Medium Access in Ad Hoc and Sensor Networks

Vojislav B. Mišić and Jelena Mišić, *University of Manitoba, Winnipeg, Canada*

INTRODUCTION

Wireless ad hoc networks are a category of wireless networks that utilize multihop packet relaying yet are capable of operating without any infrastructure support (Perkins 2001; Ram Murthy and Manoj 2004; Toh 2002). Applications that necessitate ad hoc networking capabilities include mobile, collaborative, and distributed computing; mobile access to the Internet; wireless mesh networks; military applications; emergency response networks; and others. Ad hoc networks are formed by a number of devices, possibly heterogeneous, with wireless communication capabilities that connect and disconnect at will. In addition, some of these devices may be mobile and are thus able to change their location frequently. Ad hoc networks with mobile nodes are often referred to as *mobile ad hoc networks* (MANETs). Even without mobility, nodes can join or leave an ad hoc network at will, and such networks need to possess self-organizing capability in terms of media access, routing, and other networking functions. As such, the design and deployment of wireless ad hoc networks presents several challenges that do not exist, or exist in rather different forms, in traditional wired networks. Some of the most important challenges are as follows:

- *Self-organization, adaptability, and self-healing*. The trademark feature of ad hoc networks is the ability of individual nodes to attach to and detach from such networks at will and in the absence of any fixed infrastructure. Therefore, such networks need protocols that can support and facilitate the tasks of topology construction, reconfiguration, and maintenance, as well as routing, traffic monitoring, and admission control. Note that the above constraint does not mean that an infrastructure, if present, cannot or should not be used; it just means that

the network ought to be able to function with or without such infrastructure. Furthermore, as sudden departures or even failures of individual nodes are to be expected in many applications, the network should possess self-healing capabilities so as to continue to function.

Scalability of the network refers to its ability to retain certain performance parameters regardless of large changes in the number of nodes deployed in that network. Scalability is an important aspect of ad hoc networks, and it is closely related to the self-organizing property. Scalability is highly dependent on the amount of overhead in terms of bandwidth and power expenditure needed to exchange control packets at various layers (*medium access control*, or MAC; routing and transport, etc.) of network functionality. It is also affected by the manner in which the network is organized, as will be seen in subsequent discussion.

Delay considerations are of crucial importance in certain types of applications—for example, in military applications such as battlefield communications or detection and monitoring of troop movements, or in health care applications where patients with serious and urgent medical conditions must be continuously monitored for important health variables via ECG, EEG, or other probes. Low delays can be achieved by bandwidth reservation requested by the source device through some kind of admission control that will monitor and prevent network congestion or by some kind of scheduling. In the latter two cases, control is exerted by some device that performs the role of network coordinator or base station. (Such a role can be, and often is, temporary.)

We stress that providing prescribed delay bounds is a nontrivial issue in traditional wired networks. In a network with nonstationary topology formed by mobile, resource-constrained nodes, maintaining the delays within prescribed bounds is even more complex. As a minimum, we might just try to reduce the delays—but this is not an easy task either. Delay minimization is often hampered by the instability of the network topology as well as by fluctuations in traffic characteristics. As a result, any minimum that may be achieved is likely to be of an ephemeral character, and constant monitoring and minimization of delays is necessary.

- *Throughput*. In some applications, the most important performance target is throughput rather than delay. Such is the case in several collaborative, distributed computing applications and in mobile access to the Internet, which might include significant amounts of multimedia traffic. At the physical (PHY) level, throughput may be impaired by packet errors caused by noise and interference. At the MAC level, throughput may be impaired by collisions if a contention-based medium access mechanism is used, or by unfairness if bandwidth reservation- or scheduling-based access mechanism is used. (Detailed descriptions of these mechanisms can be found below.) Cross-layer optimization that accounts for those effects—preferably, all of them—may be needed to achieve high throughput.
- *Packet and data losses*. Loss of information is not tolerated in ad hoc networks, and active measures to restore reliability of data transfers must be undertaken at both the MAC and upper layers.
- *Fairness*. In most cases where throughput is the most important performance indicator, fairness among different nodes or users is also of importance. Again, the instability of topology and the nonstationary traffic characteristics tend to make this problem much more difficult in wireless ad hoc networks than in traditional wired networks.
- *Power management*. Some of the nodes in an ad hoc network might operate on battery power, in which case power-management functions become necessary. Although energy efficiency is a desirable feature in general, it is seldom a crucial issue in ad hoc networks. The power source either has sufficient capacity (e.g., a car battery can be expected to provide ample medium access in ad hoc and sensor networks capacity to operate a laptop) or may be recharged as needed (e.g., when using a PDA device at home or in the office). Cases in which the minimization of energy consumption becomes the main limiting parameter for a wireless device are not too common.
- *Low maintenance*. Finally, all maintenance tasks in ad hoc networks should be automated or (at worst) be simple enough to be undertaken by nonspecialist human operators such as owners of laptop computers and PDAs. This requirement might be considered as an extension (or perhaps generalization) of the requirement for self-organizing capabilities mentioned above.

MAC Design Goals in Ad Hoc Networks

The MAC protocol is that part of the overall network functionality that deals with problems of achieving efficient, fair, and dependable access to the medium shared by many different devices (Stallings 2002). The role of the MAC protocol is particularly important in wireless networks that differ from their wired counterparts in many aspects. The most important among those differences stem from the very nature of the wireless communication medium, where two devices need not be explicitly connected to be able to communicate; instead, it merely suffices that they are within the radio transmission range of each other.

For example, when two or more packets are simultaneously received, the receiver may encounter problems. At best, the unwanted packets are treated as noise that impairs the reception of the packet intended to be received but can be filtered out. At worst, the correct packet may be damaged beyond repair and the receiver may be unable to make any sense out of it; this condition is referred to as a *collision*. Collisions waste both network bandwidth and power resources of individual devices, transmitters, and receivers alike, and active measures should be taken to reduce the likelihood of their occurrence.

Common approaches for collision minimization in wired networks include detection and avoidance. Collision detection is widely used in wired networks, where it involves the simple act of listening while transmitting. However, this is not feasible in wireless communication, where few devices are equipped with the required capability

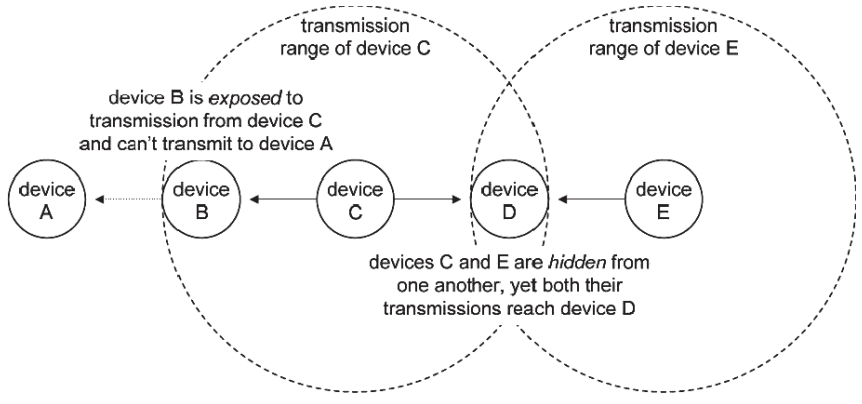

Figure 1: Hidden and exposed terminal problem

(Stallings 2002). Furthermore, packet collisions in wireless networks may occur in scenarios that cannot occur in wired ones. Two of the most common scenarios—commonly referred to as the *hidden* and *exposed terminal problems*, respectively—are depicted in Figure 1. Let us assume that the network contains five identical devices or nodes A, B, C, D and E, and that the distances between the pairs A and B, B and C, C and D, and D and E are equal and just a bit smaller than the transmission range of each node.

- The hidden terminal problem occurs when the nodes C and E simultaneously start transmission toward node D. They cannot hear each other's transmission (they are hidden from each other), but node D can hear both of them and it senses a collision on the medium.
- The exposed terminal problem is experienced by node B, which wants to transmit a packet to node A. However, on checking the medium, node B overhears the transmission from C, even though it is not the intended recipient, and must defer its transmission for a while. At best, it has to wait until C finishes its transmission. Time and bandwidth are thus wasted because there can be no collision at A, which cannot hear transmissions from C.

The hidden and exposed terminal problems are particularly harmful because some of the offenders may be entirely unaware of them; as a result, such conditions are difficult to control and prevent. Both problems become even more complex if different nodes transmit at different power levels and thus have different transmission ranges. It should come as no surprise that these problems are addressed–with a varying degree of success—by virtually all of the MAC protocols for wireless ad hoc networks.

Because collision detection is not available, MAC protocols for wireless networks must rely on collision avoidance techniques (including explicit scheduling, bandwidth reservation, and listening to the medium) *before* attempting to transmit a packet. This last procedure is commonly known as clear *channel assessment* (O'Hara and Petrick 1999; IEEE 2003a, 2003b), although other terms may be occasionally encountered as well.

Obviously, MAC protocols in wireless networks face both traditional challenges encountered in wired networks and new ones that stem from the use of the wireless communication medium. The most important features of MACs in ad hoc wireless networks can be summarized as follows (Ram Murthy and Manoj 2004):

- The operation of the protocol should be distributed, preferably without a dedicated central controller. If the use of such a controller cannot be avoided, then the role should be only temporary, and devices with appropriate capabilities must be allowed to undertake it for a certain period of time.
- The protocol should be scalable to large networks.
- The available bandwidth must be utilized efficiently, including the minimization of packet collisions and minimization of the overhead needed to monitor and control network operation. In particular, the protocol should minimize the effects of hidden and exposed node problems.
- The protocol should ensure fair bandwidth allocation to all the nodes. Preferably, the fairness mechanism should take into account the current level of congestion in the network.
- The MAC protocol should incorporate power-management policies to minimize the power consumption of the node and of the entire network.
- The protocol should provide *quality-of-service* (QoS) support for real-time traffic wherever possible. Real-time, in this context, implies data traffic with prescribed performance bounds; these may include throughput, delay, delay jitter, and other performance indicators.

Two additional issues deserve mention. The first issue is time synchronization among the nodes, which is required for the purpose of bandwidth reservation and allocation. Time synchronization is usually achieved by having one of the nodes periodically broadcast some sort of synchronization signal (the *beacon*), which is then used by other nodes. Although the use of periodic beacon transmissions facilitates the process of placing the reservation requests and subsequent broadcasting of reservation allocations, it requires that some node is capable of, and willing to, act as the central controller—which is somewhat contrary to the distributed, self-organizing character of an ad hoc network. In particular, additional provisions must be made to replace the controller node when it departs from

the network or experiences a failure; this is part of the self-healing property of ad hoc networks described above. Furthermore, the use of beacons consumes the bandwidth and affects the scalability of the MAC algorithm.

The second issue is related to the interference from neighboring nodes. As this interference is harmful, steps have to be taken to reduce it, most often through appropriate multiplexing techniques. According to Stallings (2002), multiplexing techniques are available in the following domains:

- in the frequency domain (i.e., through frequency division multiple access), wherein different frequency bands are allocated to different devices or subnetworks;
- in the code domain (i.e., through code division multiple access), wherein different devices use different code sequences;
- in the time domain (i.e., through time division multiple access), wherein different devices transmit at different time; or
- in the space domain, where the range and scope of transmissions is controlled through the use of transmitter power control and directional antennae, respectively.

