily focus on reducing energy losses related to the wireless medium. Other design constraints, such as fairness, latency, and throughput, appear for specific applications and we present MAC protocols designed with these constraints.

Several aspects of sensor networks differentiate the MAC protocol design from MAC protocols in other networks. First, sensor nodes conserve energy by turning off unneeded hardware because most hardware, even when not active, consumes a non-negligible amount of energy. Thus, each sensor node must somehow coordinate with its neighbor to ensure both devices remain active and participate in communication. Sensor network MAC protocols most often perform—or actively participate in—this functionality so upper layers have only an abstract concept of viable links or topology information. Several techniques, such as schedule-based clustering and separate wakeup communication, exist and we mention them when used in the discussed protocols. Secondly, sensor networks produce traffic that differs from the communication patterns existing in other networks. Environmental monitoring provides a typical sensor network application. Sensor nodes monitoring a particular environmental characteristic periodically send data to a central entity for collection and analysis. These devices individually produce traffic at periodic rates with small payloads. Both the data characteristics, which may exhibit strong periodic generation and high spatial correlation, and the small payload size, which increases the impact of protocol overhead, differentiate sensor networks from other networks. Third, the limited resources available to a sensor node prevent the use of common MAC protocol techniques. Many wireless MAC protocols constantly listen to the wireless channel for activity either for reception or before transmitting. However, a transceiver that constantly senses the channel will quickly deplete the sensor node energy resources and shorten the network lifetime to unacceptable levels.

Resource limitations also complicate the implementation of common functions available in traditional networks. Security functions become difficult to utilize because of the limited memory and computational resources available on the sensor nodes, but many researchers have proposed to implement some functionality at the MAC layer. Security becomes a primary concern for many sensor network applications, such as surveillance and target tracking, where the end users may wish to hide the information collected or even the presence of the sensor network. We do not present security aspects of sensor networks in this paper, but TinySec [4] provides an example of functionality a MAC protocol might include. Synchronization also becomes a problem within sensor networks since the requirement for low cost devices often necessitates the use

of lower precision hardware. Protocols that function based on some form of time synchronization must take into consideration that clock drifts become significant over a sensor network's lifetime.

Scalability poses a further problem for protocol designers. Sensor networks may operate with many hundreds to thousands of devices, so centralized protocols have a distinct disadvantage due to the implicit overhead associated with information distribution. Distributed algorithms, even sub-optimal ones, fit the functionality and platform of sensor networks much better than centralized algorithms [5]. As sensor nodes deplete their energy resources, they become useless and fail to participate in the application operation. Protocols must adapt to these changes without consuming needless overhead. Adaptive MAC protocols may also react to sensor node mobility and the effect of gray areas [6,7] more easily. Finally, sensor network application requirements and characteristics exhibit large variability. Even more than other networks, researchers may have to develop many sensor network protocols that each fit a particular application and deployment. The strict constraints placed on sensor nodes also forces protocols to limit generality to improve some performance metric.

In this paper, we present an introduction to MAC protocols for sensor networks including the constraints faced by protocol designers and a summary of currently proposed MAC protocols. Interested readers can find a briefer survey in a paper by Demirkol et al. [8] and a quantitative comparison of selected protocols in work by Halkes et al. [9]. We focus this paper on wireless sensor networks without mobility, but mention some protocols that address mobility. The sensor nodes we consider have very limited computing, storage, communication, and energy resources. Section 2 provides an introduction to previous MAC protocols proposed for wireless networks and explains why these protocols do not fit the needs and constraints of sensor networks. We discuss previous protocols to illustrate the need for new designs and to introduce many of the techniques used in proposed sensor network MAC protocols. Section 3 discusses the unique attributes of sensor networks that differentiate them from other networks and drive the MAC protocol design. We present several MAC protocol examples from current literature in Section 4, which classifies the MAC protocols into two groups: protocols based on scheduled communication and protocols based on unscheduled communication. We present some areas of possible future research in Section 5 and conclude the paper in Section 6.

2 Wireless MAC Protocols

Wireless networks have received much attention in the past decades from researchers and commercial development. Unfortunately, these advances do

not directly apply to sensor networks because the goals and constraints differ from sensor networks. The largest difference comes from the limited energy resources available within sensor networks, which does not commonly limit traditional wireless network devices.

CSMA and CSMA/CA

Perhaps the simplest form of medium access control involves carrier sense multiple access (CSMA) [10]. Many MAC protocols discussed in this paper use CSMA techniques. Two versions of CSMA exist: non-persistent CSMA and p-persistent CSMA. In non-persistent CSMA, a wireless device that wishes to transmit a message senses the channel to determine if another device has already started transmitting. If the device detects activity on the channel, it performs a backoff operation by waiting before attempting to transmit again. When the device senses no activity on the channel, it transmits the message immediately. p-persistent CSMA differs by having devices continue to sense the channel when they detect activity instead of delaying and checking again later. When the device senses no activity on the channel, either on the first try or at the completion of a previous transmission by another device, it transmits a message with probability p and delays the transmission with probability $1-p$. The channel access times and backoff delays consist of continuous values for unslotted CSMA or discrete time values for slotted CSMA. Traditional CSMA requires devices to remain in the receive state when not transmitting. As mentioned previously, constant channel sensing prevents sensor nodes from using CSMA without modification because the transceiver consumes energy too quickly.

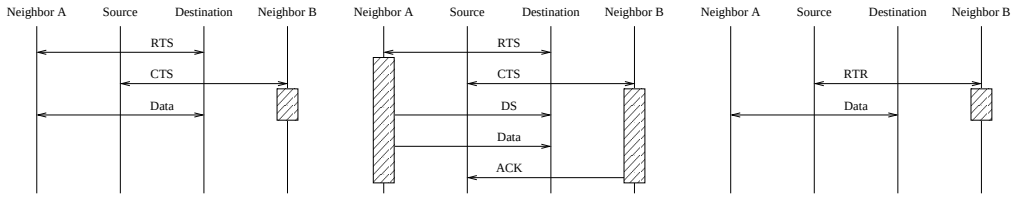
An extended version of CSMA, called CSMA with collision avoidance (CSMA/CA), adds mechanisms to limit the number of messages lost when nearby devices transmit at the same time. Wireless networks attempt to avoid collisions instead of detecting them for two reasons. First, data corruption from a collision occurs at the receiver, so collision detection, commonly used in wired networks, does not indicate that the transmission has failed in a wireless network. Second, collision detection requires transmitting and receiving on the channel at the same time. Adding a full duplex transceiver or a second half duplex transceiver would increase the monetary and energy costs, and complicate the device design. CSMA/CA attempts to avoid collisions by using a control message exchange to reserve the wireless channel before each data message transmission. A device with a message to send first performs the CSMA algorithm to find an appropriate transmission time. Once the CSMA algorithm determines a transmission time, the source device transmits a request to send (RTS) control message to the intended destination. If the destination can receive the pending data message it responds with a clear to send (CTS) control

message. The source device retries the transmission at a later time if it does not receive a CTS within a certain time. A destination device does not respond with a CTS if it can not safely transmit or receive; for example, if the destination detects a transmission, but the source does not detect the transmission, then the destination will defer to the ongoing communication and not send a CTS. After successful reception of a CTS, the source transmits the data message. Neighboring nodes that receive an RTS or CTS message know a data transfer will occur soon and delay attempting any message transmissions until a later time. While CSMA/CA reduces the effect of hidden terminals and associated energy losses in wireless networks, it requires devices to transmit multiple messages for each data message. For sensor networks, where data messages have sizes comparable to control messages, CSMA/CA introduces significant overhead. The benefit of CSMA/CA techniques in sensor networks depends on the traffic conditions, wireless channel characteristics, and network topology, so in some cases it may prove beneficial and in others an unnecessary overhead.

MACA and Variants

The MACA [11] protocol attempts to improve CSMA/CA by eliminating some inefficiencies. First, the author argues that since collisions occur only at the receiver, carrier sensing does not provide an adequate result on which to base channel availability. Therefore, MACA does not use carrier sensing, but instead relies on message timeouts and message responses to detect collisions or channel capture—also called packet sensing. A second modification adds the remaining data exchange length to the RTS and CTS messages so devices that overhear these messages can determine how long to delay before attempting a transmission. Knowing the length of the current transmission allows devices to delay for the optimal time instead of a static, predetermined time, such as the time to transmit the maximum message size. A final addition allows devices that receive an RTS message destined for another device, but do not receive the expected CTS message, to begin a data exchange. In CSMA/CA a device that receives an RTS for another device always remains quiet, but this can lead to exposed terminal inefficiencies. Similar to CSMA, MACA requires devices to constantly sense the wireless channel, so MACA does not satisfy the constraints of sensor networks.

Bharghavan et al. make further modifications to CSMA with the MACAW [12] protocol. Within MACAW destination devices transmit an acknowledgment (ACK) message after successfully receiving the data message to ensure reliability. As a result, devices may not transmit when they only receive the RTS message, as in MACA, since further transmissions may collide with the acknowledgment. MACAW also adds a data sending (DS) control message be-



(a) MACA Data Transfer (b) MACAW Data Transfer (c) MACA-BI Data Transfer

Boxes show when nodes may not transmit.

Fig. 1. Data Transfer in MACA, MACAW, and MACA-BI

tween the CTS and data messages. The DS message allows devices near the source to verify that a transmission will occur so they know to delay for the entire data message. If a device hears an RTS, but not a DS after a timeout period, then it knows the destination did not transmit a CTS and a different transmission may occur. MACAW provides data reliability at the MAC layer, but does so at the cost of an additional control message. Sensor networks that require reliable transmission use similar techniques, but not all sensor networks have this requirement. The DS control message, while a possible improvement in local area networks, does not improve the primary goal of reduced energy consumption within sensor networks. Most sensor network applications would trade the added throughput provided by the DS control message for the added lifetime provided by not transmitting or receiving the DS message.

Lastly, the MACA by invitation (MACA-BI) [13] protocol improves upon MACA in networks where devices continually generate data. MACA-BI differs from MACA and MACAW by having the destination devices initiate the data transfer process. Instead of a three-way transfer—RTS, CTS, and data—MACA-BI uses a two message transfer of a ready to receive (RTR) message from the destination followed by the data message from the source. MACA-BI thus saves a message transmission over MACA and increases the theoretical maximum throughput. However, MACA-BI's performance heavily depends on the destination's ability to predict the data it will receive. To help the destination predict traffic, the MACA-BI protocol provides an optional field within the data message that indicates the number of messages queued for the destination. Reducing the control message overhead makes MACA-BI more applicable to sensor networks than previous protocols, but constantly sensing the channel precludes its adoption.

Figure 1 shows data transfers for the MACA, MACAW, and MACA-BI protocols. For each protocol, boxes indicate when neighboring devices may not transmit because they defer access to a previous communication.

The simplicity of the CSMA, MACA, and derivative protocols certainly meet the requirement of simplicity for sensor networks. Unfortunately, the protocols require the transceiver to operate continuously, so sensor nodes would

consume energy far too quickly to make the deployment useful. Using the previous protocols, sensor nodes would only sleep when a transmission occurs, since no mechanism exists for devices to collaborate on a communication time. Further limitations come from the high overhead associated with using control messages for small data messages.

IEEE 802.11

Due to the popularity of the IEEE 802.11 [14] standard in wireless local area networks, we provide a brief introduction, but show that it does not suit sensor network applications for several reasons. IEEE 802.11 provides two modes of operation for wireless devices: an infrastructure mode where devices communicate through a central entity called an access point (AP) using the point coordination function (PCF), and an ad-hoc mode where devices communicate with each other directly using the distributed coordination function (DCF). The PCF extends upon the DCF and provides mechanisms for collision-free transmissions and device synchronization with the AP. Both the PCF and DCF use a channel access mechanism similar to slotted CSMA/CA and use acknowledgments for reliability. In addition to sensing the channel according to the CSMA algorithm, called physical carrier sensing, IEEE 802.11 devices perform virtual carrier sensing by tracking channel utilization with control messages. Each device maintains a counter, called the network allocation vector (NAV), that indicates the channel has activity on it whenever the NAV has a non-zero value. Devices update the NAV based on the data length present in control messages they receive. Periodically, each device decrements its NAV so that the current transmission ends when the NAV reaches zero. Using the NAV allows a device to quickly check for possible channel activity without having to activate the device's transceiver. For the purpose of determining channel activity, an IEEE 802.11 device considers the channel busy whenever physical channel sensing detects a transmission or when the NAV contains a non-zero value.

The DCF in IEEE 802.11 operates similar to slotted CSMA/CA with the use of virtual carrier sensing and acknowledgments. When first trying to transmit a message, a device senses the channel and, if free for a time period, transmits the message. If the device detects activity on the channel it defers access to the current transmission and performs the backoff algorithm. A device using the DCF considers the wireless channel idle if it detects no activity on it for a time period called the DCF interframe space (DIFS). An IEEE 802.11 device performs the backoff algorithm by randomly selecting a number of time slots to wait and storing this value in a backoff counter. For each time slot where the device senses no activity on the channel, it decrements its backoff counter and transmits a frame when the count reaches zero. If the device

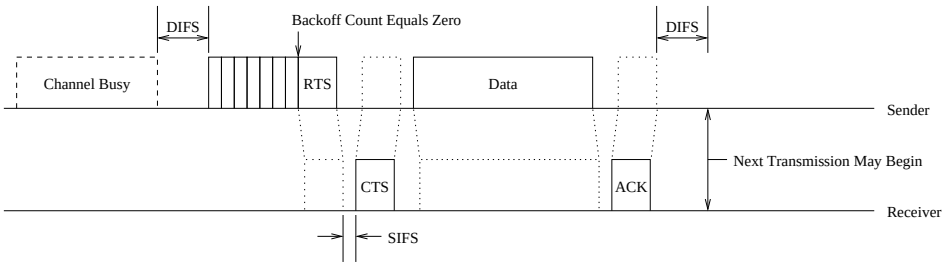


Fig. 2. IEEE 802.11 DCF Backoff Algorithm and Message Transfer

detects activity on the channel before the backoff counter reaches zero, it halts the countdown, defers access to the current transmission, and continues the countdown after the channel becomes idle for a DIFS. Devices that successfully receive a data message respond by transmitting an acknowledgment after a short interframe space (SIFS). IEEE 802.11 defines a SIFS shorter than a DIFS so that other devices do not physically sense an idle channel and cause a collision by transmitting over a control message. Figure 2, modified from the IEEE 802.11 standard, shows a message transfer when the sender detects channel activity upon the first carrier sense.

The PCF extends the DCF by having the AP coordinate collision-free time periods within its transmission range. The AP prepares for collision-free transmissions by broadcasting a beacon message that includes a list of devices to receive data during the next time period and an indication of the contention-free period's length. During the contention-free period the AP transmits messages to the devices listed in the beacon or, optionally, transmits polling messages to devices, which allows the devices to initiate data transfer with the AP. Before transmitting messages the AP waits for the channel to become idle for a PCF interframe space (PIFS) and will timeout after this period when it does not receive any expected response from a device. IEEE 802.11 defines the PIFS between the DIFS and SIFS in length; this allows the AP to have priority over devices operating in its range according to the DCF, but allows devices to transmit replies, such as CTS and ACK messages.

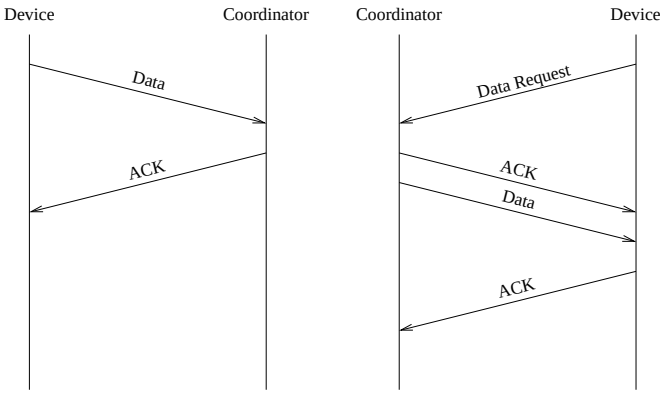
IEEE 802.11 does not suit sensor networks due to the differences of the intended applications. Characteristics important to devices operating on a wireless local area network, such as fairness, mobility support, high throughput, and low latency, influenced the design of the IEEE 802.11 standard, but these do not have as high a priority in sensor networks as energy conservation. As a result, IEEE 802.11 devices consume large amounts of energy due to the high percentage of time spent listening without receiving messages [15]. IEEE 802.11 does provide a simple energy management capability, called a power save mode, to devices operating according to the PCF. Devices that wish to sleep inform the AP using special control messages and enter sleep mode when they do not have messages to receive or transmit. Each device wakes up to receive beacon messages from the AP to determine if it must receive messages

during the contention-free period and to remain synchronized with the AP. The work by Ye et al. [15] provides some discussion of the IEEE 802.11 power save mode and notes the following limitations: power save mode only operates in infrastructure mode, so scalability becomes a problem, and the IEEE 802.11 standard does not specify when or for how long devices should sleep. Additionally, the protocol overhead in IEEE 802.11, which local networks can tolerate, becomes very large when used in sensor networks where applications may only generate a few bytes of data per message.

IEEE 802.15.4

In contrast to the IEEE 802.11 standard, IEEE created the 802.15.4 [16] standard for small devices that consume low power and require lower data rates. The IEEE 802.15.4 standard provides bitrates of 20kbps, 40kbps, and 250 kbps—much lower than the 1-54Mbps rates in IEEE 802.11—in the 868MHz, 915MHz, and 2.45GHz frequency bands, respectively. Similar to IEEE 802.11, the IEEE 802.15.4 standard provides a centralized topology, called the star topology, and a distributed topology, called the peer-to-peer topology. However, in every IEEE 802.15.4 personal area network (PAN) a single device acts as the PAN coordinator to control device association within the network. In the star topology all communication and resource reservation occurs through the PAN coordinator. Within the peer-to-peer topology, devices operate independently and need not communicate through the PAN coordinator, but all devices must associate with the PAN coordinator prior to participating in the network. The IEEE 802.15.4 standard focuses on the star topology and leaves many options and functionality of peer-to-peer networks undefined. As a result, the following discussion will focus on star-topology networks, but the standard does provide some hints on how the protocol may work in peer-to-peer networks.

Devices in an IEEE 802.15.4 network may operate in a beacon-enabled mode, where the PAN coordinator periodically broadcasts a beacon for synchronization and management purposes, or in an unsynchronized mode without beacons. Beacon-enabled PANs utilize the synchronization provided by the beacon to perform slotted channel access while PANs without beacons use unslotted access. IEEE 802.15.4 uses a slightly modified CSMA/CA algorithm to access the wireless channel. First, the device performs a random backoff before sensing the channel. If the device does not detect activity on the channel, and uses unslotted CSMA/CA, then it transmits the frame immediately. Devices using slotted CSMA/CA wait until the next slot and check the channel availability again. If a slotted CSMA/CA device detects no activity on the channel for two consecutive slots after the initial backoff period, then it transmits the message. Any time a device detects channel activity during the contention



(a) Data Transfer to Coordinator

(b) Data Transfer from Coordinator

Fig. 3. IEEE 802.15.4 Data Transfer

procedure, it performs the backoff algorithm and begins the process again at a later time. Devices only backoff a limited number of times before giving up on transmitting a message.

Since IEEE 802.15.4 focuses on energy constrained devices, the PAN coordinator does not initiate any data transfer. Figure 3 shows how data transfers occur within IEEE 802.15.4. Devices with data for the PAN coordinator transmit it according to the channel access mechanism described previously. The PAN coordinator may send an optional acknowledgment upon successful data reception. Data transfer from PAN coordinator to device uses more messages, but the receiving device still initiates the transfer. The device first sends a data request command to the PAN coordinator indicating that the data transfer may occur. If desired, the PAN coordinator may transmit an acknowledgment indicating it received the command successfully. The PAN coordinator then transmits the data message according to the channel access mechanism described previously. Finally, an optional acknowledgment lets the PAN coordinator know the device received the data. Beacon messages may include addresses of devices with pending data to signal the devices to begin a data exchange. PANs operating without beacons require devices to poll the PAN coordinator for data.

While IEEE 802.15.4 focuses on applications similar to sensor networks, several disadvantages exist for its use in sensor networks. First, the standard does not clearly define the operation of devices in a peer-to-peer topology, but only defines communication mechanisms for star topologies where devices can directly communicate with the PAN coordinator. Most sensor networks will have too many devices spread over too great a geographical area for all devices to use a single PAN coordinator. The standard does allow the inter-operation of different PANs, but it does not explore this method in detail. The Zigbee Alliance [17], an industrial consortium that defines the upper layer protocols used on top of IEEE 802.15.4, may outline standards for some of these opera-

tions and act as an informal standard. Research by Bougard et al. [18] shows some of the energy characteristics of the IEEE 802.15.4 standard through analytical modeling. The authors explore the benefits of power scaling transmissions and packet aggregation at the source, and provide a breakdown of energy consumption for various operations (e.g., beacon transmission, contention operations, and transmission). Despite the above disadvantages of the IEEE 802.15.4 standard, simply providing a standard may help in the proliferation of sensor networks and related applications, such as smart environments and ubiquitous computing. Crossbow Technology [19], a major sensor node manufacturer in the United States, has already begun shipping sensor node platforms based on the IEEE 802.15.4 standard and other manufacturers have produced devices for other applications.

3 Sensor Network MAC Protocol Differences and Constraints

The previous sections highlighted the differences between sensor networks and other wireless networks, and how they impact the MAC protocol design. This section expands upon the previous discussion and introduces common terms used throughout this paper.

3.1 Sensor Network MAC Protocol Differences

As discussed in the previous section, wireless MAC protocols proposed for other networks do not suit sensor networks for many reasons: the limited resources available on a sensor node, multihop operation of a sensor network, and different application requirements. Traditional wireless MAC protocols attempt to provide high throughput, low latency, fairness, and mobility management, but often have little or no consideration for energy conservation. Sensor network MAC protocols, however, must provide the best performance at the smallest amount of energy consumption due to the limited energy resources available to each sensor node. Sensor network MAC protocols often trade performance characteristics, such as throughput and latency, for a decrease in energy consumption to length a sensor node's lifetime. The most common approach to reduce energy consumption involves cycling the sensor node hardware between high power active states and low power sleep states. Sensor nodes can not function in the network while asleep, but putting the sensor node to sleep when unneeded can dramatically increase a sensor node's lifetime. Duty cycles—the fraction of time the sensor node spends awake—often dip below one percent in many sensor network applications in order to extend the network lifetime to acceptable levels. Further energy conservation comes from operating the sensor network in a multihop fashion where sen-

sor nodes forward messages to the destination for other sensor nodes. Single hop, or infrastructure, MAC protocols would consume too much energy for sensor networks deployed over large geographical regions because the transmit power required to correctly receive a message increases geometrically with distance—-typically between d^2 and d^4. Applications also differ between sensor networks and traditional wireless networks. Typical examples for sensor networks include environmental monitoring and target tracking and sensing; both of which consume small to moderate network resources under normal operation, but can produce large volumes of traffic when events occur. The wide variety of proposed applications for sensor networks provides a challenge for protocol designers because each application may produce traffic with different characteristics and require dramatically different performance metrics. Messages within sensor network applications often have a much smaller size when compared to traditional wireless networks. The smaller message sizes imply that protocol overheads from message headers increase and that the MAC protocols need not reserve long time periods for the transmission of typical messages.

Despite the substantial differences between sensor network MAC protocols and other MAC protocols, several common problems and solutions exist. Much of the research done for Ad Hoc networks may also apply to sensor networks since both operate as multihop wireless networks with power constraints. Ad hoc networks, however, focus on device mobility, while sensor networks normally have limited or no mobility. Ad hoc network devices typically have more resources available to them and lay between sensor networks and wireless local area networks in the spectrum of capabilities and resources. Long studied problems in wireless networks, such as the hidden terminal problem, also exist in sensor networks, so protocol designers must handle these issues in addition to the characteristics unique to sensor networks. Researchers now have the challenge to solve existing problems from traditional wireless networks under the constraints introduced by the limited resources available in sensor networks.

3.2 Sensor Network MAC Protocol Constraints

MAC protocols must perform the functionality required by the application while utilizing the limited resources available on sensor nodes. Limited energy resources place strict limits on the operations a sensor node may accomplish and differentiate sensor networks from other networks. Application and protocol designers must utilize the hardware resources on the sensor nodes judiciously to conserve energy and prolong the network lifetime. Three main hardware resources exist within a typical sensor node: the transceiver, the processor, and sensors. All MAC protocols utilize the transceiver and processor

during operation, but do so at different levels based on the protocol design and current sensor node conditions. Additionally, a MAC protocol design may require sensors or additional circuitry for proper operation, such as a Global Positioning System (GPS) receiver. Useful MAC protocols provide the highest level of functionality for a minimum of resource utilization.

Most current research on sensor network MAC protocols focuses on reducing the transceiver's energy consumption because the transceiver often uses more power than any other hardware resource. Designers attempt to limit transceiver energy consumption by preventing or limiting *collisions*, *overhearing*, *idle listening*, and *overhead*. Collisions within sensor networks cause the same problems as other wireless networks: performance limitation and energy waste. While many sensor network applications can cope with a slight performance decrease because they have low data rate requirements and high delay tolerances, energy waste due to frequent collisions can significantly decrease a sensor node's lifetime. Retransmitting a message requires the sensor node to operate its transceiver at the highest power levels—as opposed to sleeping—and consume multiple times the minimum energy required for that message. For sensor networks that do not require a reliable link layer, and thus do not retransmit messages, collisions have a smaller impact, but the loss of data may decrease the application's accuracy. Several sensor nodes may receive the same transmission, possibly multiple times with retransmissions, even though the source intended it for only one recipient. In these cases the unintended receivers overhear the message and waste energy on reception and processing. MAC protocols may limit, but can not prevent overhearing from occurring in some fashion. Fortunately, MAC protocols can leverage overhearing to infer information about the wireless channel, such as sensor node availability or link status, and decrease the effective energy loss. A MAC protocol may also end a reception early and enter the sleep state to limit the energy losses associated with overhearing messages once it determines the message belongs to another node. For example, if the message format includes the destination address early in the transmission and receiving sensor nodes can obtain the message data as it arrives, then the transmission can end after the sensor node has processed the address.

Energy waste also occurs when no sensor node transmits a message, but nearby sensor nodes attempt to receive a message. In this case the receiving sensor nodes perform idle listening and waste the energy consumed by the transceiver during this time. Reception does not consume as much energy as transmission in most designs, but it does consume many times more power than if the sensor node placed the transceiver in the sleep state. Idle listening can account for a significant portion of the energy a sensor node consumes in some cases [20]. A typical solution to limit idle listening uses a timer to end reception if the sensor node does not detect any activity on the channel. Note that idle listening does not include carrier sensing, which many MAC protocols require for proper

operation. In carrier sensing, the transceiver performs useful work for the MAC protocol, so it counts as a protocol requirement and not an energy waste.

Carrier sensing, however, provides one example of a protocol overhead. The overhead required by a MAC protocol depends on its design and may range from an increased switching rate to additional message communications. Typical overheads in sensor network MAC protocols include synchronization messages, longer preambles, and control messages. The protocol overhead serves some purpose for the MAC protocol and differentiates the protocols from each other. For example, MAC protocols may use synchronization messages to organize sensor nodes together or allow sensor nodes to estimate distances based on the received signal strength. The most common overhead for MAC protocols involve using control messages to solve the hidden terminal problem and provide reliability.

MAC protocol designers must also contend with the functionality provided by the transceiver chosen for the sensor node. Designers commonly consider the power consumption for the various modes of operation, but other characteristics may have equal importance. Most sensor network transceivers consume the same energy in receive mode whether they receive a message or only receive noise. A transceiver that can listen to the channel with very low power can save a great deal of energy normally expended on idle listening. While a low power listen mode may never consume as little energy as a sleep mode, it can have a large impact on power savings over the lifetime of the sensor network if utilized properly by the MAC protocol. A transceiver that has multiple energy conservation states provides the MAC protocol the flexibility to conserve as much energy as possible and still respond quickly when needed. For example, most transceivers have a single sleep state where nearly all circuitry remains off. Energy conservation in the sleep state comes at the cost of a considerable delay to switch the transceiver to an active state, during which time the transceiver can not do any useful work. A transceiver with a near-sleep state that keeps critical circuits operational allows the MAC protocol to still conserve some energy, but also allows it to respond quickly to various demands. MAC protocol designers must also consider the transceiver state switching times when constructing protocols to prevent violating protocol timing. For example, a protocol that attempts to sleep for a time period shorter than the state switching time may miss a transmission it expects when it awakes. Similar problems also arise from the use of low accuracy oscillators to reduce sensor node cost. Several interrelated factors affect the transmission radius of a sensor node. Transmission power provides the clearest example: transmitting with a higher power will, in general, allow sensor nodes further away to communicate at the cost of more energy. The available modulation schemes in a transceiver can also affect the transmission range for a given bit-error-rate (BER). However, complex modulation schemes may require a more complex transceiver, which can cost more and consume more energy. Modu-

lation schemes proposed or used for sensor networks range from very simple, such as On-Off Keying (OOK) and Binary Phase Shift Keying (BPSK), to very complex, such as Direct Sequence Spread Spectrum (DSSS) and Ultra-Wide Band (UWB). Researchers have considered the transmission rate and modulation schemes of sensor nodes in an effort to reduce energy consumption [21,22]. Channel coding provides another way to extend the transmission distance or improve the BER at the cost of computational resources [22,23]. Finally, the transceiver choice determines the possible bit rates available, but modulation schemes, coding, and protocol overheads lower the effective available data rate.