Strictly speaking, all of these techniques belong to the PHY layer, and they are reviewed in greater detail in some of the earlier chapters. Although the MAC layer is completely oblivious to the first two techniques, it can utilize the latter two (multiplexing in time and space domain) or even integrate them to a certain extent. (For example, time multiplexing is a close relative of scheduling.) This cross-layer integration and optimization allows the MAC protocol to better address the requirements outlined above. We note that such integration is not too common in ad hoc networks, where the MAC layer is more likely to cooperate with the network (and possibly transport layers above it) than with the PHY layer below; still, MAC protocols exist that make use of it, as will be explained later.

CLASSIFICATION OF MAC PROTOCOLS FOR AD HOC NETWORKS

Before we present some of the important MAC protocols for wireless ad hoc networks, we will give a brief overview of possible criteria for classifying those protocols; the reader will thus be able to grasp main features of different MAC protocols and identify the important similarities as well as differences among them.

Mechanism for Accessing the Medium

Probably the most intuitive among the classification criteria is the manner of accessing the medium, which comes in the following three main flavors:

- *contention-based protocols* are those in which a potential sender node must compete with all others in order to gain access to the medium and transmit its data;
- *Bandwidth reservation-based protocols* in which provisions exist for requesting and obtaining bandwidth (or time) allocations by individual senders; and
- *scheduling-based protocols* in which the transmissions of individual senders are scheduled according to some predefined policy that aims to achieve one or more of the objectives outlined above, such as the maximization of throughput, fairness, flow priority, or QoS support.

Note that the third option requires the presence of an entity that is responsible for implementing the aforementioned policy. In most cases, this requirement translates into the requirement for a permanent or temporary central controller. Note also that the policy to be pursued should be adaptive, depending on the traffic or other conditions in the network. The presence of a central controller is also sometimes needed in protocols that use the second option.

Quite a few among the existing MAC protocols offer more than one of those mechanisms. This may be accomplished by slicing the available time into intervals of fixed or variable size, which are referred to as *cycles* or *superframes* (O'Hara and Petrick 1999; IEEE 2003a, 2003b), and assigning certain portions of those intervals to different categories of access from the list above. For example, the IEEE 802.11 *point coordinator function* (PCF) uses superframes in which the first part is reserved for (optional) contention-free access, while the second part is used for contention-based access (ANSI/IEEE 1999; O'Hara and Petrick 1999). A similar approach is adopted in the IEEE 802.15.4 protocol in its beacon-enabled, slotted *carrier sense multiple access with collision avoidance* (CSMA/CA) mode (IEEE 2003b), except that the contention access period precedes the contention-free period in the superframe.

On the other hand, some MAC protocols offer optional features that modify the manner in which the protocol operates and effectively introduce a different mechanism for medium access control. For example, the IEEE 802.11 *distributed coordinator function* (DCF) utilizes pure contention-based access in its default form but allows bandwidth reservation on a per-packet basis through the optional *request-to-send, clear-to-send* (RTS–CTS) handshake (ANSI/IEEE 1999).

Alternative Classifications on the Basis of Medium Access Mechanism

An alternative classification criterion could be devised by assuming that contention-based access will always be present and then using the presence or absence of the latter two access mechanisms as the basis for classification. This approach results in the common (and marginally more practical) classification into pure contention-based MACs, contention-based MACs with reservation mechanisms, and contention-based MACs with scheduling mechanisms (Ram Murthy and Manoj 2004). A variant of this approach distinguishes between contention-based or random access–based protocols, scheduling or partitioning ones, and polling-based ones. Yet even these classifications are neither unambiguous because the presence of optional features outlined above leads to the same protocol being attached to more than one category nor comprehensive because some of the existing protocols cannot be attached to any single category (Ram Murthy and Manoj 2004). Because of these shortcomings, it is listed as an alternative only.

Mechanism Used for Bandwidth Reservation and Its Scope

These two criteria apply only to MAC protocols that employ some form of bandwidth reservation and thus actually represent subclassifications within the previous one based on the mechanism used to access the medium. With respect to the mechanism used for bandwidth reservation, we can distinguish between the protocols that use some kind of handshake (e.g., RTS–CTS) and those that use out-of-band signaling, most notably the *busy tone approach*, which is an extension of the familiar concept from the traditional telephony systems.

With respect to the scope of bandwidth reservation, we can distinguish between the protocols that request bandwidth for a specified time (i.e., for a single packet or for a group of consecutive packets, commonly referred to as a *burst*) and those that request bandwidth allocation for an unspecified time. In both cases, time can be measured in absolute units or in data packets. In the former case, bandwidth allocation is valid for the transmission of a specified number of packets only; in the latter case, it has to be explicitly revoked by some central authority or perhaps waived by the requester itself.

Another scheme based on the concept related to bandwidth reservation is the family of the so-called multichannel MAC protocols. Most communication technologies use only one channel out of several available in the given frequency band. Multichannel MACs exploit this feature to employ channel hopping in order to improve bandwidth utilization or reduce congestion.

Presence and Scope of Synchronization

The presence or absence of time synchronization among the nodes in the network is another criterion that can be used to classify MAC protocols for wireless ad hoc networks. Synchronization, if present, may need to be extended to all of the nodes in the network (*global synchronization*); alternatively, it may apply to just a handful of nodes that are physically close to one another (*local synchronization*). In the former case, a central controller may be needed to initiate and broadcast the necessary synchronization information.

Synchronization is most often linked to scheduling and bandwidth reservation, because basic synchronization intervals are often used to apportion the available bandwidth to appropriate sender nodes. However, bandwidth reservation and allocation can be accomplished in an asynchronous manner, in particular when reservation is requested on a per-packet basis, whereas synchronous protocols can be used even with pure contention-based access. For example, the IEEE 802.15.4 protocol in its beacon-enabled, slotted CSMA/CA mode without guaranteed time slots uses pure contention-based access, yet all transmissions must be synchronized to the beacon frames periodically sent by the network coordinator (IEEE 2003b).

Synchronization is one of the most important factors that may affect scalability of the network. As the size of the network grows, synchronization becomes more difficult and more costly to establish and maintain. In particular, protocols that rely on global synchronization will suffer the most degradation; for example, it has been shown that the construction and maintenance of a globally optimal schedule in a multilevel Bluetooth network (a *scatternet*) is a nondeterministic polynomial time-complete problem (Johansson, Körner, and Tassiulas 2001).

Presence and Permanence of a Controller

Another possible classification criterion is the presence and permanence of a central network controller or coordinator. Whereas wireless ad hoc networks by default should be able to function without a permanent or dedicated central controller, quite a few protocols rely on certain monitoring and control functions that can only be provided by a local or global controller. This is the case with several of the MAC protocols that use bandwidth reservation, as well as with all of the MAC protocols that use scheduling. In fact, even some pure contention-based protocols rely on the presence of a controller for administrative tasks such as time synchronization and sometimes even node admission. Again, the presence of a controller affects the scalability of the network because the amount of work the controller has to do—most of which is administrative and control overhead—must grow with the number of nodes. Hierarchical decomposition or layering is often used to reduce this overhead, but it leads to additional problems in synchronization and delays.

Interference Reduction Mechanism

With respect to the interference reduction mechanisms mentioned above, a clear-cut classification may be hard to define. Multiplexing in the time domain (i.e., through *time-division multiple access*, or TDMA) effectively means that the MAC protocol utilizes some kind of scheduling or, at the very least, bandwidth reservation. As a result, any classification based on the use of TDMA techniques is effectively a replication of the first classification above—that is, the one based on the mechanism for medium access control.

However, the use (or absence) of techniques for space domain multiplexing can still be useful as a criterion for classifying the MAC protocols, in which case we can distinguish among:

- protocols that do not use any form of multiplexing in the space domain,
- protocols that use power control to limit the transmission range, and
- protocols that use directional antennae to limit the scope of their transmissions.

We note that the second and third categories are not mutually exclusive and MAC protocols exist that use one or the other multiplexing technique—or both at the same time.

Interdependence of Classification Criteria

As can be seen, not all of the classification criteria outlined above are entirely independent of each other; rather, they exhibit a certain overlap or redundancy. Still, they are

useful in the study of MAC protocols because they tend to highlight different aspects of their design and operation. In the discussions that follow, we will look at the MAC protocols in the following order: contention-based protocols; protocols that use bandwidth reservation; protocols that use multiple channels, out-of-band signaling, and directional antennae; and protocols that use polling.

CONTENTION-BASED MAC PROTOCOLS

Basic Carrier Sense Multiple Access

Most of the MAC protocols are derived from the basic *carrier sense multiple access* (CSMA) mechanism (Bertsekas and Gallager 1991). CSMA is a pure distributed protocol without centralized control, which operates as follows. The node that wants to transmit a packet first performs the clear channel assessment procedure—that is, it listens to the medium for a prescribed time. If the medium is found to be clear (or idle) during that time, then the node can transmit its packet. Otherwise—that is, if another transmission is in progress—the node backs off by waiting for a certain time before undertaking the same procedure again.

Different MAC algorithms use different ways to calculate the time they need to listen to the channel during the clear channel assessment procedure and to calculate the time to wait (i.e., the duration of the backoff period) before the next transmission attempt.

It is possible that the transmissions from two or more nodes overlap in time, which results in a collision and loss of all packets involved. If lossless communication is desired, collisions must be detected so that the lost packets can be retransmitted. Because a collision can be detected only at the receiver side, some form of acknowledgment from the receiver may be needed; some MAC protocols provide this facility, whereas others leave it to some of the upper layers (most likely, the transport layer). The former approach is more efficient in terms of reaction time, whereas the latter allows for much simpler implementation of the MAC protocol used.

In the basic CSMA protocol, carrier sensing is performed only at the sending node. Therefore, the hidden terminal problem is still present. Moreover, the exposed terminal problem leads to deferred transmissions and thus reduces bandwidth utilization.

We note that a separate chapter is devoted to the details of the CSMA protocol.

IEEE 802.11 MAC

Strictly speaking, the IEEE 802.11 protocol (O'Hara and Petrick 1999) is intended for *wireless local area networks* (WLANs) rather than wireless ad hoc networks. However, it is interesting to examine it in some detail, mainly on account of its ubiquity, and because it uses most of the main concepts that are reused in many MAC protocols for ad hoc networks. The protocol covers the functional areas of access control, reliable data delivery, and security; in the following we will focus on the first two areas, as the last one (security) is beyond the scope of this chapter.

Reliable transfer is achieved through the use of *acknowledgment* (ACK) packets or frames, sent by the destination node upon successful data-packet reception. Medium access is regulated in two ways, the first of which is a distributed contention-based mechanism known as the *distributed coordination function*, which does not require a centralized controller. The DCF, based on the CSMA protocol described above, operates as follows. The node that wants to transmit a packet first performs the clear channel assessment procedure—that is, it listens to the medium for a time equal to the *interframe space* (IFS). If the medium is found to be clear (or idle) during that time, then the node can transmit its packet immediately; otherwise—that is, if another transmission is in progress—the node waits for another IFS period. If the medium remains idle during that period, the node backs off for a random interval and again senses the medium. During that time (referred to as the *backoff window* or *contention window*), if the medium becomes busy, the backoff counter is halted; it resumes when the medium becomes idle again. When the backoff counter expires and the medium is found to be idle, the node can transmit the packet.

A possible scenario in which this procedure is applied is schematically depicted in Figure 2. There are several points worth mentioning. First, the backoff interval is chosen as a random number from a predefined range. After each collision, the range is doubled to reduce the likelihood of a repeated collision. After each successful transmission, the range is reset to its initial value, which is typically small. This approach is known as *binary exponential backoff* (BEB) (Stallings 2002) In this manner, the protocol ensures a certain level of load smoothing in case of frequent collisions caused by heavy traffic.

Second, to enhance reliability and avoid the hidden and exposed terminal problems to a certain extent, the RTS–CTS handshake—well known from wired communications—may optionally be used. In this case, the node that wants to send a data packet first sends a request-to-send packet to the designated receiver; if ready, it responds with a clear-to-send packet. Both RTS and CTS packets contain information about the duration of the forthcoming transmission, including the optional acknowledgment. Once the sender receives the CTS packet, it may begin actual data transmission, which may optionally be followed by an ACK packet. The RTS–CTS handshake constitutes a

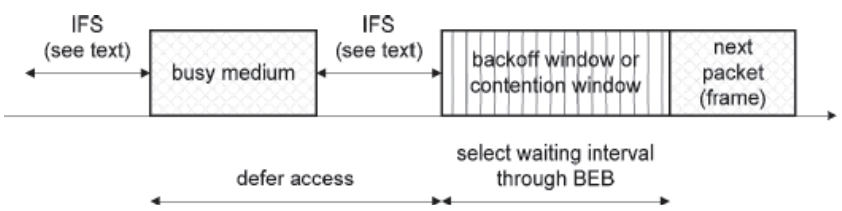

Figure 2: Basic access method in IEEE 802.11 DCF

simple form of bandwidth reservation on a per-packet or per-group basis, as will be explained below.

Reliability of transmission is enhanced because the RTS and CTS packets are generally much shorter than data packets; if they collide, the time waste is not high, but the risk that subsequent data packets will experience a collision is substantially reduced. The hidden terminal problem is avoided because other nodes within the transmission range of the receiver, on hearing the CTS packet, become aware of a forthcoming data transmission and defer their transmission for the time interval specified. On the other hand, a transmission from an exposed terminal may prevent the sender from initiating the RTS–CTS handshake. However, once the sender receives a proper CTS packet, it can assume that the receiver is not affected by the interfering transmission and can thus proceed with the data-packet transmission.

Third, to ensure the proper functioning of the protocol, three different IFS intervals are used: a *short IFS* (SIFS), a medium-duration *point coordination function IFS* (PIFS), and a long-duration *distributed coordination function IFS* (DIFS). The existence of several IFS intervals of different duration actually serves to implement different priority levels for different types of access. The DIFS interval is used for ordinary asynchronous traffic, whereas the SIFS interval, being the shortest, is used in the following cases:

- when the receiver sends an ACK packet on successful reception of a data packet (in this manner, ACK packets are safe from collisions because regular data packets wait longer);
- when the sender wants to send another data packet on receiving an ACK packet for a previous one (in this manner, a burst of packets commonly obtained by segmenting a longer packet from the upper layers can be delivered quickly and with little risk from collision, although such transmissions can result in unfairness because there is a limit on the duration of the burst that can be transmitted); and
- when the node sends a CTS packet on receiving a RTS packet from a prospective sender (again, the use of the SIFS interval minimizes the risk that the CTS packet will experience a collision).

The PIFS interval is used in an alternative access method known as the *point coordination function*, which is implemented on top of DCF. The PCF requires the presence of a central point coordinator, hence the name. The point coordinator defines an interval known as a *superframe*. In the first part of the superframe, the coordinator issues polls to all nodes configured for polling. The polls are sent using the regular CSMA algorithm outlined above. When a poll packet is sent, the polled node may respond using the SIFS interval. If the coordinator receives the response, then it issues another poll but using the PIFS interval. The polling continues in round-robin fashion (i.e., one node at a time) until all of the nodes are polled. Then the point coordinator remains idle until the end of the superframe, which allows for DCF-style contention-based access by all other nodes. The structure of the PCF superframe is schematically shown in Figure 3. The duration

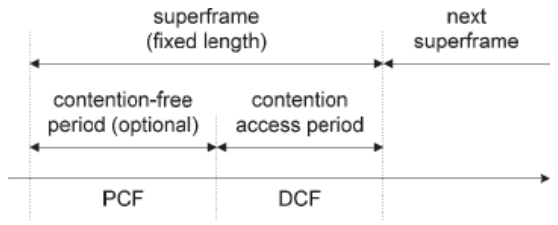

Figure 3: Superframe structure with IEEE 802.11 PCF

of the superframe is fixed, but an ongoing transmission may force the coordinator to defer the beginning of a polling cycle; in this case, the useful duration of the superframe will be reduced.

Although the IEEE 802.11 DCF is able to deal with asynchronous traffic, the presence of synchronous traffic with specified (and reasonably stable) throughput over prolonged periods of time is well served by its PCF counterpart. Still, the PCF functionality is designated as an optional facility in the 802.11 standard (ANSI/IEEE 1999), and it is rarely used in practice.

Multiple Access Collision Avoidance

One of the pioneering attempts to define a MAC protocol for ad hoc networks is the *multiple access collision avoidance* (MACA) protocol described by Karn (1990). Similar to the 802.11 DCF, the MACA protocol uses the three-way handshake with RTS and CTS packets preceding the data-packet transmission; but unlike the 802.11 protocol, MACA does not use carrier sensing. The sender initiates the handshake with a RTS packet, to which the receiver should respond with a CTS packet. Once the sender receives the CTS packet without errors, it commences the transmission of the data packet. If a packet is destroyed through a collision, the sender undertakes a backoff procedure using the BEB algorithm. Figure 4 depicts the operation of the MACA protocol. Note that explicit acknowledgments of successful data-packet transmissions are not used in this protocol.

Both RTS and CTS packets contain information about the duration of the data transmission. Nodes in the vicinity of the sender, such as node A in Figure 4, hear the RTS packet and defer the transmission of their packets so that the sender can receive the CTS packet. Nodes in the vicinity of the receiver (node B in Figure 4) hear the CTS packet and defer the transmission of their packets so that the receiver can receive the data packet. In this manner, both the hidden and exposed terminal problems are avoided—to a certain extent, because the risk of collisions is reduced but not entirely eliminated—and RTS and CTS packets can still collide. However, the exponential backoff used in the MACA protocol creates the risk of a different yet equally severe problem—namely, that of unfairness, which can quickly lead to starvation under heavy loads.

As an example, consider the network that consists of two sender and two receiver nodes, positioned in the manner schematically depicted in Figure 5. Let both senders 1 and 2 generate a high volume of traffic, and let sender 1 capture the medium first and start transmitting packets. Because of random timing of packet transmissions, a number of

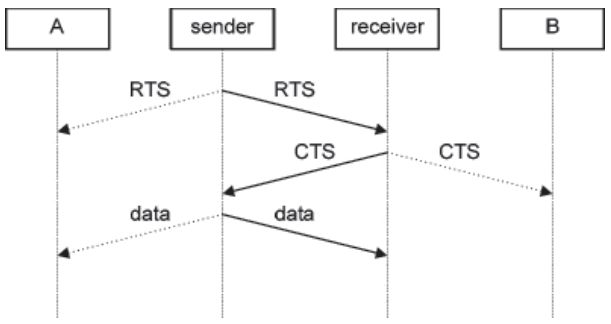

Figure 4: Timing diagram of the control handshake in the MACA protocol

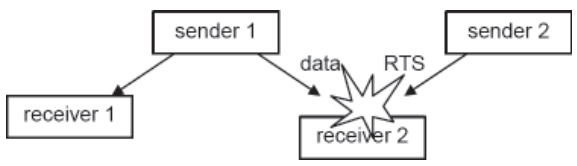

Figure 5: Collisions of data and RTS packets can cause starvation in the MACA protocol

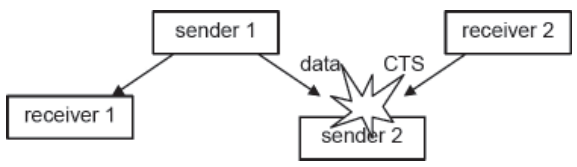

Figure 6: Collisions of data and CTS packets can cause starvation in the MACA protocol

packets from sender 2 may experience collisions. Because receiver 2 does not send proper CTS packets, sender 2 backs off repeatedly—but the backoff window keeps getting longer and longer because of the BEB algorithm. As a result, sender 2 is starved—that is, it is effectively blocked from accessing the medium.

Several variations of this problem occur in different scenarios. Consider, for example, the network with two senders and two receivers, shown in Figure 6, in which an exposed node (sender 2) is able to hear the transmissions from sender 1 but not those from receiver 1. Thus, it is free to commence its handshake while sender 1 transmits a data packet to the receiver 1. Sender 2 sends a RTS packet to receiver 2 which responds with a CTS packet, which in turn collides with the data packet from sender 1. Because the sender 2 does not receive a CTS packet, it will keep increasing its backoff window unnecessarily and effectively starve in the process.

A similar scenario is shown in Figure 7, where sender 2 is within the transmission range of receiver 2 but beyond the transmission range of sender 1. If sender 2 transmits data to receiver 2, then receiver 1 is able to hear those transmissions and unable to respond to the RTS packet sent by sender 1. Because sender 1 does not get any response (i.e., CTS packets) to its RTS packets, it concludes that there are collisions and effectively starves by continually increasing its backoff window.

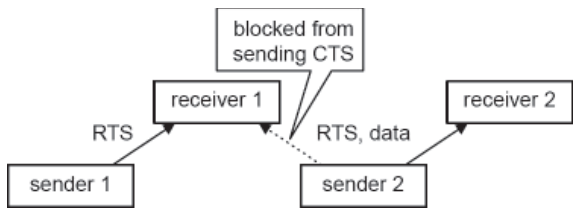

Figure 7: An exposed receiver can lead to starvation of a remote sender in the MACA protocol

Multiple Access Collision Avoidance Protocol for WLANs

To alleviate the problems of the MACA protocol, Bharghavan et al. (1994) have proposed several modifications to it; the improved protocol is known as *multiple access collision avoidance protocol for WLANs* (MACAW). The bulk of those modifications attempt to solve the starvation problems, most notably those caused by unfairness. An obvious solution is to somehow balance the backoff windows of neighboring nodes, which should result in balancing their respective probabilities to access the medium. First, modification consists of augmenting the packet header with an additional field that contains the current value of the backoff counter of the transmitting node. A node that receives the packet reads and copies it into its own backoff counter. In this manner, the backoff windows of the nodes will tend to have similar values.

Second, modification aims to avoid the rapid increase of the duration of the backoff window. The reader will recall that the original BEB algorithm prescribes that the range within which a random value is chosen for the backoff window is to be doubled after each collision and reset to the initial value after a successful transmission. As a result, the duration of the backoff window can exhibit rather large and abrupt variations. Instead, the MACAW algorithm uses the so-called multiplicative increase, linear decrease backoff mechanism. In this approach, a collision causes the backoff counter to be increased by a constant factor—for example, the value of 1.5 is typically used—whereas a successful transmission simply decrements the backoff counter by one. In this manner, the backoff counter changes much less abruptly, and long contention windows after unsuccessful transmissions are avoided.

Starvation of an exposed node is addressed through the introduction of a small control packet designated as a *data-sending* (DS) packet; this packet is sent immediately before the actual data packet. The corresponding timing diagram is shown in Figure 8, where nodes A and B defer their transmissions because of a pending data packet from sender to the receiver—although B learned about it from the CTS packet and A learned about it through the DS packet.

Another change introduces per-flow fairness instead of per node, as is the case in MACA. When several data flows originate at a single node, separate queues are maintained for each flow and the backoff procedure is performed independently for each queue. When a node has one or more packets ready for transmission, then it chooses the queue for which the backoff window is the shortest.