Another concern for MAC protocol designers comes from the limited computation and storage resources available on sensor network nodes when compared to wireless devices used in other networks. Few MAC protocol proposals consider the processing requirements required for normal operation, but a complex MAC protocol might decrease the time a sensor node spends in the sleep state or consume a large fraction of the available processor time and limit the processing available for the application and other protocols. An overly simple MAC protocol, however, may not provide comparable energy savings to a more complex protocol that can adapt to channel conditions and decrease transceiver energy consumption. Moreover, a more complex MAC protocol may provide functionality, such as clustering and topology estimation, required by other protocols for less energy than if the functionality occurred independent of the MAC layer. MAC protocol designers must consider the processing resources required by their protocols and ensure that the functionality they provide enables the sensor node to perform useful work at the application layer. Sensor nodes also provide limited memory resources, and their use parallels many of the trade offs seen for processing resources. A MAC protocol that maintains large amounts of state will consume more memory than MAC protocols that maintain no history, but tracking the sensor node or channel information may allow the protocol to conserve energy in other areas, such as decreasing collisions. Utilizing memory also leaves fewer memory resources available for data collected by the application, control structures for other protocols, and program space. Frequent data memory accesses also increase the energy consumed by the memory circuitry as the memory cells switch more often.

Several forces drive sensor network protocol and application designers to focus on distributed algorithms rather than centralized organization [5]. For MAC protocols, this implies that traditional methods of resource allocation and management that rely on centralized, global information will not perform well within sensor networks. The low data rate and multiple hops necessary to share information across the entire sensor network greatly increase the protocol response time. By the time the resource management entity could adapt to a change in the sensor network, the conditions may have become worse or

the anomaly may have subsided. Additionally, sharing this information consumes large amounts of energy as the sensor nodes transmit and forward the control messages. The protocol designer, however, must balance the benefit of sharing some information between nearby sensor nodes, in order to reach a locally optimal operating point, with the cost of sharing that information. MAC Protocols must provide scalability both in network size and sensor node density to support sensor networks of many hundreds to many thousands of sensor nodes.

Finally, the MAC protocol may require sensor readings for operation. The sensors, along with other needed circuitry such as analog to digital converters, consume power and thus cause additional overhead on the energy resources and additional cost for the sensor node production. An example includes MAC protocols that measure the received signal strength for distance or link quality estimates. Similar to other resources, the benefits provided by the information gathered from the sensors may offset the sensor's cost, but this depends on the sensor node protocols and the application. MAC protocols that use sensors already present for the application can achieve the benefits with minimal additional cost.

4 Sensor Network MAC Protocols

Many researchers have recognized the unique operating environment and platform present in sensor networks and proposed many MAC protocols specifically for them. We cannot cover the multitude of proposed protocols in the literature because of space, but include in this section a discussion of many representative protocols. Two general classifications for sensor network MAC protocols exist: scheduled protocols and unscheduled, or random, protocols. Scheduled MAC protocols attempt to organize nearby sensor nodes so their communications occur in an order way. The most common scheduling method organizes sensor nodes using time division multiple access (TDMA) where a single sensor node utilizes a time slot. Organizing sensor nodes provides the capability to reduce collisions and message retransmissions at the cost of synchronization and state distribution. Unscheduled protocols attempt to conserve energy by allowing sensor nodes to operate independently with a minimum of complexity. While collisions and idle listening may occur and cause energy loss, the unscheduled MAC protocols typically do not share information or maintain state. Some proposed MAC protocols do not easily fit into this classification scheme and other classifications exist, but the discussion below focuses on dividing the MAC protocols based on their large-scale organization of sensor nodes or lack thereof.

Most sensor network MAC protocols have some overlap in their effort to limit

energy consumption. The most common and effective way to conserve energy places the transceiver and processor into a low power sleep state when the resources have no work to perform. In this way the sensor node can consume much less energy—typically several orders of magnitude less—than if the processor entered a busy loop and the transceiver entered an idle state. Sensor network MAC protocols may sleep periodically for fixed, known durations or may sleep for random lengths of time depending on how a sensor node interacts with other sensor nodes. The duty cycle of a sensor node corresponds to the fraction of time the sensor node remains in an active state. Sensor nodes that maintain a high duty cycle can respond to traffic and network changes more quickly, but consume energy at a higher rate. A lower duty cycle MAC protocol can save energy, but low activity levels place a limit on the protocol's complexity, the possible network capacity, and the message latency. MAC protocols often have the duty cycle as a protocol parameter.

4.1 Unscheduled MAC Protocols

Unscheduled MAC protocols offer the advantage of simplicity. Without having to maintain and share state, an unscheduled MAC protocol may consume fewer processing resources, have a smaller memory footprint, and decrease the number of messages that a sensor node must transmit. Additionally, sensor nodes that get added to the network, through redeployment or movement, can begin to participate much more quickly because they do not have to obtain the current schedule or join another sensor node group. However, unscheduled MAC protocols experience, in general, a higher rate of collision, idle listening, and overhearing because the sensor nodes do not coordinate transmissions. Mitigating the effects of these common problems requires unscheduled MAC protocols to use additional techniques, such as channel sensing and channel reservation messages, which may offset the benefit of not organizing the sensor nodes. Unscheduled MAC protocols also allow sensor nodes to adapt more easily to changing traffic conditions because channel reservation can occur with finer granularity and sensor nodes can adaptively contend for the channel. Scheduled MAC protocols must coordinate the sensor nodes to redistribute resources, which causes a delay between resource reservation and resource utilization. An unscheduled MAC protocol can decrease or remove the resource allocation delay, allowing a much faster adaptation to changing conditions. Fairness becomes an issue in unscheduled MAC protocols because no mechanism implicitly exists that equalizes the channel usage, unlike in a scheduled MAC protocol.

4.1.1 Multiple Transceiver MAC Protocols

Since the transceiver consumes so much energy per use, it may seem counterproductive to use multiple transceivers on each sensor node, but several design approaches could yield a net energy reduction for the sensor node. For example, each transceiver may operate at a lower duty cycle than a single transceiver by dividing the sensor node's communication requirements between the transceivers. Multiple transceivers also enable the sensor node to communicate simultaneously on separate channels, if needed, to increase bandwidth or shorten response time. These benefits come at the cost of additional hardware requirements. First, transceivers constantly consume energy, even while asleep, so adding transceivers increases the energy consumption a sensor node can not control by power cycling hardware. Second, a multiple transceiver system must possess the computational capability to receive and process data from multiple channels. Therefore, multiple transceiver systems require higher performance communication mechanisms and processor capabilities than single transceiver systems. Finally, adding multiple transceivers and a more powerful processor may lower the overall energy consumption of the node, but requires the sensor node design to include an energy source that provides enough power for all the hardware devices when operated in unison. To make multiple transceiver MAC protocols viable, protocol and device designers must overcome the energy losses in transceivers that arise independent of utilization and contend with the additional sensor node complexity and cost.

PAMAS

The Power Aware Multi-Access with Signaling (PAMAS) [24] protocol, originally proposed for Ad Hoc networks, attempts to conserve energy by utilizing two transceivers: one for data messages and the other for control messages. By separating the message transfers devices can prevent collisions of the larger data messages and save the power otherwise used on retransmissions and overhearing. Control channel exchanges use RTS and CTS messages like MACA, but PAMAS also uses busy tone transmissions as proposed by Tobagi and Kleinrock [25]. A receiving device uses the busy tone to indicate that other devices, which may have missed the RTS and CTS messages, may not transmit on the data channel. Figure 4 shows a message transfer in PAMAS.

Message transfer in PAMAS starts by the source sending an RTS message to the destination on the control channel. The destination then decides if it should transmit a CTS by examining the data and control channels. If the destination does not detect activity on the data channel and has not heard an RTS or CTS message recently it responds with a CTS message. A source that does not receive a CTS in time will backoff using a binary exponential algorithm. Once the source receives a CTS message it transmits the data message over the

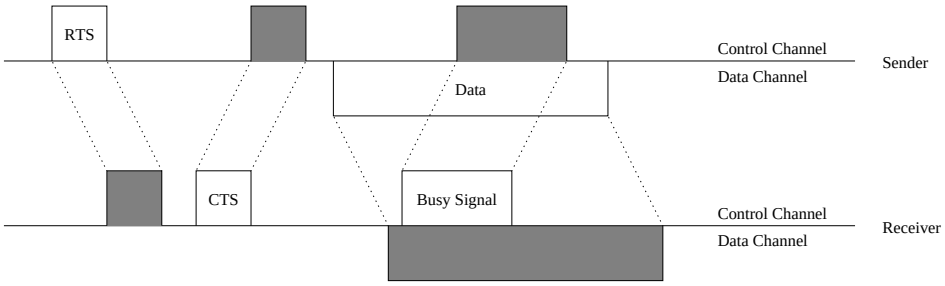


Fig. 4. PAMAS Data Transfer

data channel. The destination starts transmitting a busy tone over the control channel once it starts receiving the data message so that nearby nodes realize they may not use the data channel. PAMAS implements a busy tone as a message twice the length of an RTS or CTS message. Furthermore, during the data reception the destination will transmit a busy tone any time it receives an RTS message or detects noise on the control channel to corrupt possible CTS message replies and prevent further data transmissions.

PAMAS devices power down under two conditions: the device has no data to transmit and a neighbor device begins transmitting to another device, or when the sensor node has two neighbors involved in communication. The first case saves energy since the device can not receive a data message without corruption, so the node may power down the transceivers. The second condition saves energy since the device can not transmit or receive without a collision resulting at itself or its receiving neighbor. To determine the length of time to sleep, each data message includes the transmission duration so a device that overhears the start of the message can calculate the length of time to sleep. However, if the device awakes to an ongoing message transmission it must determine the length of time to sleep. To do this, a device transmits a probe request message onto the control channel that requests if the message transmission will end over a particular interval of time. Any neighboring device currently transmitting a data message replies on the control channel with the remaining transmission duration. If the probing device receives a response, it will know to sleep until the time included in the response. The probing device may receive multiple responses that collide and in this case the device must perform a binary search over the interval until it receives a single response. Only devices that do not have messages to transmit need to use the probing process since a successful transmission might still occur in parallel with a neighboring device. In this case, the device transmits an RTS message as normal and the neighboring receiver, if it exists, responds with the busy tone, which includes the remaining time for message reception. However, the device may receive noise due to message collisions and in this case the device polls neighboring receivers and transmitters in a similar manner to that discussed previously. The device can then sleep for the longest transmission or reception of its neighbors.

The authors propose other options and possible improvements for PAMAS. First, the probing protocol could include CSMA mechanisms to reduce collisions, but the authors argue not to do so because under light load devices will likely find the channel available, and under heavy load the awakened device will likely have messages to transmit and get the information from a busy tone response to an RTS. A trade off exists between using the probe protocol or leaving the control channel radio always powered. If the device does not turn off its control radio, then it will receive the RTS and CTS messages of its neighbors and will know the length of any data transmission. Leaving the radio powered on could save energy over using the probing algorithm, especially if collisions frequently occur in probe replies and devices commonly probe the channel. Further improvements could include the addition of ACK messages, allowing nodes to transmit an ACK instead of a CTS if errors corrupted a previous ACK, message aggregation to decrease the overhead of control message exchanges, and support for broadcasts.

Perhaps the largest drawback to PAMAS involves the multiple radio requirement. Including multiple radios on a device will greatly increase the energy consumption and the device cost for sensor networks. Additionally, controlling access to two wireless mediums increases the MAC protocol complexity. The small message size present in most sensor networks also decreases the benefits of separating the data and control transmissions. However, ideas such as those proposed through PAMAS may work for sensor networks with large data messages if the sensor node and transceiver design can decrease the cost of an additional transceiver.

4.1.2 Multiple Path MAC Protocols

One technique for medium access involves simplifying the MAC layer to such an extent that it only transmits messages after a delay. Eliminating control messages and carrier sensing removes the overhead involved with those operations. However, to increase the probability of message delivery, many copies of each message may propagate through the network. The backoff mechanism provides the main function for the MAC protocol and must decrease the chances of collision. Any simplifications must overcome the overhead associated with transmitting a message many more times than necessary in order to provide benefit to the application. The following protocols take this approach by probabilistically forwarding multiple copies of each message to the destination.

SRBP, ARBP, and RARBP

In contrast to the previous protocol, those proposed by Chatzigiannakis et al. [26] do not use control messages, but transfer messages along multiple, different paths. The MAC protocols conserve energy by reducing the probability of collision with a random delay before each transmission. In this way nearby sensor nodes that receive a message to forward do not all transmit at the same time and the probability of a successful transmission increases. To route the messages, the protocols use the Probabilistic Forwarding Protocol (PFR), presented earlier [27], at the network layer. PFR assumes that sensor nodes have a directional transmission capability, knowledge of the base station direction (sensor nodes do not need the actual base station location), and that sensor nodes generate traffic only for the base station. A sensor node that receives a message will broadcast it with a certain probability based on the angle formed between the message source, the forwarding sensor node, and the base station. Sensor nodes that have an angle closer to 180° broadcast the message with a higher probability than sensor nodes further from the line connecting the source to the base station. Sensor nodes drop messages not selected for forwarding.

Chatzigiannakis et al. propose three protocols where each variation slightly improves the previous one [26]. First, the authors propose the Simple Random Backoff Protocol (SRBP), which functions by simply transmitting a message after an initial random backoff. The sensor node does not attempt to sense the channel before transmission nor does it transmit any control messages. To limit collisions the sensor node selects the backoff, t_b, at random from a range of values, $\mathcal{T}_b = [\mathcal{T}_b min, \mathcal{T}_b max]$, which remain constant during the sensor network's lifetime. The second protocol, the Adaptive Random Backoff Protocol (ARBP), attempts to improve performance by taking into consideration the sensor node density in the local region and the current traffic conditions. It does this by adjusting the maximum backoff value, $\mathcal{T}_b max$, according to two sub-protocols that estimate the sensor node density, d_l, and the traffic density, $\mathcal{I}_l$. To estimate the sensor density, the sensor node maintains a list of the node IDs it has heard recently. The sensor node removes a node ID from the list if it does not receive a message with that ID over a time period. The count of the node IDs estimates the local sensor node density. A simple counter of the number of messages received per time period estimates the traffic density at the sensor node. To determine the next maximum backoff value, $\mathcal{T}_b max$, the sensor node uses the previous value, $\mathcal{T}_b^- max$, along with the traffic and sensor node densities according to the function $\mathcal{T}_b max = \mathcal{T}_b^- max + \alpha C_d + \beta C_t$, where end users may select $\alpha, \beta \in [0, 1]$ as system parameters, $C_d = \mathcal{T}_b^- max \frac{d_l - d_l^-}{d_l + d_l^-}$, and $C_t = \mathcal{T}_b max \frac{\mathcal{I}_l - \mathcal{I}_l^-}{\mathcal{I}_l + \mathcal{I}_l^-}$. Similar to the maximum backoff value, $\mathcal{I}_l^-$ and d_l^- correspond to the previous traffic and sensor node density estimates, respectively. The final protocol, the Range Adaptive Random Backoff Protocol (RARBP),

attempts to decrease message latency by giving sensor nodes further from the transmitter a higher probability of transmitting earlier. To do this, sensor nodes now select the random backoff value for each message from a normal distribution with mean $\mathcal{T}_b min + (\mathcal{T}_b max - \mathcal{T}_b min) \frac{d_{es}}{R}$ and standard deviation $\frac{1}{d_l}$, where d_{es} corresponds to the estimated distance from the previous transmitter to the forwarding sensor node and sensor nodes can communicate up to a distance R. By allowing farther sensor nodes to transmit earlier, RARBP shortens the message latency since each message traverses fewer hops, but this requires sensor nodes estimate distance or posses location information.