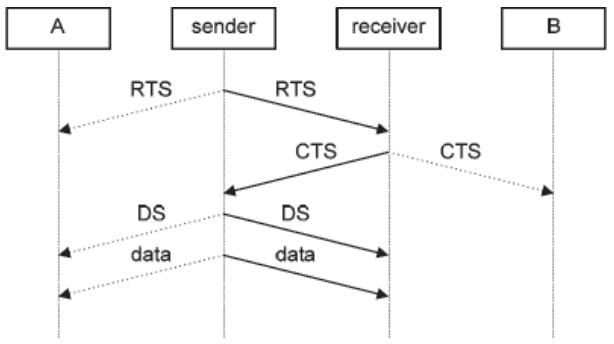

Figure 8: The role of the DS packet in the MACAW protocol

As mentioned above, the MACA protocol does not use acknowledgments, and reliable transport is the responsibility of the transport layer. However, most implementations of the TCP protocol use long time-out delays—on the order of hundreds of milliseconds—to be able to accommodate a wide range of transmission delays. Correction of erroneous or lost packets thus incurs substantial delays. To reduce those delays, the MACAW protocol introduces explicit ACK packets that are sent by the receiving node on successful reception of a data packet. If the ACK packet is missing, then the sender will attempt to retransmit the same packet, only this time with an increased value of the backoff counter. It retransmits the RTS packet for the same data packet. Now if the data packet was not correctly received the first time, the entire transmission cycle is repeated. However, if the data packet was correctly received (which means that the ACK packet was lost, most likely because of a collision), the receiver responds with the appropriate ACK packet instead. This informs the sender that the original data transmission was successful, and the sender may move on to the next data packet.

Another modification addresses the scenario depicted in Figure 7. This type of starvation is alleviated through the use of another control packet known as *request to request to send* (RRTS). In other words, when receiver 1 receives a RTS packet to which it cannot respond, it sends a RRTS packet to sender 1 in the next contention period. Neighboring nodes that hear this packet—including sender 2—are obliged to remain silent for two successive RTS–CTS cycles, which gives sender 1 sufficient time to retransmit its RTS packet and receive the corresponding CTS packet. In this manner, the remote sender is able to conclude the data-packet transmission without starvation. Figure 9 shows the corresponding timing diagram.

Floor Acquisition Multiple Access

Fullmer and Garcia-Luna-Aceves (1995) have proposed a family of protocols known as *floor acquisition multiple access* (FAMA) that generalize the approach based on CSMA access with the control packet handshake. In this sense, both the MACA and MACAW protocols belong to the FAMA family.

The basic concept behind the FAMA protocols is that the sender node has to acquire the floor before attempting to transmit its data. This is accomplished through the control packet handshake. Although control packets themselves

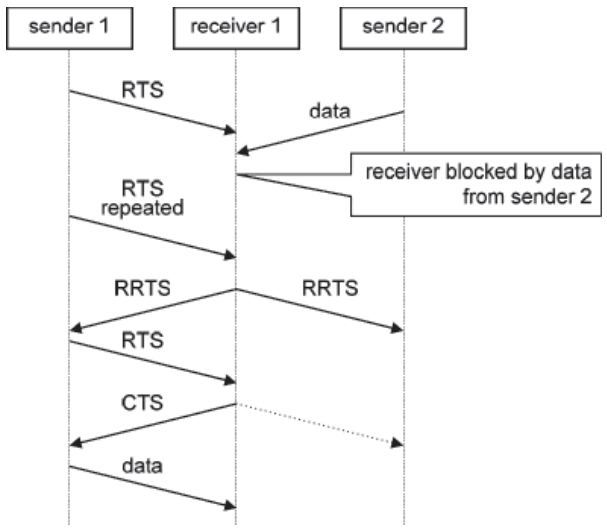

Figure 9: The role of the RRTS packet in the MACAW protocol

may suffer collisions, the protocols ensure that the data-packet transmission is collision-free. The original proposal (Fullmer and Garcia-Luna- Aceves 1995) discusses several variants of the FAMA protocol and derives the timing relationships and constraints that allow it to achieve collision-free data transmission. In this process, the round-trip propagation time of the channel is used to derive the waiting times at particular steps in the protocol.

Another interesting feature of the FAMA protocols is that some among them allow the sender to use a single control RTS–CTS handshake before sending a packet burst, similar to the 802.11 DCF protocol. In this case, the bandwidth utilization is improved because the overhead incurred by the control handshake is shared by all of the packets in the burst.

MACA by Invitation

Talluci, Gerla, and Fratta (1997) have adopted a different approach that aims to reduce the number of control packets (and, by extension, the overhead thus incurred); this protocol is known as *MACA by invitation* (MACA-BI). The main change from the original MACA protocol is that data transmission can be initiated by the receiver, rather than the sender. The receiver sends the *ready to receive* (RTR) packet to the sender, which, if ready, simply responds with the data packet. The corresponding timing diagram is shown in Figure 10, where the receiver's hidden neighbor A hears the RTR packet and defers its transmission until the data packet is received. However, collisions are not eliminated altogether through this technique. In particular, RTR packets may collide with each other (for example, when two potential receivers request data from the same sender) or with data packets sent from a hidden terminal in the vicinity.

The main problem of the MACA-BI protocol stems from the fact that the receiver does not know in advance the moments of packet arrivals at the sender or the length of those packets. Instead, the receiver must estimate those values and embed the estimates in the RTR packet. To improve the accuracy of those estimates and thus increase

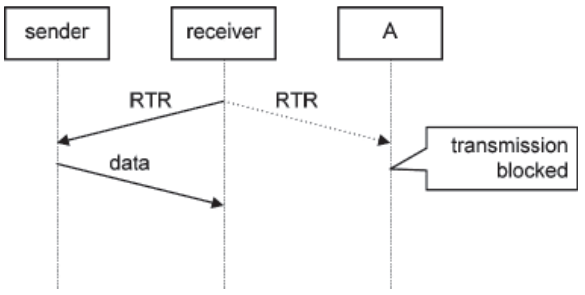

Figure 10: Receiver-initiated data-packet transmission in the MACA-BI protocol

the overall efficiency of the protocol, the format of the data packets may be modified to include control information about the data flows at the sender—for example, the number of backlogged packets queued for transmission and their individual lengths. Furthermore, the sender is allowed to initiate transmission in the usual manner—that is, by sending an RTS packet with appropriate control information.

MACA with Reduced Handshake

In many cases, the sender usually sends a burst of packets to the receiver, rather than a single packet: for example, when TCP connections are established in a wireless ad hoc network. This observation has inspired Toh et al. (2000) to devise a protocol known as *MACA with reduced handshake* (MARCH) that aims to reduce the overhead incurred by the control packet handshake. In this protocol, data-packet transmission is initiated by the sender, which sends a regular RTS packet to the receiver; the receiver responds with a CTS packet. Subsequent transmissions use only RTS packets preceding the data packets. Obviously, the more packets are transmitted in a burst, the more efficient transmission will be because the control packet overhead is shared by all of the packets in the burst.

By itself, this feature would not be worth mentioning because other protocols described above contain similar provisions. But the importance (and the main contribution) of the MARCH protocol is in its use of a reduced handshake for multihop transmissions through one or more intermediate nodes, a common scenario in wireless ad hoc networks (Toh 2002) that neither the original MACA nor its numerous variants specifically address. Consider the network with four nodes A, B, C, and D, in which data packets are sent in bursts from A to D through intermediate nodes B and C. In MACA, this transmission has to be implemented as a series of single-hop (i.e., node-to-node) transfers with the full control handshake for each hop, as shown in Figure 11. The MARCH protocol makes use of the fact that the RTS–CTS exchange between the sender and the receiver can often be heard by the next hop receiver; in our example, the control handshake between nodes A and B actually alerts the next hop receiver C to the forthcoming data transmission. Consequently, the control handshake between the nodes B and C can be simplified by omitting the RTS packets. Instead, node C can simply send the CTS packet to node B as soon as the data transmission from node A is finished. In this

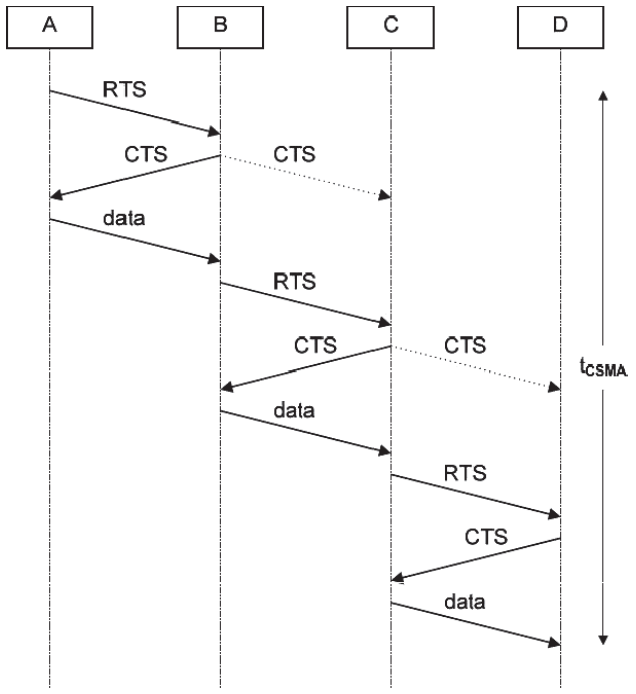

Figure 11: Multihop tramsmission of a packet burst in the original MACA protocol

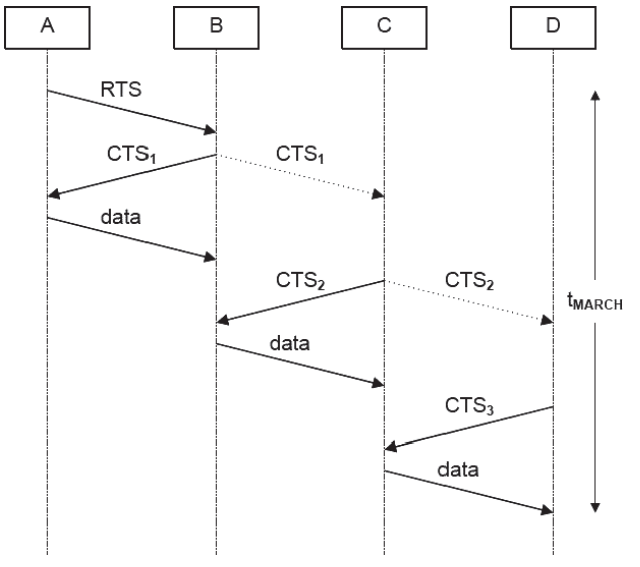

Figure 12: Multihop transmission of a packet burst in the MARCH protocl

manner, the control handshake is simplified and the associated overhead is reduced, which leads to substantial improvements in efficiency (i.e., increased throughput and reduced end-to-end delay) when compared to the original MACA or its variants described earlier. The control handshake in the MARCH protocol is shown in Figure 12.

We note that the MARCH protocol does require assistance from the routing algorithm because the CTS packets have to carry proper route identification so that the intended receiver (and only it) can issue the required CTS.

MAC PROTOCOLS THAT USE BANDWIDTH RESERVATION

Bandwidth reservation can be implemented only if some form of multiplexing in the time domain—that is, TDMA—is used. In the TDMA approach, available time is partitioned into time frames or slots that can be requested by and subsequently allocated to the nodes that have data to send. As noted above, this approach necessitates the presence of a centralized control in the form of a dedicated controller node. Time partitioning is typically done in a hierarchical manner, with smaller intervals (slots or frames) grouped into larger ones (cycles or superframes). Node transmissions are then aligned to the slot boundaries. In this manner, the controller node is able to monitor the utilization of the channel and to allocate bandwidth while taking into account QoS requirements and fairness among the nodes; bandwidth allocation also can be performed on a permanent or one-off basis.

In the area of wireless networks, the concept of bandwidth reservation was historically first exploited with satellite networks (Bertsekas and Gallager 1991) and later in wireless ATM starting with the *packet reservation multiple access* (PRMA) protocol described by Goodman et al. (1989). A nice overview of reservation techniques for wireless ATM networks can be found in the work by Sanchez, Martinez, and Marcellin (1997). We will now describe a few representative protocols that use the reservation approach in wireless ad hoc networks.

Distributed Packet Reservation Multiple Access

In the *distributed PRMA* (D-PRMA) protocol proposed by Jian et al. (2002), each frame is divided in slots, and each slot is divided into minislots, as shown in Figure 13. Each minislot can be used for data transmission or for control handshake. In the latter case, the first part of the minislot is used for the RTS packet or *busy indication* (BI) signal, while the second part is used for the CTS or BI. All nodes that have a packet for transmission must listen to the medium at the beginning of each slot. If the medium is free, they compete for access by sending the RTS packet in the RTS–BI part of the first minislot of each slot. If the node receives the CTS response in the same minislot, it can use the remaining portion of the slot (minislots 2 to m) for its transmission. If a collision occurs in the RTS–BI field, the contention process continues in subsequent minislots of the same slot until one node wins the slot.

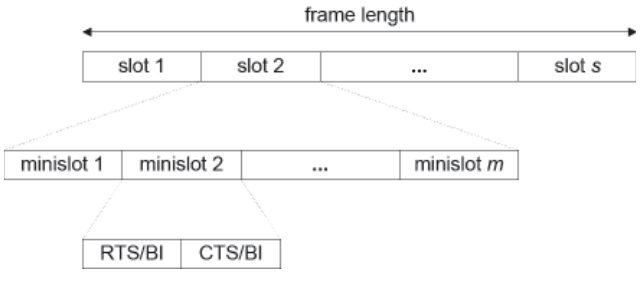

Figure 13: Frame structure in D-PRMA protocol

Once a node wins the slot, it transmits its data in the next minislot. Transmission lasts until the end of the slot or until there is no more data, whichever is shorter. Other nodes that want to transmit will find the channel busy and have to wait. Provisions are made to eliminate the risk of collision for CTS packets (the designated receiver sends a CTS packet only if it receives a RTS packet correctly—that is, without collision), the hidden terminal problem (nodes that hear the CTS packet defer their transmission until the end of the current slot) as well as the exposed terminal problem (nodes that hear the RTS but not the CTS packet are still allowed to transmit).

The D-PRMA protocol tries to cater to the fact that nodes in an ad hoc network can carry both synchronous (voice or multimedia) and asynchronous data traffic; the former is generally delay-sensitive but much less loss-sensitive than the latter. To give priority to synchronous traffic as soon as it is generated by the application, the D-PRMA protocol introduces priorities to each node. Priority is a parameter p with a value between 0 and 1; nodes will start contending for access in a free minislot with probability p or skip it and wait for the next opportunity with probability $1 - p$. Nodes with synchronous traffic are allowed two exceptions: (1) Any such node can start the contention process in minislot 1 with the probability of 1 and (2) any such node that manages to win a slot is allowed to reserve that same slot in every frame until the end of the session. (Nodes with asynchronous traffic can win a single slot only and have to contend again for each subsequent slot.)

Reserved slots are identified through a BI signal sent by the sender (the node with synchronous traffic that has won the slot in the previous frame) in the RTS–BI portion of minislot 1, and by the receiver in the CTS–BI part of that same minislot. In this manner, other nodes learn about the reservation and do not contend for that slot in the current frame. End of the data stream and the corresponding reservation is announced simply by the absence of the BI signal.

We note that the RTS packet carries the address of its one-hop destination, and only the destination node is allowed to reply with a CTS packet. Nodes that hear the CTS packet are not allowed to transmit within the same slot. Similar to the MACA protocol, any node that hears the RTS packet but not the CTS one is allowed to transmit.

Hop Reservation Multiple Access Protocol

The *hop reservation multiple access* (HRMA) protocol utilizes multiplexing in both time and frequency domain (Tang and Garcia-Luna-Aceves 1999a). In the time domain, available time is partitioned into frames, which are in turn divided in slots. In the frequency domain, different frequencies are used for different slots; in this manner, the HRMA is effectively a variant of the well know *frequency hopping spread spectrum* (FHSS) approach (Stallings 2002). The first slot in the frame is selected as the synchronization slot, which is why it always uses the same frequency. All other slots consist of two subslots that use different frequencies from the hopping sequence. The first subslot is used for transmitting hop-reservation packets, RTS packets, CTS packets, and data packets. The second subslot is used for receiving acknowledgement packets for

the data transmission received during the previous subslot. The frame structure of the HRMA protocol is shown in Figure 14.

Collision Avoidance Time Allocation Protocol

The *collision avoidance time allocation* (CATA) protocol divides the frame into a number of equally sized slots (Tang and Garcia-Luna-Aceves 1999b). Each slot is divided into five minislots: The first four are control minislots, and the fifth one is the data minislot. A node that wishes to transmit data reserves the data minislot by transmitting a RTS packet in the second control minislot. The intended receiver responds with a CTS packet in the third control minislot. On receiving the CTS packet, the sending node transmits a *not-to-send* (NTS) packet in the fourth minislot; the NTS packet serves to regulate the multicast or broadcast transmission. The frame structure of the CATA protocol is shown in Figure 15.

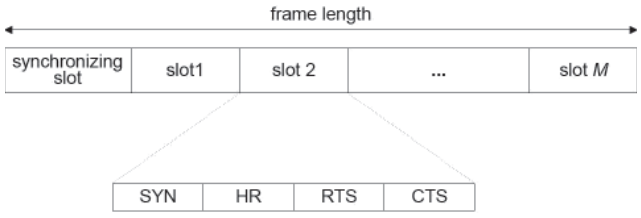

Figure 14: Frame structure in HRMA protocl

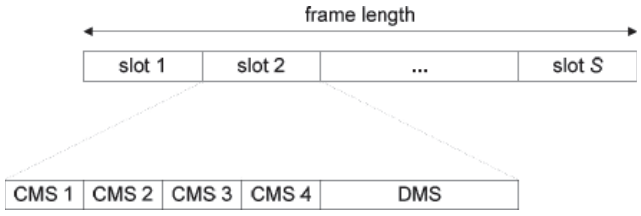

Figure 15: Frame structure in the CATA protocol

MACA with Piggybacked Reservation

The *MACA with piggybacked reservation* (MACA-PR), originally proposed by Lin and Gerla (1999), is an example of the cross layer interaction of MAC and routing layers with two priority levels: for real-time and data packets, respectively. Communications in the MACA-PR protocol are schematically shown in Figure 16. As in many other protocols, time is divided in slots. Each node maintains the *reservation table* (RT) with all reserved slots of the nodes in its transmission range. A node that has a non–real-time packet ready for transmission will first perform carrier sensing in the slot that is labeled as free in its RT. If the channel is found to be free in that slot, then the node transmits the RTS packet and receives CTS packet. Both RTS and CTS packets contain the information about the time period needed for data-packet transmission (labeled *network allocation vector*, or NAV). The CTS packet is followed by the data packet, which is in turn followed by the acknowledgement.

For real-time traffic in which data packets are sent with known periodicity, the first packet from the stream is sent as a regular non–real-time packet. However, the reservation for the next packet is piggybacked on the current data packet. This reservation contains the time period in which the next data packet is to be transmitted. All neighbors that hear the reservation will update their reservation tables accordingly. In case of transmission failure, real-time packets are not retransmitted.

The MACA-PR protocol avoids the hidden terminal problem by periodic exchange of reservation tables. Furthermore, reservation information times out if it is not refreshed during a certain time period.

Distributed Priority Scheduling

The *distributed priority scheduling* (DPS) protocol was initially proposed by Kanodia et al. (2004). The DPS protocol assumes that nodes use the 802.11 distributed

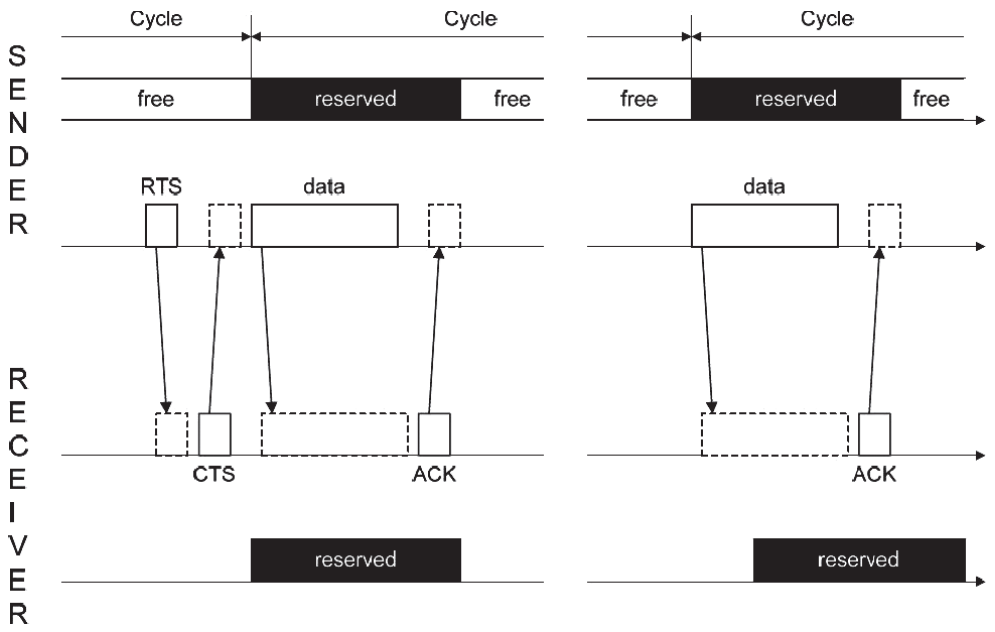

Figure 16: Communication in MACA-PR protocol

coordination function with the RTS–CTS–data–ACK handshake. Nodes send packet priority information piggybacked at the end of the RTS packet. Each node stores the information about the priority of overheard packets, and calculates its own priority relative to the nodes in the neighborhood. The receiving node will append the priority information at the CTS packet together with the sender's ID. Neighboring nodes then extract priority and node ID information from the RTS and CTS packets they have overheard and calculate their own rank relative to the neighbors. The rank is used to calculate the node backoff window according to the 802.11 DCF. For multihop communications, where end-to-end delay bounds for the packet are specified, the packet priority may change (increase) in the downstream nodes in order to meet the bound. Communication of table updates in the DPS protocol is shown in Figure 17.