The resource requirements and inefficiencies of these protocols may outweigh the benefits of their functional simplicity. Transmission will likely result in many collisions, despite the proposed backoff algorithms, since the transmissions occur without any coordination. In order to reduce the probability of collision to reasonable levels, the backoff time may have to increase to intervals that would result in unacceptable message latencies, especially for dense networks or sensor networks that generate large amounts of data. Additionally, since sensor nodes do not communicate information about transmission success the protocol wastes energy transmitting the same message along multiple paths and can not provide reliable or guaranteed delivery. However, for some applications that generate light traffic and only require some messages to arrive at the destination, especially for sensor nodes with very limited computing resources, these protocols may provide an advantageous solution.

4.1.3 Event-Centered MAC Protocols

Sensor network applications have varying application requirements and traffic patterns, so MAC protocols may conserve the most energy by taking advantage of unique characteristics within a network. For example, a target detection sensor network will have very little traffic most of the time, but may produce relatively large volumes of data when an event of interest occurs. A MAC protocol that operates based on the assumption of constant traffic generation would waste energy when the sensor network contained no targets. Further energy conservation could come from the MAC protocol playing an active role in forwarding messages according to some application parameters, such as a maximum number of reports to forward or an accepted latency. The following protocol considers the application requirements to control the energy expended by forwarding traffic.

CC-MAC

Vuran and Akyildiz [28] take a more holistic approach to MAC protocol design by allowing application requirements to influence the MAC protocol's

operation. The spatial Correlation-based Collaborative MAC (CC-MAC) protocol attempts to conserve energy, while fulfilling application requirements, by utilizing the knowledge that sensor nodes located near each other generate correlated measurements. To achieve energy savings, CC-MAC filters measurements from highly correlated sensor nodes in an effort to reduce the number of messages the sensor network must handle. Lowering the message volume reduces wireless medium contention, so fewer collisions occur, reduces the number of messages sensor nodes must transmit and receive, and allows sensor nodes to utilize lower duty cycles.

To estimate the amount of filtering to perform, the authors introduce an analytical framework that models a sensor node's sensing capabilities and the effect of filtering on the application result. Based on analysis within the framework, the authors introduce the Iterative Node Selection (INS) algorithm that generates a filtering parameter, called the correlation radius, based on statistical information about the sensor network deployment. Sensor nodes closer than the correlation radius produce correlated, and therefore redundant, information while sensor nodes located farther than the correlation radius generate independent results. Thus, the protocol may filter data from sensor nodes closer than the correlation radius while still satisfying the application constraints. Since the INS algorithm proposed requires more computational resources than a typical sensor node has available, the sensor network sink runs the algorithm during the network setup and distributes the calculated correlation radius throughout the network. Note that since the INS algorithm only requires statistical and not actual data about the sensor node deployment the sink only needs to calculate the correlation radius during the network initialization.

CC-MAC itself consists of two components: the Event MAC (E-MAC), which filters sensor node measurements to reduce traffic and the Network MAC (N-MAC), which forwards the filtered measurements to the sensor network sink. E-MAC reduces the traffic generated in an area by having only sensor nodes separated by at least the correlation distance generate measurements. Other nodes periodically sleep to save energy and awake to forward messages. Correlated sensor nodes rotate the role of generating measurements to balance energy consumption throughout the network. Sensor nodes get elected as the representative of the correlated sensor nodes by winning contention for the wireless medium. E-MAC slightly modifies the standard RTS/CTS/DATA/ACK scheme in the IEEE 802.11 standard by introducing a First Hop (FH) bit into the control packet headers. The sensor node actively reporting measurements sets the FH bit when it transmits messages so that other nodes can decide to generate measurements or not. If a sensor node lies further than the correlation radius from all other sensor nodes generating measurements, then it will begin to also generate measurements. The authors discuss cases where the transmission radius of the sensor nodes extend

further than the correlation radius and where the correlation radius extends beyond the transmission range. Once the originating sensor node has transmitted the measurement, the FH bit gets cleared and the message becomes a forwarding message for the N-MAC protocol. N-MAC forwards messages from sensor nodes generating measurements to the sensor network sink, but since the E-MAC protocol has removed most of the redundancy present in multiple measurements the forwarded traffic becomes more important. To compensate for this, N-MAC protocol transmissions take preference over E-MAC transmissions through the use of smaller backoff windows and inter-packet times in same way that the PCF in IEEE 802.11 receives preferential access to the wireless channel over the DCF.

The authors compare the CC-MAC protocol to several other sensor network MAC protocols through simulation and show that CC-MAC can achieve a good balance of low energy consumption and favorable traffic performance compared to the other protocols. Additionally, the analytical framework proposed in their work allows users to apply the CC-MAC protocol to applications with various data fidelity requirements. CC-MAC, however, requires that sensor nodes posses or obtain ranging information about their neighbors in order for N-MAC to filter data from correlated sensor nodes. The complicated nature of the INS protocol may also limit the application of the protocol. As the number of sensing events increases, especially if the sensing conditions change with time, the overhead associated with computing the correlation radius and distributing throughout the network increases. For large networks this overhead may become significant.

4.1.4 Encounter-Based MAC Protocols

MAC protocols, especially unscheduled ones, face the challenge of awaking sensor nodes that must communicate. In an unscheduled MAC protocol, the sensor nodes may not know the sleeping schedules of their neighbors, so they must somehow probe with messages until the neighbor awakes. Once the communicating sensor nodes encounter each other in time they can begin the message transfer. Several techniques exist, beyond developing a schedule, for the encounter mechanism and the following protocols illustrate these techniques. The energy savings provided by encounter-based MAC protocols come from only synchronizing nearby sensor nodes when needed and only for the duration of the transmission. Traffic patterns, however, dictate whether the encountering mechanisms will consume less power than scheduling sensor nodes continually, with rare and random message generation patterns benefiting more from an unscheduled MAC protocol.

STEM

An early unscheduled MAC protocol design for sensor networks includes the Sparse Topology and Energy Management (STEM) [29] protocol. While STEM does not provide some common MAC protocol functionality, we present it here because it illustrates a necessary function of sensor network MAC protocols: coordinating sensor nodes that may sleep independently so communication can occur. STEM attempts to conserve energy by allowing sensor nodes with a message to transmit to wake up neighboring sensor nodes that may have entered the sleep state with as little effort as possible. A sensor node wakes a neighbor by transmitting either repeated beacon messages (STEM-B) or a wakeup tone (STEM-T). In STEM-B a sensor node with messages to transmit alternates between transmitting beacon packets and listening for a reply from the intended receiver. By periodically sensing the channel, the receiver can catch one of the beacon packets and reply to the source with a small acknowledgment packet. STEM-T works in a similar way except that the source sensor node transmits a tone of sufficient length that the destination will have a high probability of sensing the tone. Once the nodes finish signaling, a full-functioned MAC protocol transfers the message. In the paper, the authors argue that the wakeup and data transfer should occur on separate radios, but that the process also works with single transceiver sensor nodes.

TICER and RICER

Similar protocols include those presented by Lin et al. [30] as the Transmitter Initiated Cycled Receiver (TICER) and Receiver Initiated Cycled Receiver (RICER) protocols. The TICER protocol operates similarly to STEM-B, by having sensor nodes with data to send periodically transmit RTS control messages followed by a sensing period. Receivers periodically listen to the wireless channel and if they detect an RTS message, reply with a CTS message. The sensor nodes can then transfer the data message. RICER reverses the operation, so receivers periodically transmit beacons when they awake from their normally scheduled sleep time. Sensor nodes with data to transmit listen on the channel until they hear the beacon from the intended receiver. The authors compare the performance of RICER and TICER in the paper and show that protocol parameters, such as the time between control messages, and the channel characteristics play an important role in overall performance. Further investigations into various forms of synchronicity, number of receivers, and using a wakeup radio show the benefits of these techniques.

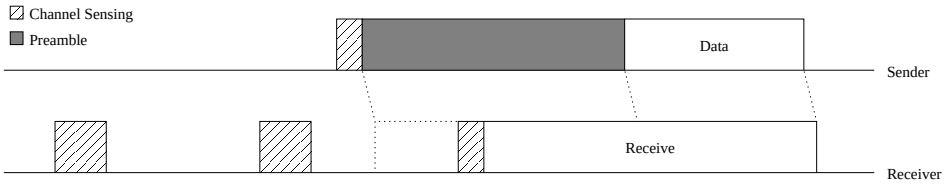


Fig. 5. BMAC Data Transfer

B-MAC

Similar to STEM-T, the Berkeley MAC (B-MAC) [31] protocol, which extends previous work [20], uses a tone to wakeup sleeping neighbors. In B-MAC sensor nodes independently follow a sleeping schedule based on the target duty cycle for the sensor network. Since the sensor nodes operate on independent schedules, B-MAC uses very long preambles for message transmission. The source sensor node transmits a preamble long enough that the destination, which periodically senses the channel, has enough time to wakeup and sense activity. Sensor nodes that sense activity on the channel remain awake to receive the message following the preamble or return to sleep if they do not detect activity on the channel. Before transmitting, sensor nodes delay a random time to prevent synchronization, and sense the channel to prevent corrupting an ongoing transmission. Figure 5 shows a message transfer in BMAC. Since B-MAC relies on accurately determining the channel status, it defines a filtering mechanism that increases the reliability of channel assessment. Additionally, the B-MAC authors provide a great deal of flexibility through a protocol interface that allows the sensor node to change many operating variables in the protocol, such as delay and backoff values.

Typical of an unscheduled MAC protocol, B-MAC relies on a version of CSMA suited for a sensor network platform. As such, B-MAC provides no implicit protection against traditional wireless problems, such as the hidden terminal problem. Other protocols must provide the functionality or accept the performance overhead associated with the losses. Sensor nodes using B-MAC have instant access to the network once deployed or moved since the protocol requires no setup or prior communication. Furthermore, unlike scheduled protocols, B-MAC does not have to delay messages waiting for a valid time access the channel. As long as a sensor node does not corrupt an ongoing reception, a sensor node can begin transmitting a message immediately. The long preambles in B-MAC and similar protocols do introduce an additional latency, but end users can consider this in the sensor network design and sensor nodes may control it through the protocol interface. A shorter sleep time will yield a lower latency at an additional energy cost.

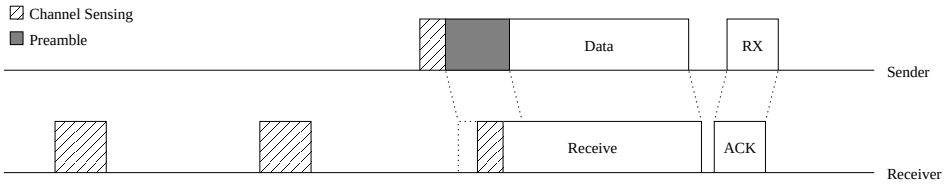


Fig. 6. WiseMAC Data Transfer

WiseMAC

A similar protocol, WiseMAC [32], developed about the same time as B-MAC, uses similar techniques, but attempts to reduce energy consumption by having sensor nodes remember the sampling offsets of their neighbors. An extra field in ACK packets allow sensor nodes to notify their neighbors of the time until their next channel sampling. By learning the sampling times of its neighbors, a sensor node can delay transmitting the preamble until just before the receiver wakes up to sense the channel. WiseMAC can thus decrease the amount of time a sensor node transmits preambles and the number of sensor nodes that overhear each message at the cost of an extra field in the ACK messages and the memory required to store neighbor's sampling offsets. Figure 6 shows a message transfer using WiseMAC. Notice that for the same sample rate the time spent receiving and transmitting the message preamble has reduced from that in B-MAC.

CSMA-MPS

Researchers further attempted to improve energy and latency over BMAC and WiseMAC in the development of the CSMA with minimal preamble sampling (CSMA-MPS) [33] protocol. In CSMA-MPS instead of transmitting a long preamble the source sensor node alternates between transmitting small control messages and listening for a response from the receiver sensor node very similar to STEM-B and TICER. Using small control messages has several advantages. First, it allows the source sensor node to determine sampling offset of the destination sensor node with moderate accuracy, so learning a neighbors sampling offset requires no extra fields the ACK messages. Second, the small control messages sent by the source node can act as RTS messages and the destination's reply can act as the CTS. Finally, for very small data messages the control messages can perform the data transmission with the reply acknowledging the reception. These improvements come at the cost of a greatly increased switching rate for the transceiver. Figure 7 shows a message transfer in CSMA-MPS where the destination receives the second wakeup message.

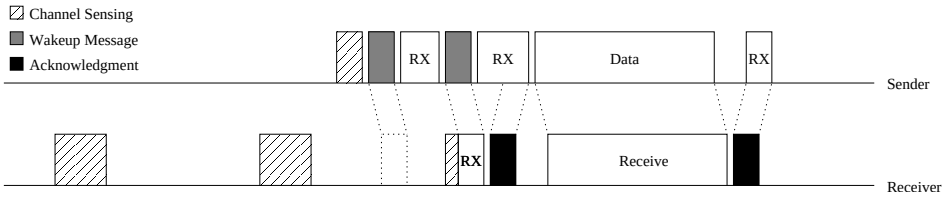


Fig. 7. CSMA-MPS Data Transfer

Protocol Type	**Summary**	**Advantages**	**Disadvantages**
Multiple Transceiver	Separate data and control traffic on different transceivers	Reduce collisions with long data messages	Hardware and energy resource cost
Multiple Path	Forward messages along multiple paths	Simple protocol	Collisions more common, messages forwarded multiple times
Event-Centered	Manage traffic based on application requirements	Filter redundant data, sensing fidelity framework	Parameter calculation and global distribution
Encounter-Based	Beacons or periodic tones to coordinate communication	Simple protocol, used only when needed	Many or long control messages sent per data message

Table 1
Unscheduled MAC Protocol Summary

4.1.5 Unscheduled MAC Protocol Summary

As the discussion throughout this section illustrates, unscheduled MAC protocols leverage simplicity to minimize resource utilization within a sensor node. However, they generally provide less functionality than a scheduled protocol, so other protocols must implement needed operations. Coordinating neighboring sensor nodes for communication, a problem implicitly solved in scheduled MAC protocols, becomes a primary function of unscheduled MAC protocols. End users that require very simple MAC protocols because of resource constraints or only require limited functionality may find an unscheduled MAC protocol the best option. Table 1 summarizes the unscheduled MAC protocols discussed for sensor networks.

Scheduled MAC protocols attempt to reduce energy consumption by coordinating sensor nodes with a common schedule. Most proposed protocols use some form of TDMA since other forms of multiple access, such as frequency or code division, would increase the cost and power requirements of the sensor nodes. By producing a schedule, the MAC protocol clarifies which sensor nodes should utilize the channel at any time and thus limits or eliminates collisions, idle listening, and overhearing. Nodes not participating in message communication may enter the sleep mode until they have work to perform or need to receive a message. Additionally, the MAC protocol can share traffic or status information so that the individual sensor nodes can optimize energy consumption over a collection of sensor nodes instead of at just a single sensor node. For example, nodes with important traffic or with a larger backlog of messages may get preferential treatment in the assignment of time slots. Simple traffic engineering also becomes possible by sharing state among sensor nodes, allowing a much higher level of fairness to exist within the sensor network.