MULTICHANNEL MAC PROTOCOLS

The main distinguishing feature of the protocols from the multichannel MAC protocols group is the utilization of several available channels—effectively, a form of frequency and channel multiplexing—to reduce congestion. Conceptually, this idea is not new: Many protocols apply channel partitioning—for example, 802.11 (ANSI/IEEE 1999; O'Hara and Petrick 1999) or HRMA (Tang and Garcia-Luna-Aceves 1999a)—and the FHSS variant of the CDMA approach is widely used at the physical level (Stallings 2002). But all of these protocols allocate individual channels according to a predetermined hopping sequence or a pseudo-random sequence in case of FHSS. On the contrary, multichannel MAC protocols select channels in a dynamic fashion that takes into account the current state of congestion of all of the available channels. Two main approaches are possible: (1) a dedicated channel is used for coordination and control and (2) the entire transmission (i.e., both control handshake and data) of each flow uses a different channel (Shi, Salonidis, and Knightly 2006). Although both approaches are capable of improving throughput, they suffer from different problems. In the first, some node pairs may be unable to communicate for prolonged periods; in the second, the dedicated control channel incurs some overhead, and its capacity should be carefully determined to avoid congestion in that channel. We will now present a few of the most important multichannel MAC protocols.

Multichannel CSMA with Soft Channel Reservation

This multichannel CSMA protocol was originally proposed by Nasipuri, Zhuang, and Das (1999). It assumes that the available frequency band is partitioned into N channels, and each node monitors all N channels whenever it is not transmitting. Channels in which the total received signal strength is below a predefined threshold are marked as idle, and the time instants at which this parameters drops below the threshold are marked; remaining channels are busy. When a packet arrives, the node checks its list of idle channels to see if the last channel it used is free. If so, this channel is used to transmit the packet; if not, a channel is randomly selected from the list. If there are no free channels, the node waits for the first channel to become idle. If the selected channel has remained idle during a period of long interframe space, the node initiates the transmission; otherwise, a random backoff is undertaken. An ongoing backoff is immediately cancelled if the channel becomes busy, and a new backoff is scheduled when the channel becomes idle again. *Soft reservation* refers to the fact that the scheme obviously gives preference to the last channel that was successfully used and thus tends to reserve a channel for each node (Nasipuri, Zhuang, and Das 1999). Distributed, dynamic channel allocation allows this scheme to significantly reduce the chances of collision, even in heavy traffic conditions, but it still suffers from two major problems. At the implementation level, the radio subsystem must be sophisticated enough to monitor the received signal strength on a per-channel basis for all channels; such

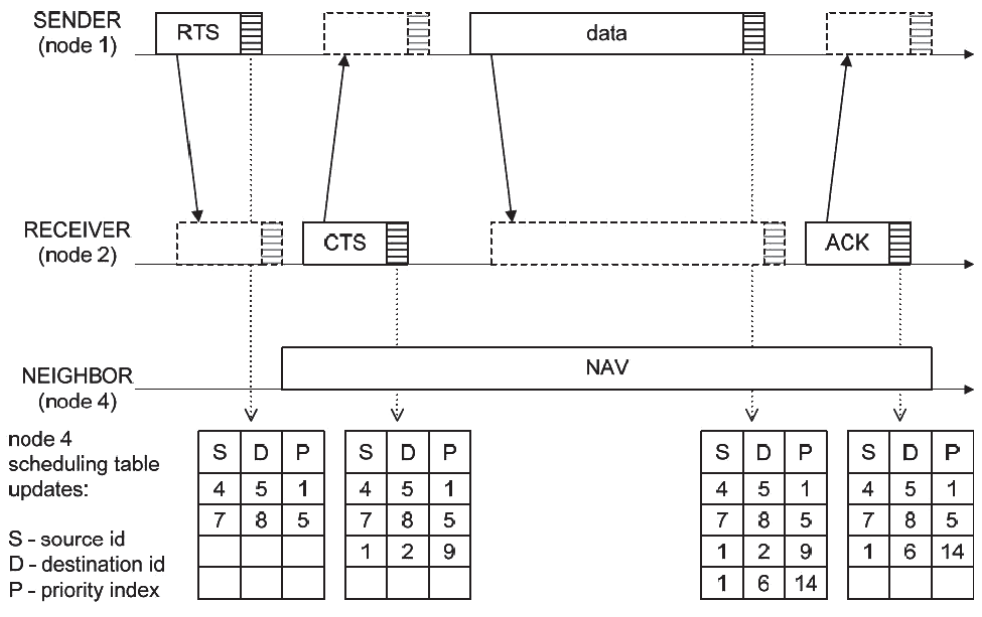

Figure 17: Packet transmission in the DPS protocol

radios are more complex and more costly. At the conceptual level, the fact that channels are selected by the transmitting node means that the scheme still suffers from the hidden terminal problem at the receiver node.

This last observation was the foundation for the improved scheme in which channel selection is performed by the receiver (Jain, Das, and Nasipuri 2001). In the improved scheme, a dedicated, shared channel is used to transmit RTS and CTS packets, which is effectively a form of out-of-band signaling (to be discussed below). A node that has a packet to send begins by sensing the state of all data channels and builds a list of free channels, much as in the original proposal. If no free channel is found, ten a backoff procedure is undertaken and the sensing is repeated. Once a free channel is found, the node then sends a RTS packet via the control channel; this packet contains the transmitter's list of free channels. The receiver then forms its own list of free channels, compares it to the one supplied by the transmitted, and selects the channel to be used for actual data transmission in the following manner. If the two lists overlap, then the receiver selects the best common channel (i.e., the one with the least received signal strength and consequently the least interference); this information is sent back to the transmitter within an RTS packet. If the two lists do not overlap, then the receiver does nothing; the transmitter eventually times out and repeats the procedure after a backoff. On receiving a valid RTS packet, the data packet is transmitted on the designated data channel. Successful transmission is acknowledged on the same data channel; failure to receive proper acknowledgment triggers another backoff and repetition of the entire procedure.

It is worth noting that nodes that overhear the RTS packet do not attempt to use the designated data channel during the entire duration of data transmission. Nodes in the vicinity of the transmitter are able to hear the RTS packet but not the CTS packet; these are required to refrain from transmitting only for the duration of the CTS transmission (in fact, until the transmitter's time out) but do not have to wait until the final ACK packet. In this manner, collisions between control handshake and data packets are avoided, the hidden terminal problem is alleviated, and throughput is improved. Again, the main problem is related to the implementation of the radio subsystem capable of monitoring the state of all channels in a given frequency band.

Slotted Seeded Channel Hopping

The *slotted seeded channel hopping* (SSCH) approach was initially proposed by Bahl, Chandra, and Dunagan (2004). SSCH is a distributed protocol for making and coordinating channel switching decisions, with the main objective of improving the throughput. Although the original paper describes the implementation in the IEEE 802.11 environment, the SSCH protocol is portable to other environments as well because it can be implemented in software without any modification of the radio subsystem. In SSCH, time is partitioned into slots of 10 ms, which suffices for approximately thirty-five transmissions of packets of maximum allowed length; this value was chosen to minimize the overhead of channel switching. Each node maintains a list of channels that will be used, as well as the times when transmissions will switch to the corresponding channels; this is referred to as the *channel schedule*. Each node also maintains, or strives to maintain, the channel schedules for all of the other nodes it is aware of. In this manner, the protocol ensures that any given pair of nodes will ultimately be able to communicate, despite the possibility of unexpected schedule changes. Channel schedules are kept in a compact form as a current channel and a rule used for updating the channel to avoid the excessive overhead needed for synchronization. The rule is actually a seed used to update the channel table, hence the name of the protocol. Data packets are kept in separate per-neighbor queues that are themselves ordered by perceived neighbor reachability. At the beginning of a slot, packet transmissions are attempted, preceded by the IEEE 802.11 RTS–CTS control handshake; the protocol visits all of the queues in a round-robin fashion. Unsuccessful transmissions cause the corresponding flow to be assigned lower priority for one-half of a slot duration, which limits the bandwidth waste incurred when transmitting on a wrong channel, or to an unreachable node. At the same time, packets are not dropped from the queue before the destination node is found unreachable during a full cycle of all channels. In this manner, the SSCH protocol is able to improve bandwidth utilization compared to other approaches.

Asynchronous Multichannel Coordination Protocol

The use of multiple channels in the *asynchronous multichannel coordination protocol* (AMCP) by Shi, Salonidis, and Knightly (2006) is guided by the goal of reducing the likelihood of starvation in CSMA-based multihop networks. In this case, the main cause of starvation is the lack of coordination between transmitters that cannot hear each other but prevent others from transmitting (even though this would not cause collisions). The AMCP protocol addresses this problem through the use of the so-called channel tables kept by each node that contain information about the scheduled availability of the data channels. In addition, each node may choose a preferred data channel for its own transmissions. When a data packet is to be transmitted, the node will check if its preferred channel is available; if not, an available channel is selected at random. The node then attempts the RTS–CTS handshake on the control channel; the RTS packet includes the information about the selected data channel. If the receiver finds it available, it responds with a so-called confirming CTS packet and immediately switches to the designated channel for data transmission. The transmitter switches to the designated channel and transmits the data packet. On successful reception, both nodes label the channel as *preferred* and switch back to the control channel. If the receiver finds that the designated channel is not available, it responds with a rejecting CTS packet that contains a list of available channels and remains on the control channel. The transmitter randomly selects another channel from those available to both transmitter and receiver and undertakes another round of control handshake.

In the AMCP protocol, coordination is accomplished as follows. A neighboring node that overhears an RTS or a

confirming CTS packet updates its channel table by labeling the designated channel unavailable for the announced duration of data transmission (including the CTS and ACK packets). No action is taken when a rejecting CTS is overheard. Starvation is avoided because this node can still use other available channels for its data transmissions. Furthermore, if the neighboring node wants to transmit data to one of the nodes that have just started their control handshake, it will defer its transmission for the entire duration of the ongoing transmission but set its contention window size to the minimum value; in this manner, it can undertake the deferred transmission as soon as possible, with a high probability of success, and thus minimize the likelihood of starvation.

MAC PROTOCOLS WITH POWER MANAGEMENT

Protocols from the power-management group try to reduce contention by controlling the transmission power on a per-packet basis. The obvious but naïve solution is to transmit RTS and CTS packets at the maximum available power so that they reach the widest possible audience. On the contrary, data and ACK packets should be transmitted at the lowest power level needed to reach their target to minimize the risk of contention. Although the risk of contention is indeed reduced but not fully eliminated, the susceptibility of data and ACK packets to noise and interference is increased. In extreme cases, this approach may result in unidirectional links—that is, one node can hear the other but the opposite is not true. These cases are much more difficult to handle.

Power-Aware Multiaccess Protocol with Signaling

Singh and Raghavendra (1998) were among the first to try to combine the MAC protocol with power-conserving features. The communication protocol itself is based on the MACA protocol (Karn 1990), augmented with a separate control and signaling channel similar in concept to the *busy tone multiple access* (BTMA) approach of Tobagi and Kleinrock (1975). In *power-aware multiaccess protocol with signaling* (PAMAS), a node that wants to transmit a packet sends the RTS packet to the designated receiver. The receiver that hears the RTS packet checks the control channel first; if no transmission is heard for a specified time t (equal to the round-trip time plus the duration of the RTS–CTS packet), it responds with the CTS packet and transmits a busy tone on the control channel. Unlike some of the algorithms described earlier, the busy tone in PAMAS is shorter: It lasts twice as long as the RTS–CTS packet. The busy tone is also sent when the node receiving a packet hears a RTS packet or detects some activity on the control channel; this prevents other potential receivers from sending their own CTS and thus prevents a potentially interfering transmission (the hidden terminal problem). On receiving the CTS packet, the sender begins the data-packet transmission. If the CTS packet is not received within the specified time-out, the sender performs a binary exponential backoff and repeats the attempt. The backoff countdown may be interrupted if an RTS request is received from another node, in which case the sender switches to reception mode according to the rules outlined above.