However, these advantages come at the cost of increased messages to create and maintain a schedule. Node mobility, node redeployment, and node death all complicate schedule maintenance. Sensor nodes that enter the network must wait until they learn, and possibly join, the schedule in order to utilize the channel. Additionally, some delay exists between the time a sensor node dies and the time neighboring sensor nodes reassign its resources, so some resources may go unused and lead to unnecessary delays or packet loss. Scheduled MAC protocols must also operate properly under situations where sensor nodes posses incorrect state. A segmentation of the MAC state may lead to conditions where collisions cancel the benefits provided by the scheduled protocol. Synchronization becomes an important problem for a scheduled protocol and may occur through a periodic beacon, which increases the transceiver utilization, or by using higher precision oscillators, which increases the sensor node cost. Scheduled MAC protocols must also minimize the effect of added latency and limited throughput. Typically, each sensor node can only access the wireless channel for a fraction of the possible time. With a TDMA-based MAC protocol the time a sensor node may access the channel depends heavily on the time slot length. Typically, only one sensor node may transmit during that interval, so any unused time goes to waste. Reducing the time slot length may decrease the waste, but also decreases the maximum message length without fragmentation. Sensor nodes that wish to transmit messages at a higher rate than the current reserved time slots can handle must coordinate with other sensor nodes on the schedule to gain access to the extra time slots. Thus, each sensor node must queue messages until it has a chance to transmit them. Several scheduled MAC protocols attempt to overcome the limitations

on throughput and latency at the cost of sharing additional information in messages or higher duty cycles.

4.2.1 Priority-Based MAC Protocols

The series of protocols proposed by Bao and Garcia-Luna-Aceves [34] base channel access on the priority of nodes or links derived from a random function. Sensor node IDs and time slots numbers provide an input to a random function that establishes the priority within a two hop neighborhood. Each of the three protocols activates different entities, but they all use the idea of giving access to the entity with the highest priority. For example, using sensor node IDs as the entity, a sensor node, i, may get assigned priority $p_k^t = \text{Rand}\,(i \oplus t) \oplus i$ for the time slot t. The protocols share topology information by including neighbor information in data messages and each sensor nodes maintains information about its two-hop neighborhood.

NAMA

The first protocol proposed, called Node Activation Multiple Access (NAMA), activates individual nodes to transmit a single message in each slot. NAMA uses TDMA with time divided into blocks of S_b sections. P_s parts constitute each section and the parts contain T_p time slots. Each node selects a single part, chosen to balance channel utilization across the parts, and contends with the other sensor nodes that select the same part. NAMA reserves the last section of each block for signaling messages that allow sensor nodes to join the network. Each sensor node computes its priority along with the priority of its neighbors and uses these to determine who has access to the current time slot within the sensor node's chosen part. A sensor node gets assigned a particular slot within a section based on its priority. If a sensor node has the highest priority among its two hop neighbors for the given time slot, then the sensor node may transmit. If no sensor node's priority maps to a time slot, then the sensor node with the highest priority may use the time slot.

LAMA

Another protocol, Link Activation Multiple Access (LAMA), activates links to destination sensor nodes based on the Direct Sequence Spread Spectrum (DSSS) code assigned to the receiver and the priority of the transmitter. Each sensor node gets a code assigned from a finite set of pseudo-noise codes. During each time slot the sensor node with the highest priority in a two hop neighborhood, calculated based on sensor node ID as in NAMA, may activate a link by using the code assigned to the receiver. Using orthogonal codes allows

sensor nodes to communicate when they would normally interfere and using the topology information prevents collisions at the receiver.

PAMA

Finally, the Pairwise-link Activation Multiple Access (PAMA) protocol activates links between sensor nodes by assigning priorities to the links and by varying the codes and priorities of links based on the current time slot. A communication link between two sensor nodes, u the source and v the destination, gets activated if the link (u, v) has the highest priority among all links of nodes u and v and node u has the highest priority of its two hop neighbors using the code assigned to link (u, v). Similar to LAMA, the use of DSSS allows nodes to communicate on different codes without interruption and the protocol algorithm prevents collisions on the same code.

Perhaps the largest drawback to the NAMA, LAMA, and PAMA protocols arise from the resources required. All the protocols require a sensor node to compute the priorities of each neighboring sensor node for each time slot. Constantly calculating sensor node priorities may consume energy resources quickly and degrade the network lifetime to unacceptable levels. Additionally, LAMA and PAMA require the sensor nodes have radios with spread spectrum capabilities, which increases sensor node cost. Dynamic slot assignment also prevents sensor nodes from developing a regular sleep schedule since the priorities vary based on the current slot number.

4.2.2 Traffic-Based MAC Protocols

MAC protocols that adapt to network conditions may consume a minimum of energy resources while providing responsive performance since they can operate over a range of conditions. Sensor networks that sporadically generate large volumes of traffic provide the best cases for MAC protocols that modify their operation based on traffic conditions. However, to provide this benefit MAC protocols must estimate and share traffic information with neighbors and utilize resources to maintain a current and correct view of the network state.

TRAMA

The Traffic-Adaptive Medium Access (TRAMA) [35] protocol attempts to balance the benefits of scheduled and unscheduled protocols by providing scheduled slots with no contention for longer data messages and random access slots for small, periodic control messages. Additionally, sensor nodes adapt to traffic and network conditions by sharing traffic needs with neigh-

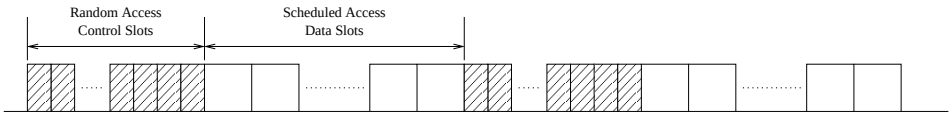


Fig. 8. TRAMA Frame Format

bors and learning the two-hop topology of their neighbors. TRAMA accomplishes all this through the use of three sub-protocols: the Neighbor Protocol (NP), which shares the topology information; the Schedule Exchange Protocol (SEP), which allows nodes to share what traffic they have queued; and the Adaptive Election Algorithm (AEA), which selects the slots to use for data transfer based on the topology and traffic conditions. Frames within TRAMA consist of several slots, where the random access control slots occur together at the beginning of a frame and the scheduled data slots occur at the end as shown in Figure 8.

To share the topology information, sensor nodes pick a random control slot and transmit a list of their one hop neighbors according to the NP. All sensor nodes receive control messages from neighbors by listening during time slots in which they do not transmit. Using the information collected from neighbors, a sensor node determines the sensor network topology within a two-hop neighborhood. Since collisions may occur for the control messages, the authors describe the number of retransmissions a sensor node should use and the total number of control slots based on the expected number of two-hop neighbors.

The SEP performs a similar function by distributing a sensor node's traffic information among its neighbors through schedule packets and schedule summaries. Sensor nodes append schedule summaries to data packets. Sensor nodes transmit schedule packets during the last slot they own in each frame and include the number of slots the sensor node owns in the next frame as determined by the AEA, a bitmap of the intended receivers, and the data slots the sensor node plans to use. Bitmaps allow the sensor node to decrease the message size and to transmit messages with arbitrary destinations (e.g., one receiver, broadcast, multicast). Schedule summaries provide a backup mechanism to protect against schedule packet loss and include a shorter bitmap that indicates only the slots the sensor node plans to transmit in. In order to further limit the effects of unsynchronized schedules, each sensor node must listen to the last data message of each sensor node in its one-hop neighborhood to get the schedule summary. Note that schedule summaries contain information about the slots remaining in the current frame while schedule packets contain information about the slots in the next frame.

Each sensor node runs the AEA to determine the data slots in which it must sleep, transmit, or receive. To assign data slots, TRAMA defines a node priority as a hash of the sensor node's unique ID and the slot number. The node with the highest priority within a two-hop neighborhood owns the correspond-

ing slot. A sensor node transmits in a slot if it has a message to transmit and it owns the slot. Likewise, a sensor node attempts to receive a message whenever the schedule for the slot owner indicates it will transmit to the sensor node. Otherwise the sensor node sleeps to conserve energy. The authors describe the inconsistency problem, similar to the exposed terminal problem, where two neighboring sensor nodes make a different decision on a slot's owner because of a third node not in the two-hop neighborhood of the first two. In this case a receiver may miss a message because a sensor node it considers the winner does not have data for it while another neighboring node considers itself the slot owner and transmits a message. To account for this the authors propose a sequence of steps that safeguard when a node can safely sleep and also provides a mechanism to arbitrate the ownership of unused slots.

Several advantages arise out of the TRAMA design. First, the scheduled access to the data slots limits message collisions and reduces the total energy the transceiver requires. Providing the random access slots once per frame time allows the protocol to quickly adapt to changes in the local sensor network. To lengthen a sensor node's sleep time, TRAMA groups the data slots a sensor node gives away at the end of the frame. Finally, TRAMA provides a great deal of flexibility to network and traffic conditions by sharing state among the sensor nodes. Sensor nodes minimize the state data shared by appending the information to other messages, as with schedule summaries, and by using smaller message sizes through bitmaps. However, TRAMA has several disadvantages typical of a scheduled protocol. First, by depending on the state information sensor nodes may not operate optimally when inconsistent state develops, which can lead to decreased performance. Some aspects of TRAMA, using schedule summaries and requiring sensor nodes to listen during a transmitters final data slot, attempt to limit state inconsistencies at the cost of increased energy consumption. Secondly, TRAMA utilizes resources more intensely than many other protocols. Sensor nodes must stay awake during the control slot portion of each frame and must listen during the final data slot of each neighbor, which can severely increase the effective duty cycle of a sensor node. Despite grouping data slots so that a sensor node's sleeping slots remain toward the end of the frame, TRAMA does not attempt to make a sensor node's active slots contiguous. This may result in a much higher frequency of state changes, and therefore a higher energy consumption rate, especially for highly utilized networks. Finally, and perhaps most limiting, TRAMA has a higher level of complexity than other MAC protocols. The complexity not only means a higher processor utilization, but TRAMA must maintain large amounts of state on the node (e.g., neighbor lists, schedules) and update that state frequently.

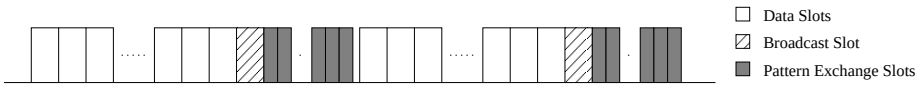


Fig. 9. PMAC Frame Format

PMAC

An alternative approach to scheduling time slots includes the Pattern MAC (PMAC) [36] protocol. Similar to TRAMA, PMAC adjusts its duty cycle based on traffic conditions allowing sensor nodes with more data to utilize more slots than sensor nodes that have no data to transmit. To accomplish this, sensor nodes share their proposed sleep and awake times for the next frame through a pattern sharing procedure. A sensor node can then compare its pattern with its neighbors' patterns to develop the actual schedule it will use. In this way all the sensor nodes can determine their schedules in a distributed manner that allows communication between any neighboring sensor nodes. The pattern a sensor node announces can increase or decrease in activity based on the traffic it has to handle. Figure 9 shows the frame format for the PMAC protocol. Several data slots begin the frame and allow sensor nodes to transfer data messages. A special data frame for broadcast messages occurs after the regular data slots. Finally, PMAC reserves several time slots for pattern exchange between sensor nodes.

A sensor node's pattern consists of a bitmap of time slots during which it plans to sleep (bit cleared) or stay awake (bit set) during the upcoming frame. All sensor node patterns have the format of zero or more sleep slots followed by an active slot. The pattern repeats for the entire frame. To reduce message transmission length, sensor nodes share the minimum amount of information necessary. For example, if a sensor node had a 25% duty cycle it would transmit the pattern 0001. Other sensor nodes would understand to expand this pattern to fill the entire frame, such as 0001000100 for a 10 slot frame. Pattern growth follows a scheme similar to TCP window growth. Sensor nodes start with a pattern of 1, or active for the entire frame time. Every time a sensor node enters an active state it decreases the activity of its pattern. Patterns decrease multiplicatively in activity by doubling the number of sleep periods per active period, up to a bound. So after the first active period the sensor node's pattern would decrease in activity to 01, and after the second active slot it becomes 001. Similarly, the third pattern would decrease in activity to 00001. After reaching the growth bound the pattern increases linearly by adding a single sleep slot. If the protocol has a multiplicative bound of δ, the pattern increases as $0^{\delta}1$, $0^{\delta}01$, $0^{\delta}001$, etc. A sensor node's pattern immediately increases to 1 whenever it has messages to send. Sensor nodes constantly update their pattern based on current conditions, but remain in operation according to the previously shared schedule. The sensor node shares its current pattern in the pattern exchange slots at the end of a frame using CSMA.

At the end of each frame, sensor nodes use several reserved slots to share patterns between neighboring sensor nodes. Each sensor node uses the patterns of its neighbors along with the pattern it generates to calculate the schedule it will follow for the next frame. A schedule consists of one of three possible operations for each slot: transmit, listen, or sleep. A node wakes up and transmits within a slot whenever it has a message for a neighbor and that neighbor advertises a 1 for the slot. A node listens whenever it advertises a 1 for a slot. To conserve energy, a sensor node may wakeup and listen for a short time and return to sleep if it does not detect any activity. Listening for a short time before sleeping prevents the sensor node from missing a message from a neighbor. Finally, if none of the previous conditions hold the sensor node sleeps through the entire slot. Following these rules allows a sensor node to compute the schedule it will follow for the next frame.

Data transmission occurs using CSMA/CA with ACKs providing reliability. To facilitate faster message delivery to sensor nodes with very low activity schedules, every sensor node remains awake for the final data slot in a frame. Broadcast messages could also occur within this slot since all sensor nodes remain active.

PMAC offers a simple way to advertise messages and form schedules between sensor nodes in a neighborhood. The capability to quickly adapt to changing traffic conditions may also make PMAC an attractive choice for a sensor network deployment. However, the schedule generation algorithm has several possible disadvantages. First, some sensor nodes may not receive an updated pattern due to channel errors while others correctly receive the update. This may lead to different schedules present in the same neighborhood and cause collisions, idle listening, and wasted transmissions. Also, the functionality of the protocol relates directly to the traffic intensity. Each time the sensor node operates in an active time slot it performs the pattern update algorithm. During times of high traffic intensity, the processing requirements may become large as the sensor node operates in many active time slots.