To conserve power, any node without packets to send goes to sleep. Moreover, if an ongoing transmission is detected in the neighborhood (through the presence of a busy tone on the control channel), all unaffected nodes should go to sleep. In the latter case, the sleep should last as long as the ongoing transmission. If the node wakes up without any packets to transmit, it goes to sleep again, but if it wakes up with a packet ready to be sent, it simply sends an RTS packet. If another transmission is in progress, the node conducts a probe over the control channel to find out how long will this transmission last. The probe protocol can be simplified considerably if the control channel is always active (Singh and Raghavendra 1998); however, this defeats the purpose for which the power control mechanism was introduced in the first place.

The PAMAS protocol is important because it shows that the medium access control mechanism can be linked with the power-conservation mechanism without affecting the end-to-end packet delay (Singh and Raghavendra 1998). Its main problem stems from the separation of data and control channels because such radios are infeasible to implement in resource-constrained sensor nodes. A further problem is the implicit assumption that switching in and out of the active state is much shorter than the average packet transmission time, which simply does not hold in most real systems.

Dynamic Power-Saving Mechanism

Jung and Vaidya (2002) have described a *dynamic power-saving mechanism* (DPSM) that optimizes the power saving mechanisms available in the IEEE 802.11 DCF (ANSI/IEEE 1999). In the DPSM scheme, time is divided into beacon intervals to which all nodes should synchronize. At the beginning of each beacon interval, all nodes must be awake for the duration of the so-called *ad hoc traffic indication message* (ATIM) window. Any node that has data to transmit announces its intention using the ATIM frame, which is subsequently acknowledged by the corresponding receiver. These transmissions are performed using the regular CSMA/CA mechanism of IEEE 802.11. Nodes that are about to receive data stay awake throughout the beacon interval, and so do the transmitting nodes that have received the proper acknowledgment. Other nodes can doze off—that is, they can switch to a low power state until the next beacon. In the original 802.11 standard (ANSI/IEEE 1999), the duration of the ATIM window is fixed, which does not give optimal results. In other words, an ATIM window that is too short means that not all nodes that have data will succeed in announcing and actually performing their transmissions, which degrades throughput. On the other hand, an ATIM window that is too long leads to higher energy consumption because all nodes must remain awake throughout this time interval, which leaves too little time for actual data transmissions and thus degrades throughput at high loads.

In DPSM, this power-saving mechanism is enhanced in several ways, all of which aim to improve power efficiency. First, ATIM windows of variable duration allow

the network to adjust to traffic conditions; in addition, each node can choose its own ATIM window size. Second, a single ATIM frame is to be used per destination node, thus reducing contention during the ATIM window. However, each data packet must contain information about the remaining number of packets to be sent, which allows the receiver to determine whether all packets were received or not. Third, a node that finishes a data transmission, either as the transmitter or the receiver, is allowed to doze off until the next beacon interval, which improves energy efficiency.

Dynamic adjustment of the ATIM window size uses several criteria: the number of packets pending transmission, the information overheard from other nodes' ATIM frames (each ATIM frame contains the information about the current contention window size of the transmitting node), the ATIM frames received while waiting for a previously announced data transmission, and the information on the number of retries for the current ATIM frame. In this manner, the DPSM scheme can be fine-tuned to improve energy efficiency without affecting throughput.

Power Control Mechanism

The power control mechanism of Jung and Vaidya (2005) uses the basic scheme described above but with the following change: The sender changes its power level from minimum to maximum during the transmission of a data packet, as shown in Figure 18. The duration of the maximum level transmission is chosen to be long enough for the nodes in the vicinity to sense the ongoing transmission and defer those of their own. In this manner, the risk of collisions is reduced (albeit not altogether eliminated), while the power consumption is reduced below that of the original IEEE 802.11 CDF.

MAC PROTOCOLS THAT USE DIRECTIONAL ANTENNAE

The protocols in the group use directional antennae to limit the spatial coverage of data transmissions and, consequently, reduce the risk of collisions. The underlying assumption is that the difference in received signal strength (i.e., the selection diversity) will allow the receiving subsystem to determine the approximate location of the correspondent (i.e., which antenna segment is to be used). The basic idea is simple: The sender sends a RTS packet in all directions and the receiver responds with an omnidirectional CTS, which informs all prospective neighbors about the forthcoming transmission. The sender transmits the data packet using a single antenna segment: the one that is oriented toward the receiver. In this manner, it will minimize its interference with its neighbors in all other directions. This scheme is depicted in Figure 19; it was first proposed by Nasipuri et al. (2000).

The problem with this approach is that it must be coupled with power control insofar as the directional transmissions must use reduced power levels to achieve a transmission range that is equal or close to the one obtained with omnidirectional transmission. In case the same power level is used for both types of transmissions, an increase in collisions in the coverage of the directional beam will be observed.

A similar protocol is described by Ko, Shankarkuman, and Vaidya (2000), as well as by others, the main difference being the assumption that the sender knows the location of the receiver and thus can use a directional RTS instead of the omnidirectional one.

An interesting variation of the basic protocol is described by Korakis, Jakllari, and Tassiulas (2003). In this case, the RTS packet is transmitted omnidirectionally but through a one-by-one antenna segment in a circular fashion, while the CTS packet is transmitted directionally after all directions have been covered by the RTS. The RTS packet contains the information about the duration of the forthcoming transmission and the segment it covers. Hence, the nodes that hear the RTS (but are not the designated receiver) can decide whether or not to defer their transmissions. The sender listens for the CTS omnidirectionally. The data and ACK packets are sent directionally. In this manner, the potential for interference (and resulting collisions) is reduced. A simple extension of the protocol allows the nodes to record the information about their neighbors' locations, which helps avoid certain hidden terminal scenarios that result from the use of directional antennae.

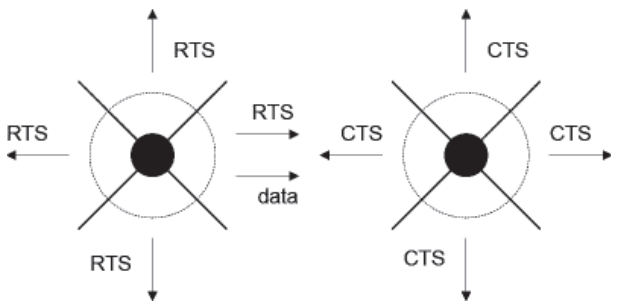

Figure 19: Packet transmission using a directional antenna with four segments

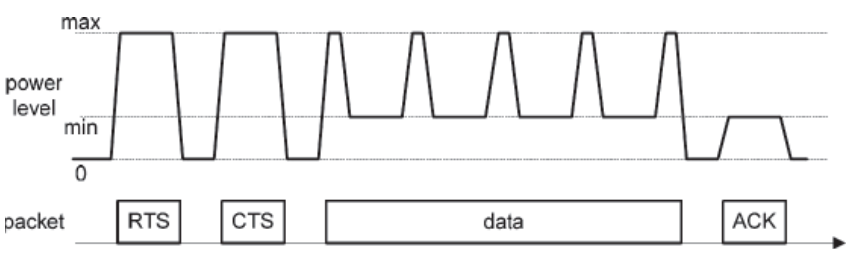

Figure 18: Power profile of a packet transmission in the PCM scheme

MAC PROTOCOLS THAT USE OUT-OF-BAND SIGNALING

Ensuring collision-free data transmissions requires that the potential interfering senders are informed about pending data transmissions in a timely fashion. The protocols described in the previous section use the control handshake before the actual transmission. Although this approach does help alleviate some of the problems that may cause collision, it is unable to completely eliminate collisions. Moreover, transmitting control packets uses bandwidth, and together with backoff windows of variable size reduces the overall bandwidth utilization of the network. The protocols described in this section adopt a different approach—namely, they use a separate channel (the control channel) to convey information that prevents collisions or to conduct the actual handshake. The idea has its origins in traditional telephony systems in which a special busy tone informs the party that wants to initiate a call about another transmission in progress.

Busy Tone Multiple Access

The busy tone multiple access protocol by Tobagi and Kleinrock (1975) noted earlier is the first reported work on the busy tone approach. In BTMA, a separate control channel is used to transmit a busy tone when the node is active. The busy tone could be as simple as a pure sine wave at a predefined frequency. Any node that wants to transmit will check the control channel first. If the busy tone is present, the transmission will be deferred for some random time; if the control channel is idle, the node starts transmitting the busy tone and simultaneously starts its data transmission. In the original proposal, the transmission power of the control channel is increased to ensure that the range of the busy tone is approximately twice that of the data transmission; a later modification uses the same transmission range for both channels but mandates that the nodes that sense the data transmission in progress forward the busy tone to other nodes in their transmission range. Either way, no other node within the two-hop transmission range is allowed to transmit. This approach is simple and effective—but perhaps too effective because many of the nodes prevented from transmitting could actually engage in communications of their own without interfering with the current transmission. Bandwidth utilization tends to be poor.

Receiver-Initiated BTMA

A derivative of the basic BTMA known as *receiver-initiated BTMA* (RI-BTMA), originally described by Wu and Li (1987), combines a time-slotted approach akin to TDMA with a much simpler handshake protocol that employs a single busy tone. In this case, the prospective sender listens to the control channel until it finds a free slot—that is, the one without a busy tone. Once such a slot is found, it sends a small preamble packet on the data channel. (The time slot on both data and control channels is equal to the duration of the preamble packet.) The designated receiver responds by activating its busy tone on the control channel. On hearing this tone, the sender may begin the data-packet transmission. Once the packet is received, the receiver turns off the busy tone.

Dual BTMA

Deng and Haas (1998) have proposed an improved version of this approach, called the *dual busy tone multiple access* (DBTMA). In this protocol, the data channel is used exclusively for data transmissions, whereas the control channel is used for the RTS-CTS handshake and for two busy tones that indicate a reception and a transmission in progress, respectively.

The DBTMA protocol operates as follows. A node that wants to transmit a packet checks the control channel for the presence of a receiving busy tone; if none is detected, the node sends the RTS packet to the receiver on the control channel. On receiving the RTS packet, the receiver checks the control channel for the presence of a transmitting busy tone, which would indicate that a transmission is already in progress in the vicinity. If none is present, the receiver responds with a CTS packet via the control channel and then activates its receiving busy tone to alert other nodes in its vicinity that it is receiving a data packet. On receiving the CTS packet, the sender activates the transmitting busy tone on the control channel and begins the data transmission on the data channel. When the data-packet transmission is finished, the sender turns off the transmitting busy tone; the receiver turns off the receiving busy tone after it has received the data packet.

The DBTMA protocol is able to achieve almost twice the value of channel utilization of the basic BTMA or some RTS- and CTS-based protocols such as MACA or MACAW, mainly because the use of two busy tones blocks only the transmissions from nodes in the vicinity of the receiver but not those in the vicinity of the sender.

The main problem with all protocols based on the busy tone approach is the need for two radios because the busy tone uses a different frequency or frequencies from the data channel. This requirement may not be simple to satisfy, in particular in conjunction with other requirements such as small physical size and limited energy source. On account of this, MAC protocols that utilize a separate control channel with busy tones have not been especially successful in practice.