4.2.3 Clustering-Based MAC Protocols

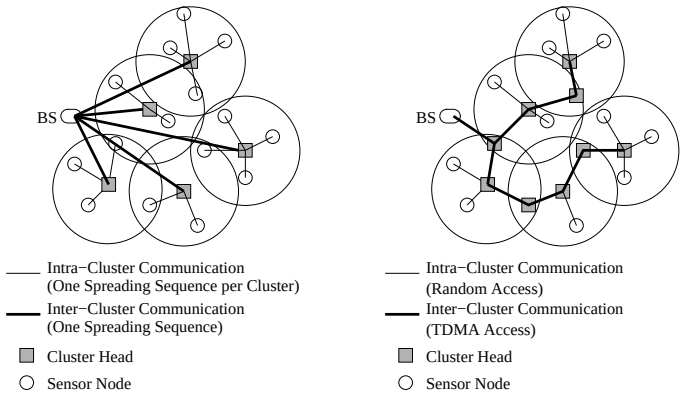
Clustering sensor nodes provides several advantages. First, locally sharing information provides a trade off between global state distribution, which would consume too much energy for the dynamic nature of sensor networks, and greedy algorithms that optimize sensor node behavior independent of other sensor nodes. Clustering also allows protocols to scale more easily since the protocol might view a cluster as a single entity. Second, clustering can differentiate local traffic from global traffic to conserve energy. Data aggregation and sensor node tasking require local traffic, while message forwarding requires traffic to cross cluster boundaries. Lastly, clustering may allow sen-

sor nodes to perform some functionality, such as synchronization, on a local scale that would consume too much energy on a global scale. These benefits, however, come at the cost of coordination message overhead. Cluster heads, those sensor nodes managing clusters, must coordinate the sensor nodes to ensure the cluster reduces energy on average. Protocols often rotate the cluster head functionality among sensor nodes to evenly distribute the additional energy consumption caused by managerial operations. Node dynamics further complicate clustering protocols since cluster formation and cluster head assignment algorithms must adapt to redeployment or sensor node death. Clustering protocol designers must take into account the balance between how often to reform clusters, the extent of cluster reformation, and the energy savings possible from cluster reformation. The following protocols cluster sensor nodes to leverage energy conservation.

LEACH

The Low-Energy Adaptive Clustering Hierarchy (LEACH) [37] protocol provides a MAC protocol along with a clustering algorithm for data gathering sensor networks. To conserve energy, LEACH groups sensor nodes into clusters where a special sensor node, called the cluster head, coordinates the cluster and forwards data generated within the cluster. To equalize the energy consumption throughout the network, the cluster head role rotates among the sensor nodes within a cluster when the current cluster head has lower available energy resources than other sensor nodes. Within each cluster the sensor nodes communicate using direct sequence spread spectrum (DSSS) to limit the interference with other clusters. Each cluster uses a spreading sequence that does not interfere with neighboring clusters and cluster heads use a reserved sequence for communication with the base station. Figure 10(a) diagrams the communication hierarchy in the LEACH protocol.

To form clusters, the sensor nodes transmit a message accepting the cluster head role after a random delay. Sensor nodes select the random delay so that sensor nodes share the cluster head position and consume energy at approximately equal rates. Once a sensor node receives a cluster head announcement, it sends a cluster join message to inform the new cluster head of its membership. Sensor nodes that receive multiple cluster head announcements can select the cluster head that requires the lowest energy for communication. Once a cluster forms the cluster head computes a schedule and distributes it to the sensor nodes it controls. Sensor nodes transmit messages to the cluster head in their time slot and the cluster head transmits the data to the base station. To prevent overloading the communication links to the base station, the authors assume that the cluster heads perform message aggregation so that each cluster produces traffic equivalent to a single sensor node. Communication with the cluster head occurs using CSMA.



(a) LEACH Communication (b) GANGS Communication

Fig. 10. LEACH and GANGS Communication Comparison

Heinzelman et al. also describe a variant, called LEACH-C, which uses the base station to select the optimal cluster heads. During the setup phase of operation each sensor nodes transmits its location and energy levels to the base station. After computing the optimal selection of clusters for energy savings, the base station transmits a list of sensor nodes that will act as cluster heads. Cluster formation then continues similar to LEACH with sensor nodes transmitting join messages and cluster heads setting and distributing schedules.

LEACH possesses several disadvantages in its design. First, it requires a complex radio capable of DSSS and power scaling, which increases the energy consumption and the sensor node cost. Second, cluster formation and restructuring can take a long time during which the sensor nodes consume energy and cannot perform any useful work. Third, LEACH assumes that each sensor node can communicate directly with the base station. Requiring this would either cause sensor nodes to consume large amounts of energy transmitting messages or limit the geographical area a sensor network can cover. The authors address this drawback and suggest forming a routing structure out of the cluster heads or performing a hierarchical structure of clusters. Finally, using LEACH-C, which the authors show can conserve more energy than LEACH, requires nodes that can determine their location. Localization support would increase the node cost and power consumption for either hardware support (e.g., GPS) or protocol support (range estimation algorithms). However, despite the drawbacks of using LEACH in a general sensor network, the authors show that for sensing applications with highly correlated, constantly streamed data LEACH can operate with low energy consumption and extend the network lifetime compared to some other protocols.

GANGS

The GANGS [38] protocol also groups sensor nodes into clusters, but, unlike the LEACH protocol, GANGS uses an unspecified contention protocol for intra-cluster communication and TDMA communication for transmissions between cluster heads. Figure 10(b) illustrates the communication mechanisms in GANGS. GANGS does not assume sensor nodes can communicate with the base station, so the cluster heads must form a routing backbone in the sensor network using a separate routing protocol. GANGS forms clusters in two phases: an initial cluster head election and a secondary process that connects clusters together. During the first phase each sensor node shares its energy resource level with its neighbors. Any node that has more energy resources left that all its neighbors declares itself a cluster head and transmits a message announcing it. During the second phase a non-cluster head sensor node may exist in one of three conditions: it could receive a single cluster head announcement, it could receive multiple announcements, or it could receive no announcements. If a sensor node receives only one announcement, it joins that cluster. For sensor nodes that receive multiple cluster announcements from the same cluster heads, the sensor node with the highest energy resources becomes a new cluster head. Lastly, when a sensor node does not receive any announcement it sends a message to the neighbor with the most energy resources requesting cluster head service and that sensor node becomes a new cluster head. Repeating this process yields a clustered sensor network with connected cluster heads, if such a network exists. As the cluster heads perform their operation they will eventually have lower energy resources than other nearby sensor nodes because of their increased functionality. When this occurs, the sensor nodes perform the cluster formation procedure again so that sensor nodes equalize energy consumption throughout the network.

To assign slots, the cluster heads perform a distributed algorithm that results in each cluster head having a slot to transmit in and knowing the slots used by each neighbor. Each cluster head picks a random number between one and the number of neighbors it has plus one and transmits this number to its neighbors. If two neighboring cluster heads pick the same number they try again by picking an unused number. If no collision occurs, then the cluster head uses the chosen time slot to transmit data. After the cluster heads determine the TDMA schedule, they distribute the information within the cluster so that the other sensor nodes may use the unassigned slots at the end of the frame for sending their data. GANGS assumes a network-wide fixed frame length greater than the maximum expected cluster head connectivity.

Similar to LEACH, GANGS has the disadvantage that cluster formation and restructuring consumes energy resources and takes time. Additionally, the authors do not describe, nor do the LEACH authors, the extent or manner of cluster reformation. When a cluster requires a new cluster head, the authors

provide no indication of the extent of cluster reformation (e.g., the whole network, only one cluster, only nearby clusters) or how the process should occur (cluster head initiated or revocation by another sensor node). These decisions could have drastic impact on the protocol's energy efficiency by affecting the cluster reformation frequency and by causing routing instability. The slot organization in GANGS also introduces wasted resources since not all slots may get used. Within a cluster's frame, the cluster heads use some slots for communication and the sensor nodes in the cluster use the slots after the last slot assigned to a cluster head. However, there may exist multiple unused slots between the slots assigned to cluster heads. Adapting their use for communication between the cluster heads or assigning them for use within the cluster will enable a higher channel utilization for an increase in energy consumption. Despite the disadvantages, the GANGS protocol provides contention-free traffic flow for forwarded traffic while retaining the flexibility and simpleness of a random access protocol within the clusters. Additionally, GANGS requires much fewer computational resources than TRAMA for normal operation and places fewer requirements on the sensor nodes than LEACH, which could allow GANGS to run on smaller and less expensive sensor nodes.

Group TDMA

A third clustering MAC protocol, Group TDMA [39], attempts to limit collisions and provide the highest channel utilization by dividing sensor nodes into groups that can communicate simultaneously. It does this by organizing clusters of sensor nodes, based on topology information, around destination nodes and assigning TDMA slots to different groups of sensor nodes so that collisions between groups do not occur. At each time, a subset of the sensor nodes act as receivers while the rest transmit any data they have during their scheduled slot. By cycling the set of sensor nodes that act as receivers all nodes can communicate. Several aspects of Group TDMA make it different from other protocols examined here. First, Group TDMA organizes the nodes so that communications from different groups do not interfere, but it does not define a specific message exchange protocol. Sensor nodes must also use a traditional MAC protocol to arbitrate which transmitters in a group may transmit to the destination, so Group TDMA may provide support for another MAC protocol or future MAC protocols may incorporate some of the functionality. Also, Group TDMA does not organize sensor nodes into strict clusters, but instead groups them together around receivers, so other protocols that require more conventional clusters cannot leverage Group TDMA operations.

Receiver group formation occurs in a distributed manner based on random timeout values. After waiting a random amount of time, a sensor node transmits a message announcing it will act as a receiver. Sensor nodes within range receive the message and become transmitters. The process continues until all

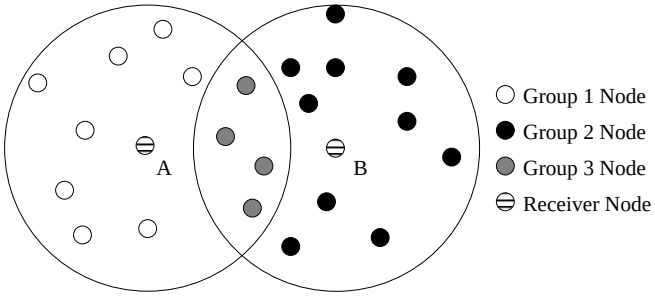


Fig. 11. Group TDMA Receiver-Based Grouping

sensor nodes become transmitters or receivers. The protocol then assigns time slots so that within a slot transmitting groups do not interfere with each other. Consider Figure 11 as an example network. In this case node A and B transmit first and become receivers, and the other nodes become transmitters divided into three groups: Group 1, G_1, can transmit only to node A, Group 2, G_2, can transmit only to node B, and Group 3, G_3, can transmit to either node A or node B. Furthermore, Group TDMA divides Group 3 into two subgroups based on which receiver they have traffic for. Define G_{31} as the subgroup of G_3 with messages for node A and G_{32} as the subgroup with messages for node B. In this case Group TDMA can organize the groups into three slots where G_1 and G_2 transmit during the first slot, G_{31} transmits during the second slot, and G_{32} transmits during the third slot. With this schedule no transmissions from one group will interfere with transmissions from another group. The receiver selection process repeats with different receivers until each sensor node becomes a receiver at least once. As time progresses, the sensor network rotates through the slots for the current receiver group, possibly multiple times, and then switches to a different receiver group.

Sagduyu and Ephremides [39] present methods to determine the throughput optimal slot length assignments given the group organization and traffic distribution, and an energy optimal receiver group activation schedule given the energy resources left in each group and their energy consumption rates. Theoretical analysis also proves the expected group sizes, number of groups, theoretical throughput based on the underlying MAC protocol, and optimal slot length assignments for various network conditions.

To assign TDMA slots to various transmitter groups, the authors present a distributed algorithm that approximates the link coloring problem. After forming a receiver group, sensor nodes that have only one receiver within transmission range form a group and use the first time slot. Nodes that have data for a randomly chosen receiver, call it R_1, but that can communicate with at least one other receiver form another group and use the next time slot. Next, choose a receiver that has at least one transmitter in common with R_1 and call it R_2. Transmitters within range of R_1 and R_2 with data for R_2 form the third group and use the third slot. This process continues until the

protocol forms all the necessary groups. Groups may reuse time slots after proper spatial separation and the authors state that the protocol requires at most 13 different slots.

As mentioned for other scheduled protocols, the setup phase of Group TDMA can consume a large amount of energy and take a significant amount of time. Thus, for highly dynamic sensor networks, Group TDMA may not work well since it would quickly consume energy resources and disrupt traffic forwarding capacity. While the protocol itself does not require extensive processing resources, it does require that sensor nodes maintain the state of receiver group membership and their transmitter group schedule for each receiver group. These schedules and lists could consume large amounts of memory resources. Finally, Group TDMA increases the message latency as a sensor node must queue messages until the next hop enters the active receiver group. This delay, typical of scheduled MAC protocols, will vary depending on the relative receiver schedules and will accumulate at each hop. Group TDMA provides the advantage of dividing the channel in spatial dimensions so that overall channel utilization reaches higher levels than in other protocols. Also, by only activating one set of transmitters for a given destination during each slot, Group TDMA allows sensor nodes to sleep during the slots of other groups if they do not have messages to transmit. Doing this limits the state switches a sensor node must perform and simplifies the schedule.

S-MAC

Ye et al. proposed the Sensor MAC (S-MAC) [40] protocol, perhaps the most studied scheduled MAC protocol for sensor networks, and extended it in further work [15]. Similar to previous protocols, S-MAC clusters sensor nodes, but does so by synchronizing the sleep schedules of neighboring sensor nodes. Thus, S-MAC forms virtual clusters, not strict clusters. Sensor nodes can awake to communicate if necessary while sleeping as much as possible. To transmit messages, sensor nodes use the RTS/CTS scheme during the active portions of the frame as shown in Figure 12(a).

To synchronize, the sensor nodes periodically transmit SYNC messages at the beginning of the active frame time. The SYNC messages allow sensor nodes to learn their neighbors' schedules so they can wake up at the proper time to transmit a message. To improve performance, however, sensor nodes adopt the schedule of their neighbors in several cases. If a node currently does not have a schedule and hears a SYNC message, it adopts the schedule and joins the virtual cluster. If a sensor node hears multiple, sufficiently different schedules, it adopts them all so as to allow communications between different virtual clusters. A sensor node that does not hear any SYNC messages from neighbors chooses its own schedule. In order to detect new schedules sensor

nodes periodically listen for a longer time period that enables them to detect neighboring schedules with high probability. Each sensor node performs a simple contention avoidance algorithm based on a random backoff to limit the number of SYNC message collisions.

Message transfer occurs using the traditional RTS/CTS/DATA/ACK procedure to limit collisions and the hidden terminal problem. As shown in Figure 12(a), sensor nodes transmit the RTS and CTS messages during the active time period, but the data message gets transferred during the inactive period so the uninvolved sensor nodes may sleep. Sensor nodes that overhear an RTS or CTS message for another sensor node can enter the sleep state to conserve energy. To lengthen sleep times and ensure that other sensor nodes do not corrupt a transmission, all sensor nodes perform both physical and virtual carrier sensing. The RTS and CTS messages contain the message transmission time, including time for the ACK message, so that sensor nodes may sleep until the transmission completes. Sensor nodes that wake up with data to send sense the channel for a random time and only transmit if they do not detect any activity.

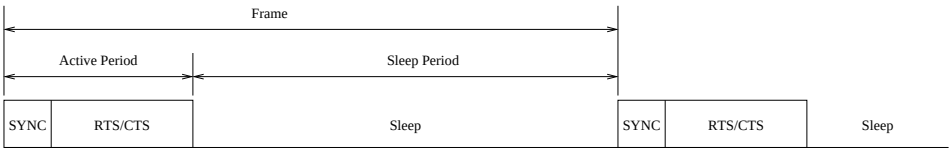
The authors also introduce two improvements to S-MAC [15]. The first attempts to improve on the limitation that sensor nodes may only forward a message over one hop per frame time. To overcome this, the authors introduce the adaptive listening technique, where nodes that overhear a CTS can wake up at the end of the data transmission to possibly act as the next hop. A sensor node that receives a message it must forward attempts to start the message transmission sequence after it sends an ACK to the original transmitter even though the sensor node would normally enter the sleep state according to its schedule. By doing this the sensor nodes may transfer a message across two hops per frame time and decrease the latency. This technique only works within a virtual cluster since sensor nodes outside the cluster likely did not receive the CTS message. S-MAC also introduces a message fragmentation option, called message passing, that allows sensor nodes to transmit larger messages as smaller fragments using a single RTS/CTS exchange. Thus, if one fragment becomes corrupt due to collision or channel error, the sensor node only has to retransmit the small fragment instead of the entire data message.

S-MAC offers several advantages for use in sensor networks. First, loosely synchronizing sensor nodes minimizes the problem of coordinating sensor nodes for communication and may provide adequate synchronization and clustering functionality for other protocols. Sharing beacon generation functionality also distributes this energy drain evenly throughout the network. Second, the protocol requires few processing resources beyond the most basic MAC protocols. Schedule and synchronization maintenance can occur quickly each beacon interval. S-MAC also requires moderate resources, such as memory for schedule offsets and timers for wakeup. Lastly, S-MAC can scale easily since the sensor

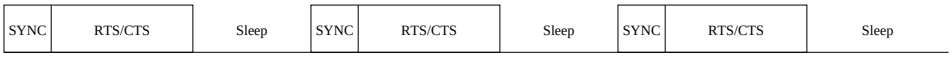
nodes do not require any wide-scale coordination. S-MAC only coordinates neighbors using beacon messages, so sensor nodes do not have to forward or share large amounts of state information. S-MAC, however, does have some disadvantages, some of which researchers have attempted to solve in the protocols of the next section. First, sensor nodes may adopt several schedules, which effectively multiples the duty cycle of the sensor node. The authors reduce the number of sensor nodes that adopt multiple schedules, but can not remove the possibility without segmenting the sensor network. As the lifetime of the sensor network progresses these nodes may die faster and cause segmentation along the borders of the virtual clusters. A second disadvantage comes from the the static duty cycle of S-MAC. Sensor nodes may not change their duty cycle based on traffic or density conditions, and thus can consume more energy than required or limit the protocol's performance. End users may set the duty cycle based on expected application requirements, but S-MAC does not adapt to changing conditions. Lastly, S-MAC does not attempt to control virtual cluster size throughout the network. Varying cluster sizes have several impacts on the protocol's performance. Large clusters reduce the number of sensor nodes that must participate in multiple schedules, but increases the message latency. S-MAC does not provide the user with the ability to control virtual cluster size. The following protocols attempt to improve S-MAC while utilizing the benefits provided by the protocol.

S-MAC Variants

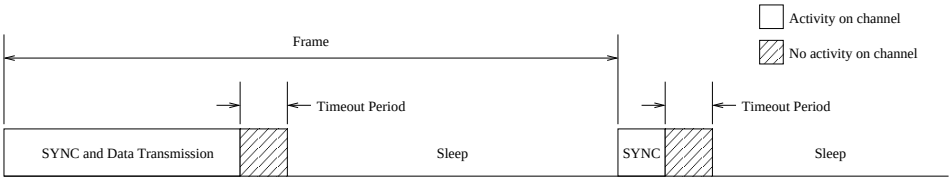
Researchers have proposed several extensions to the S-MAC protocol. The DSMAC [41] protocol extends S-MAC by allowing sensor nodes to adopt dynamic duty cycles based on traffic and energy considerations. Utilizing added fields in SYNC and data messages allows sensor nodes to increase their duty cycle when the per-hop data delay becomes too large and decrease the duty cycle if traffic conditions return to low levels. In DSMAC, sensor nodes include their duty cycle in any SYNC messages they transmit. To estimate the traffic conditions present in the network, each source calculates the queueing delay, from message reception to transmission completion, for each message and adds this to an extra field in future data messages. Additional bounds in the protocol place a limit on the energy consumption rate for a sensor node by limiting how high a sensor node's duty cycle may reach. While the added ability to adapt to traffic conditions lowers the average message latency, it increases the average energy consumption due to the higher fraction of time spent in an active state. To ensure that sensor nodes within the same virtual cluster remain synchronized, any increases to the duty cycle occur as multiplicative powers of 2. Thus, sensor nodes operating a high duty cycle can still receive any SYNC messages sent by sensor nodes operating at a low duty cycle. Figure 12(b) shows a DSMAC frame where the sensor node has a duty cycle twice the normal value.



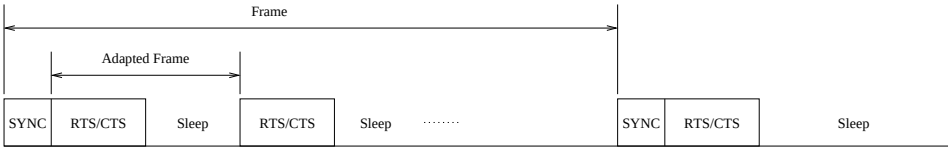
(a) S-MAC Frame Format



(b) DSMAC Frame Format



(c) TMAC Frame Format



(d) ACMAC Frame Format

Fig. 12. S-MAC and Variants Frame Format

The T-MAC [42] protocol also extends S-MAC by using a timer to indicate the end of the active period instead of relying on a fixed duty cycle schedule. Figure 12(c) shows a T-MAC frame where the first active period has the sensor node involved in a message transmission and the second active period has only a SYNC transmission. By adaptively ending the active period, T-MAC nodes may save energy by lowering the amount of time they spend idle listening and also adapt to changes in traffic conditions. In addition, the authors propose two improvements that can decrease the latency of messages and provide a simple form of flow control. To improve message latency, the authors introduce a future request to send message (FRTS) that sensor nodes can use to inform the next hop that it has a future message transfer. The FRTS messages attempt to solve the same problem addressed by the adaptive listening technique of S-MAC. The authors also introduce the message to solve the early sleeping problem that limited the number of hops a message could travel in each frame time. T-MAC also considers the buffer size of the sensor node when calculating the contention period. Sensor nodes that have a full buffer may take priority and control the channel by immediately sending an RTS message after receiving an RTS message from another sensor node. In this way sensor nodes can utilize a simple flow control mechanism and limit buffer overflow by giving sensor nodes with no room to receive a message a higher chance at

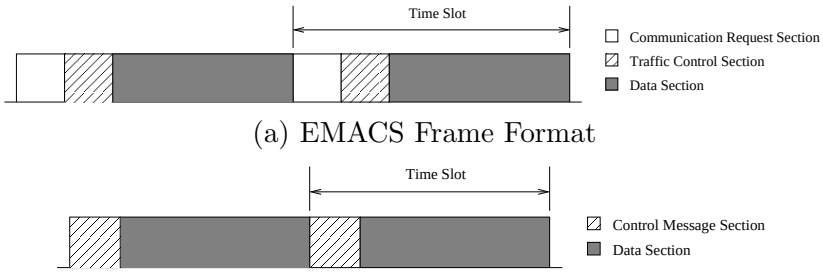
transmitting their queued messages.

Ai et al. provided an alternative approach to improve S-MAC by adding an adaptive duty cycle in the AC-MAC [43] protocol. Instead of modifying the active and sleep time period lengths, AC-MAC allows sensor nodes that have queued messages to introduce multiple data exchange periods per SYNC frame as shown in Figure 12(d). The first sensor node to transmit an RTS message sets the duty cycle used within the SYNC frame. Within the first RTS message of a SYNC frame, the transmitting sensor node includes a value proportional to its used buffer capacity. Sensor nodes that receive this RTS message can then calculate the duty cycle to use within the virtual cluster for the current SYNC period. In order to provide sensor nodes with many buffered messages a priority, each sensor node calculates its random backoff value from a contention window whose size varies inversely proportional to the amount of traffic it has buffered. To simplify the protocol, sensor nodes only adopt one schedule per SYNC period.

A final proposal to improve S-MAC comes from the MS-MAC [44] protocol that focuses on improving performance within mobile sensor networks. To decrease the time a sensor node needs to join a virtual cluster, a sensor node increases the rate at which it checks for new schedules depending on the estimated movement around the sensor node. To estimate movement, each sensor node records received signal strength values for each neighbor and uses any changes as indications of sensor node movement. Within each SYNC message a sensor node lists the maximum speed it estimates among its neighbors. Nodes with a high mobility, or sensor nodes around a highly mobile sensor node, look for additional schedules much more frequently and adopt schedules with a lower latency. MS-MAC therefore trades energy consumption for faster schedule synchronization.

4.2.4 TDMA MAC Protocols

TDMA provides a tempting solution for sensor network MAC protocols because reducing collisions and idle listening can save considerable amounts of energy. Fairness and simple traffic engineering also become possible with a TDMA-based protocol. Several complications arise, however, when designing TDMA protocols for sensor networks. Time slot assignment becomes difficult because sensor nodes can not coordinate on large scales without introducing large overhead. Synchronization functionality must exist to correct timing errors caused by clock drift within each sensor node. Strict TDMA protocols also suffer from utilization problems during periods of light traffic generation. The following protocols demonstrate how researchers have attempted to apply TDMA techniques to sensor networks.



(a) EMACS Frame Format

(b) LMAC Frame Format

Fig. 13. EMACS and LMAC Frame Formats

EMACS, LMAC, and AI-LMAC

A family of TDMA MAC protocols for sensor networks includes EMACS [45], LMAC [46], and AI-LMAC [47]. They share many similarities, but require sensor nodes to interact differently. All the protocols divide each time slot into sections that serve a particular purpose. Slot assignment among the sensor nodes occurs in identical ways by sensor nodes picking a random slot not controlled by a neighboring sensor node. Each sensor node transmits a control message during any time slot it owns. In this way sensor nodes can maintain loose synchronization and notify neighbors of forthcoming data transmissions. Figure 13 shows the time slot formats for the EMACS and LMAC protocols (AI-LMAC uses the same slot structure as LMAC).

To start the time slot ownership, the base station takes control a time slot by transmitting a control message. Neighboring sensor nodes then randomly pick a slot to own and begin transmitting during that time slot. If collisions occur, neighboring sensor nodes indicate this within the control message they transmit during their time slot. Slot ownership propagates through the sensor network with sensor nodes reusing slots at non-interfering distances.

EMACS has three sections within each time slot, as shown in Figure 13(a): communication request, traffic control, and data. Sensor nodes use the traffic control section to transmit their periodic control information. Every sensor node must transmit this information during their time slot and neighboring sensor nodes listen for the control packet of neighbors. A sensor node may request to use the data section of a time slot it does not own by transmitting a request during the communication request section. The time slot owner can give ownership to the requesting sensor node within its control message. All data transmissions occur within the data section.

Sensor nodes within the network using EMACS operate in one of three possible modes. Active nodes co-operate fully in the communications, own a slot, and transmit a control message within each slot they own. Passive sensor nodes do not own a slot and only transmit messages after requesting a slot from an active sensor node. Finally, dormant sensor nodes do not participate in

the sensor network and sleep until they wish to participate in an active or passive role. Providing varying levels of functionality allows the sensor nodes to conserve energy when the application does not need them and activate only the minimum number of sensor nodes to perform the application functionality.

LMAC differs from EMACS most fundamentally by having all sensor nodes own a slot—all sensor nodes operate in an active state. Since all sensor nodes own a slot the communication request section becomes unnecessary and the LMAC time slot does not include it, as shown in Figure 13(b). LMAC also includes a simple hop count-based routing protocol that allows sensor nodes to send messages to the base station.

The simple method used to assign time slots to sensor nodes in EMACS and LMAC seems attractive for very limited devices, but also produces several disadvantages. First, network setup may take considerable time for large deployments since the process starts at the base station and slot collisions may take several frames to resolve. Second, sensor nodes expend large amounts of overhead in slot maintenance by transmitting in every slot they own and listening during the control portion of each slot owned by neighboring sensor nodes. Lastly, sensor nodes can not adapt to traffic conditions by varying the slot ownerships. AI-LMAC attempts to solve several of these disadvantages.

The AI-LMAC protocol extends upon LMAC by varying the number of slots a sensor node owns based on traffic conditions within an environmental monitoring application. To measure traffic conditions, each sensor node maintains a Data Distribution Table (DDT) that records simple statistics on the data generated and forwarded by a node, such as values, originating node, and previous hop. AI-LMAC groups sensor nodes into a parent-child hierarchy. Based on information within the DDT, parents may suggest that a child take control of a greater or fewer number of time slots. Suggestions from the parent sensor node ensures that the assignment meets two conditions: fairness of slot assignment among siblings and ensuring aggregate child bandwidth does not overload the parent sensor node. To conserve energy, a sensor node only transmits a control message in the first time slot it owns within a frame. Within the control message the sensor node includes the time slots it owns and indicates any data messages it plans to transmit during the current frame. AI-LMAC control messages also provides data message acknowledgments not provided in LMAC.

AI-LMAC improves upon LMAC by offering adaptability to traffic conditions and reducing slot maintenance overhead. However, it still has some limitations. The overhead required for the Data Distribution Tables may quickly become large, reducing the already limited available memory for other protocols and applications. DDT maintenance may also consume computational and energy resources as sensor nodes frequently update values based on recent data.

Z-MAC

Researchers propose a more flexible approach with the Zebra-MAC (Z-MAC) protocol [48]. Similar to the previous TDMA-based protocols, Z-MAC assigns sensor nodes a time slot, but easily allows sensor nodes to utilize slots they do not own through CSMA with prioritized backoff times. This provides Z-MAC with the capability to perform similar to CSMA when applications generate less traffic, but approximates a strict TDMA scheme when traffic requirements increase.

Prior to sensor network operations, a distributed slot assignment protocol [49] provides sensor nodes with the time slots they may utilize for transmission. The schedule ensures that two-hop neighbors do not get assigned the same slot number. The authors further introduce a time frame rule that allows sensor nodes to utilize slots not assigned within the two-hop neighborhood and removes the need in some cases to run the slot assignment protocol when the network topology changes slightly. Running a slot assignment protocol introduces a large overhead during network setup, but decreases the energy expended for communications during the sensor network's lifetime. Sensor nodes must also incur this overhead when a significant number of nodes move or get deployed, but not for the more common case of varying transceiver coverage.

During each time slot sensor nodes use CSMA to determine who may transmit. However, Z-MAC gives the slot owner preference in channel access by increasing the initial backoff time for sensor nodes that do not own the slot. The owner of the current slot selects a random backoff time of up to T_o and performs CSMA. Using a random backoff for the slot owner limits the effect of incorrect synchronization among neighboring sensor nodes. Sensor nodes that do not own the current slot select a backoff time between T_o and T_{no}, where $T_{no} > T_o$, and perform CSMA. Sensor nodes receive messages according to the B-MAC protocol and maintain a receive schedule independent of the time slots.

Z-MAC also uses explicit congestion notification (ECN) messages to limit the effect of hidden terminals during periods of high contention. When a sensor node detects high contention it transmits an ECN message to the neighbor it has a message for. The neighbor broadcasts the ECN message to its neighbors, all of whom enter a high contention level (HCL) state. Sensor nodes return to a low contention level (LCL) state after a time period if they do not receive further ECN messages. While in the HCL state, a sensor node only attempts to transmit in its slot and those of its immediate neighbors, thus reducing contention between neighbors two hops apart. Sensor nodes detect contention by tracking the amount of time they spend in backoff caused by failed carrier sensing. When the time spent in backoff reaches a threshold, the sensor node

transmits a ECN message.

Perhaps Z-MAC's greatest advantage comes from its easy and rapid adaptability to traffic conditions. Approximating a CSMA protocol under light traffic conditions and a TDMA protocol under heavy traffic conditions can save large amounts of energy. Further benefits come from Z-MAC's robustness against synchronization errors. Compared to other protocols, Z-MAC requires few processing and memory resources. These benefits come at the cost of protocol overhead, primarily caused by the TDMA structure. First, developing a TDMA schedule for the sensor nodes consumes time and energy during network setup. Z-MAC increases the amount of change required to force a schedule recalculation, but for any significant change the network must perform the costly procedure again. Second, similar to any TDMA protocol, sensor nodes must consume resources to maintain synchronization. Third, Z-MAC has similar disadvantages—and advantages—to B-MAC since it uses the underlying communication mechanisms from B-MAC. Lastly, using ECN messages can reduce contention within a local area, but places a burden on an already busy network. In sensor network that generate large volumes of local traffic based on some event, Z-MAC will take time to distribute ECN messages as it transitions toward TDMA operation.

4.2.5 Scheduled MAC Protocol Summary

In this section, we presented several scheduled MAC protocols proposed for sensor networks. Many provide the capability to lower energy consumption by reducing collisions, limiting idle listening, and providing functionality for other protocols, but they require that sensor nodes expend energy to share state and maintain synchronization. Additionally, the extent and frequency to which the sensor network undergoes organization and reorganization can greatly affect its performance. However, scheduled MAC protocols may allow sensor nodes to remain asleep for longer periods of time and forward messages with less effort than those using unscheduled MAC protocols since the sensor node has some indication of its neighbor's plans. Table 2 provides a summary of the MAC protocols in this section.

5 Future Outlook

Many directions exist for future work in the area of sensor network MAC protocols. One direction currently under study combines the operation of the MAC protocol with other layers, using cross-layer or combined-layer designs, to increase performance. Sharing information between protocol layers may allow the protocols to cooperate and limit the resources needed for operation.

Protocol Type	**Summary**	**Advantages**	**Disadvantages**
Priority-Based	Slot ownership based on priority of node or link	Only local knowledge required for channel access decision	Computational requirements and sleeping schedule variability
Traffic-Based	Schedule communications with neighbors based on traffic	Activity adaptive to traffic requirements	Schedule sharing or computation and memory requirements for schedules
Clustering-Based	Organize sensor nodes into clusters	Local coordination for energy conservation	Energy resources to form and maintain clusters
Slotted TDMA	Sensor nodes control a set of slots for communication	High utilization under high load; loose synchronization provided (LMAC); adaptive to light load (Z-MAC)	Slot maintenance and synchronization overhead

Table 2
Scheduled MAC Protocol Summary

Examples include sharing MAC layer resources with the routing layer [50], the physical layer [51], or the application [52]. A cooperative scheduled MAC and proactive routing layer could use a single message to share any necessary state among sensor nodes and distribute the routing information. By combining the state maintenance messages together the sensor node can decrease the amount of energy spent handling control messages. IEEE 802.15.4 provides a limited form of this by allowing beacon messages to contain a payload from the network layer. Additionally, the MAC protocol can share link status information with the routing protocol in order to choose the best route based on more information than the network topology. Furthermore, consider a sensor network that generates various traffic types, some that require a low latency and high reliability and messages that the network can delay or drop. If the application shares a description of the data in a message, the MAC layer can use ACKs and priorities to provide the best benefit for a given cost. While a cross-layer design has many advantages it suffers from the known drawbacks of limited generality and interoperability. A MAC protocol that requires state shared by another protocol, say the routing protocol, can not operate unless the user chooses a routing protocol that shares that information. In traditional networks where the devices do not have such stringent energy and computation constraints, the efficiency benefits of a cross-layer design do not outweigh the

interoperability problems. However, in sensor networks the need to leverage every advantage and the unique requirements associated with every application makes cross-layer designs very tempting.

To conserve energy further, sensor network MAC protocols should adapt to changes both in network topology and traffic characteristics. A MAC protocol that operates well when the sensor network has light traffic, but does not adapt to changing traffic patterns may become inefficient. Without adaptation the sensor nodes may consume more energy than necessary and decrease the usefulness or lifetime of the sensor network. However, adaptation often includes complexity, which brings other disadvantages. As the MAC protocol grows to encompass various scenarios it grows more complex, especially if the MAC protocol changes in drastic ways. All the complexity increases the processing and memory resources required on the sensor nodes, and thus increases sensor node cost. The granularity of change also affects the complexity of an adaptive MAC protocol. A MAC protocol with many possible settings and operating points can operate more efficiently than a MAC protocol with only a few options. Researchers have proposed adaptive MAC protocols, but most change the protocol in small ways. Throughout the operational lifetime of a sensor network the topology will change. Sensor node movement, energy depletion, sensor node redeployment, and the changing physical environment all cause the MAC protocol to detect and communicate with different sensor nodes. While all sensor network MAC protocols must adapt to these changes, the rate at which they do it affects performance.

Further improvements in energy conservation may come with the help of more advanced hardware. A transceiver that provides the MAC layer with the ability to control aspects of low-level communications allows the MAC protocol to adapt to changes in the physical environment. A sensor node that wishes to transmit a message to a nearby recipient could decrease the power used for that transmission. MAC protocols may produce further savings if communicating sensor nodes can cooperate and change the modulation scheme used [22]. Nearby nodes could use a modulation scheme that provides a higher data rate for the same bit error rate, while nodes further apart could use a lower data rate modulation scheme more resistant to channel noise. Similar to other energy saving ideas, however, adding more complicated hardware requires a more complex MAC protocol and increases the cost of the sensor nodes.

Normally, MAC protocol design does not consider flow control. However, since the sensor nodes poses such limited resources, the MAC protocol may take action to ensure that message recipients have enough memory to store the intended message. This layer of protection would decrease the amount of messages lost to buffer overflow and could improve overall network performance by limiting the effect of bottlenecks in the network. Providing this function-

ality would require somehow sharing sensor node resource information with neighbors. The granularity and scope of the information sharing, along with how to distribute the information, provide future research possibilities.

6 Conclusion

Much research has considered MAC protocols for wireless networks in various contexts. Unfortunately, the direct application of previous protocols does not satisfy sensor network requirements since the original protocols do not consider the finite energy resources available. Recently, much research has focused on how to apply these techniques to the resource limited devices in sensor networks. This paper has covered many MAC protocols proposed thus far for sensor networks, but many more exist. Each protocol provides benefits for certain applications or under certain conditions based on the chosen design. It remains an open question, and one of great interest, if a general, flexible MAC protocol exists that supports various applications and operating environments while consuming minimal power and offering acceptable traffic characteristics.

Acknowledgements

We wish to thank Mehmet Can Vuran and the anonymous reviewers for helpful suggestions that improved this paper.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, Computer Networks 38 (4) (2002) 393–422.

[2] S. Roundy, P. K. Wright, J. M. Rabaey, Energy Scavenging for Wireless Sensor Networks with Special Focus on Vibrations, Kluwer Academic Publishers, 2004.

[3] J. F. Kurose, K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition, Addison Wesley, 2005.

[4] C. Karlof, N. Sastry, D. Wagner, TinySec: A link layer security architecture for wireless sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 162–175.

[5] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, in: Proceedings of the International

Conference on Mobile Computing and Networking (MobiCom), 1999, pp. 263–270.

[6] J. Zhao, R. Govindan, Understanding packet delivery performance in dense wireless sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 1–13.

[7] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 14–27.

[8] I. Demirkol, C. Ersoy, F. Alagöz, MAC protocols for wireless sensor networks: A survey, IEEE Communications Magazine 44 (4) (2006) 115–121.

[9] G. Halkes, T. V. Dam, K. Langendoen, Comparing energy-saving MAC protocols for wireless sensor networks, Mobile Networks and Applications 10 (5) (2005) 783–791.

[10] L. Kleinrock, F. A. Tobagi, Packet switching in radio channels: Part I—carrier sense multiple-access modes and their throughput-delay characteristics, IEEE Transactions on Communications 23 (12) (1975) 1400–1416.

[11] P. Karn, MACA – a new channel access method for packet radio, in: Proceedings of the ARRL Computer Networking Conference, 1990.

[12] V. Bharghavan, A. Demers, S. Shenker, L. Zhang, MACAW: A medium access protocol for wireless LANs, in: Proceedings of the Conference on Communications Architectures, Protocols and Applications, 1994, pp. 212–225.

[13] F. Talucci, M. Gerla, MACA-BI (MACA by invitation) a wireless MAC protocol for high speed ad hoc networking, in: Proceedings of the IEEE International Conference on Universal Personal Communications, Vol. 2, 1997, pp. 913–917.

[14] IEEE, IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999).

[15] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking 12 (3) (2004) 493–506.

[16] IEEE, IEEE Standard 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) (2003).

[17] Zigbee Alliance Home Page, http://www.zigbee.org.

[18] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, W. Dehaene, Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives, in: Proceedings of Design, Automation and Test in Europe (DATE), Vol. 1, 2005, pp. 169–201.

[19] Crossbow Technology, Inc. Home Page, http://www.xbow.com.

[20] A. Woo, D. E. Culler, A transmission control scheme for media access in sensor networks, in: Proceedings of the International Conference on Mobile Computing and Networking (MobiCom), 2001, pp. 221–235.

[21] S. Ci, H. Sharif, K. Nuli, Study of an adaptive frame size predictor to enhance energy conservation in wireless sensor networks, IEEE Journal on Selected Areas in Communications 23 (2) (2005) 283–292.

[22] S. Cui, A. J. Goldsmith, A. Bahai, Energy-constrained modulation optimization, IEEE Transactions on Wireless Communications 4 (5) (2005) 2349–2360.

[23] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, in: Proceedings of the International Conference on Mobile Computing and Networking (MobiCom), 2001, pp. 272–287.

[24] S. Singh, C. Raghavendra, PAMAS – power aware multi-access protocol with signaling for ad hoc networks, SIGCOMM Computer Communications Review 28 (3) (1998) 5–26.

[25] F. A. Tobagi, L. Kleinrock, Packet switching in radio channels: Part II—the hidden terminal problem in carrier sense multiple-access and the busy-tone solution, IEEE Transactions on Communications 23 (12) (1975) 1417–1433.

[26] I. Chatzigiannakis, A. Kinalis, S. Nikoletseas, Wireless sensor networks protocols for efficient collision avoidance in multi-path data propagation, in: Proceedings of the ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, 2004, pp. 8–16.

[27] I. Chatzigiannakis, T. Dimitriou, M. Mavronicolas, S. Nikoletseas, P. Spirakis, A comparative study of protocols for efficient data propagation in smart dust networks, Parallel Processing Letters 13 (4) (2003) 615–627.

[28] M. C. Vuran, I. F. Akyildiz, Spatial correlation-based collaborative medium access control in wireless sensor networks, IEEE/ACM Transactions on Networking 14 (2) (2006) 316–329.

[29] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Optimizing sensor networks in the energy-latency-density design space, IEEE Transactions on Mobile Computing 1 (1) (2002) 70–80.

[30] E.-Y. A. Lin, J. M. Rabaey, A. Wolisz, Power-efficient rendez-vous schemes for dense wireless sensor networks, in: Proceedings of the IEEE International Conference on Communications (ICC), Vol. 7, 2004, pp. 3769–3776.

[31] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 95–107.

[32] A. El-Hoiydi, J.-D. Decotignie, WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks, in: Proceedings of the International

Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors), 2004, pp. 18–31.

[33] S. Mahlknecht, M. Böck, CMSA-MPS: A minimum preamble sampling MAC protocol for low power wireless sensor networks, in: Proceedings of the IEEE International Workshop on Factory Communication Systems, 2004, pp. 73–80.

[34] L. Bao, J. Garcia-Luna-Aceves, A new approach to channel access scheduling for ad hoc networks, in: Proceedings of the International Conference on Mobile Computing and Networking (MobiCom), 2001, pp. 210–221.

[35] V. Rajendran, K. Obraczka, J. Garcia-Luna-Aceves, Energy-efficient, collision-free medium access control for wireless sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 181–192.

[36] T. Zheng, S. Radhakrishnan, V. Sarangan, PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the IEEE International Parallel and Distributed Processing Symposium, 2005, pp. 65–72.

[37] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670.

[38] S. Biaz, Y. D. Barowski, GANGS: An energy efficient MAC protocol for sensor networks, in: Proceedings of the Annual Southeast Regional Conference, 2004, pp. 82–87.

[39] Y. E. Sagduyu, A. Ephremides, The problem of medium access control in wireless sensor networks, IEEE Wireless Communications 11 (6) (2004) 44–53.

[40] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (InfoCom), Vol. 3, 2002, pp. 214–226.

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

[42] T. van Dam, K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 171–180.

[43] J. Ai, J. Kong, D. Turgut, An adaptive coordinated medium access control for wireless sensor networks, in: Proceedings of the International Symposium on Computers and Communications, Vol. 1, 2004, pp. 214–219.

[44] H. Pham, S. Jha, An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC), in: Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2004, pp. 214–226.

[45] L. van Hoesel, P. Havinga, Poster abstract: A TDMA-based MAC protocol for WSNs, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 303–304.

[46] L. van Hoesel, P. Havinga, A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches, in: Proceedings of the International Conference on Networked Sensing Systems (INSS), 2004.

[47] S. Chatterjea, L. van Hoesel, P. Havinga, AI-LMAC: An adaptive, information-centric and lightweight MAC protocol for wireless sensor networks, in: Proceedings of the Intelligent Sensors, Sensor Networks, and Information Processing Conference, 2004, pp. 381–388.

[48] I. Rhee, A. Warrier, M. Aia, J. Min, Z-MAC: A hybrid MAC for wireless sensor networks, in: Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2005, pp. 90–101.

[49] I. Rhee, A. Warrier, J. Min, L. Xu, DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks, in: Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2006, pp. 190–201.

[50] M. Zorzi, A new contention-based MAC protocol for geographic forwarding in ad hoc and sensor networks, in: Proceedings of the IEEE International Conference on Communications (ICC), Vol. 6, 2004, pp. 3481–3485.

[51] Y. Chen, Q. Zhao, Distributed transmission protocol for lifetime maximization in sensor networks, in: Proceedings of the Signal Processing Advances in Wireless Communications Workshop, 2005, pp. 895–899.

[52] T. Stathopoulos, R. Kapur, D. Estrin, J. Heidemann, L. Zhang, Application-based collision avoidance in wireless sensor networks, in: Proceedings of the IEEE International Conference on Local Computer Networks, 2004, pp. 506–514.