MAC PROTOCOLS THAT USE POLLING

Bluetooth

Bluetooth is among the few true polling protocols used in ad hoc networks (Bluetooth SIG 2004), thus its adoption as the IEEE standard 802.15.1 (IEEE 2002). Bluetooth devices are organized in small networks known as *piconets*, with as many as 255 devices; one acts as the master and as many as seven others can be active at any given time; the remaining ones are parked. The channel time is divided into time slots of $T = 625$ μs, and all communications in the piconet are synchronized to this clock. Bluetooth uses a variant of the TDMA protocol in which all communications are performed under the control of the piconet master; in fact, all communications in the piconet must pass *through* the master. The master polls the slaves by sending them packets with appropriate identification; slaves can

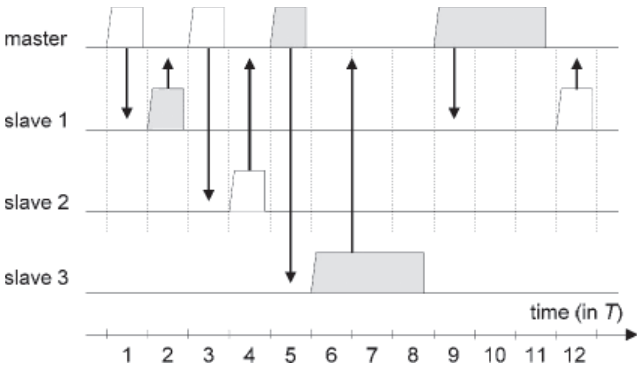

Figure 20: TDD communication in a Bluetooth piconet

talk back to the master only when addressed and only immediately after being addressed by the master. The operation of this protocol is schematically shown in Figure 20.

Master (downlink) and slave (uplink) transmissions occur in alternative slots in a scheme known as *time-division duplexing* (TDD). The Bluetooth TDD protocol requires that an addressed slave must respond to the master's poll even when it has no data to send; a similar requirement holds for the master as well (in Figure 20, white symbols denote empty packets while gray symbols denote packets that carry data). An important consequence of the TDD protocol is that both the throughput and end-to-end delays critically depend on the polling discipline—that is, the algorithm used by the master to poll active slaves in the piconet. Several such disciplines have been proposed and analyzed in recent years; a detailed overview and analysis can be found in Mišić and Mišić (2005).

The TDD protocol is collision-free and, in theory, should be more energy-efficient than the collision-based MAC protocols, but the advantage is not so noticeable in practice. The main source of inefficiency of the protocol is the fact that even empty packets use bandwidth: They last for one slot T, whereas data packets can last for one, three, or five slots T. Furthermore, all slaves must listen to master transmissions at all times and respond when polled, even when there is no data to transmit in either direction. Special modes defined by the official Bluetooth standard allow slaves to detach temporarily from the piconet to minimize energy consumption or perform other tasks (Bluetooth SIG 2004).

SENSOR NETWORKS

Sensor networks are a class of wireless networks intended for monitoring environmental phenomena in a given physical space; such networks find increasing usage in areas as diverse as military applications, object surveillance, structural health monitoring, and agriculture and forestry, among others. Monitoring may be continuous, with a prescribed data rate that may change over time; it may also be triggered by an explicit demand from a controlling node or a specific event in the environment. Environmental phenomena to be monitored include simple physical variables such as temperature, humidity, light, pressure, pH value, and the like; however, other phenomena also can be monitored, such as the presence or absence of a specific object (say, an inventory item with a radio frequency ID tag), or movements of persons and objects (e.g., cars). The spaces to be monitored include rooms, hallways, foyers, homes, backyards, streets, and larger buildings and structures (e.g., bridges) as well as open spaces such as fields or forests. Sensor nodes can be deployed in large numbers—from tens through hundreds to even thousands. Sensor networks are often expected to operate autonomously, with little or no human intervention, for prolonged periods of time. Sensor nodes are seldom mobile, and even when mobility is present, not all of the nodes are equipped with appropriate capabilities. Given such a diverse set of applications and requirements, it should come as no surprise that the constraints that guide the design and deployment of wireless sensor networks differ, sometimes substantially, from those that hold in wireless ad hoc networks (Sohrabi et al. 2000; Achir and Ouvry 2004). Let us now discuss those constraints in more detail.

Energy Efficiency

Probably the most important difference results from the fact that sensor nodes typically operate on limited battery power, which means that the maximization of network lifetime (and, consequently, minimization of power consumption) is sine qua non for sensor networks. On the contrary, power consumption is seldom the critical requirement for ad hoc networks.

The constraint of minimal energy consumption translates into two distinct yet closely related design requirements (Jones et al. 2001):

1. The communication efficiency must be maximized through the design of simple yet flexible and effective communication protocols and functions.
2. Those protocols and functions must be implemented by small chips with limited computational and memory resources. Simultaneous achievement of these objectives necessitates some kind of cross-layer protocol optimization in which the MAC layer would use the information obtained from the PHY layer to control its own operational characteristics. At the same time, optimal operation of the upper, network, and transport layers requires the knowledge of appropriate information from both the PHY and MAC layers. Again, such tight integration is not common in ad hoc networks.

An important consequence of the requirement for energy efficiency is the limited transmission range of most sensor node radio subsystems; few real devices have a transmission range of more than 100 meters (300 feet), and ranges of 10 meters (30 feet) and even less are not uncommon.

Protocol Efficiency

Regarding communication protocols, the main sources of inefficiency are packet collisions, but also overly complex handshake protocols, receiving packets destined for other nodes, and idle listening to the medium (Singh and Raghavendra 1998; Ye, Heidemann, and Estrin 2004). Actual power consumption of sensor nodes, often called *motes*, depends mostly on the radio subsystem and its operating mode. In most (but not all) cases, transmitting

uses 25 percent to 100 percent more energy than receiving; idle mode in which the radio is turned on but does not transmit or receive consumes some 10 percent to 20 percent less energy than receiving (Stemm and Katz 1997; Bhardwaj and Chandrakasan 2002). However, most savings can be made by putting the node to sleep, when power consumption drops by one to two orders of magnitude, depending on the hardware (Jung and Vaidya 2002; Van Dam and Langendoen 2003).

Use of Redundant Sensors

Because nodes are small and cheap to produce and the network lifetime needs to be maximized, it is often feasible to deploy the sensors in a given physical space in much larger numbers than necessary to obtain the desired rate of information flow. If redundant sensors are used, they can be periodically sent to sleep to minimize their duty cycle, which extends the lifetime of individual sensors and of the entire network and reduces or eliminates the need for operator intervention, thus reducing the operational cost of the network (Akan and Akyildiz 2005). The use of redundant sensors has profound implications on the design of MAC protocols, as will be seen below.

Node Specialization

Another important distinction is related to the role of individual nodes. An ad hoc network allows its nodes to choose the specific role, or roles, they would like to play—data source, destination, or intermediate router—at any given time. In most cases, a node is free to switch to a different role or roles whenever it finds it appropriate or is instructed to do so by the specific application currently executing on it. On the contrary, nodes in a sensor network have specific roles that either do not change often or never change at all. Most of the nodes act as sensing nodes, some act as intermediaries that route the traffic and (possibly) performs some administrative duties, and a small number of nodes (sometimes only a single node) act as the network sink (or sinks) toward which all the sensed data ultimately flows (Akyildiz et al. 2002). A group of sensor nodes under the control of an intermediary is sometimes referred to as a *subnetwork* or *cluster*, while the intermediary itself is known as *cluster-head*. We note that the number of intermediate levels interposed between the sensing nodes and the network sinks depends on several variables such as the size of the network, the size of the physical space that the network has to monitor, the transmission range of individual nodes, and (to some extent) the actual MAC protocol used.

Traffic Characteristics

The traffic in sensor networks is rather asymmetric because the bulk of it flows from the sensing nodes toward the network sink (this is often referred to as the *uplink* direction). The traffic in the opposite direction is generally much smaller and consists of control information and possibly queries issued by the network sink on behalf of the corresponding sensing application (Intanagonwiwat, Govindan, and Estrin 2000). Furthermore, traffic patterns in sensor networks are rather different than in ad hoc networks. For example, temperature or humidity monitoring might require periodic or nearly periodic transmissions—in essence, synchronous traffic with low data rate—whereas object surveillance and other event-driven sensing applications exhibit low average traffic volume and random bursts with considerably higher peak rates.

Furthermore, data packets are often much smaller in sensor networks. Original data from sensing nodes typically consists of only a few data values reported by appropriate sensors. Intermediate nodes may choose to aggregate those values to improve energy efficiency and reduce bandwidth and energy consumption; data aggregation is more common in networks with a larger number of hierarchical levels. At the same time, the number of sensor nodes and their spatial density may be extremely large, depending on the size of the space to be monitored and the requirements of the sensing application.

Quality-of-Service Requirements

Maintaining prescribed delay bounds in a network of resource-constrained nodes with limited transmission range is a complex issue. Delay considerations are of crucial importance in certain classes of applications—for example, in military applications such as battlefield communications and detection and monitoring of troop movement, or in health care applications where patients in special care units must be monitored for important health variables (via ECG or EEG) because of a serious and urgent medical condition. Low delays can be achieved either by bandwidth reservation, as utilized in variations of the TDMA approach, or by some kind of admission control that will prevent network congestion, if the CSMA approach is used. At the same time, the requirement for maximum throughput is relaxed because of the following. First, the exact value of the throughput requirement is usually prescribed by the sensing application, unlike general networks where the goal is to obtain as much throughput as possible. Second, energy efficiency dictates the use of protocols that incorporate power control, which will strive to keep the nodes inactive for as long as possible (Akan and Akyildiz 2005). To obtain the desired throughput, it suffices to adjust the mean number of active nodes.

Even packet losses can be catered to in this manner because we do not care whether a given packet from a given node will reach the network sink—as long as the sink receives a sufficient number of packets from other nodes. Any packet loss can be compensated for (in the long term) by varying the mean number of active nodes. In a certain sense, fairness is not needed at the node and packet level as long as it is maintained at the cluster level (Callaway 2004). On the contrary, fairness at the node and packet level is important in ad hoc networks.

Differences from Ad Hoc Networks

The requirements outlined above lead to several important differences between sensor networks and ad hoc networks, most notably the following:

- Power efficiency and lifetime maximization are the foremost requirements for sensor networks.

- Self-organization is important in both ad hoc and sensor networks. In the former case, this is because of dynamicity and node mobility, which cause frequent topology changes and makes self-organization more difficult; in the latter, this is mostly caused by sensor nodes exhausting their battery power (i.e., dying), although mobile sensors are used in some applications.
- Throughput maximization is often required in ad hoc networks but is not too common in sensor networks.
- Delay minimization is typically assigned much higher priority in sensor networks than in their ad hoc siblings.
- The use of redundant sensors allows for a certain level of fault tolerance; on the contrary, packet losses are intolerable in ad hoc networks.
- Scalability is an important issue because of the potentially large number of sensors; scalability is also important in ad hoc networks, but it is limited by the available bandwidth and the desired throughput.
- Nodes in ad hoc networks are often mobile, whereas most sensor networks have no mobile nodes.

In more than one sense, wireless ad hoc networks are a class of networks with flexible topology but without infrastructure, which should cater to all kinds of networking tasks. On the other hand, sensor networks are highly specialized networks that perform a rather restricted set of tasks under severe computational and communication restrictions.

MAC Protocols for Wireless Sensor Networks

The requirements and constraints outlined above mean that the design of wireless sensor networks and their associated protocols is a rather challenging task. The discussions that follow will present several MAC protocols for wireless sensor networks and highlight the conceptual approaches in which they attempt to address those challenges. Similar overviews can be found in Akyildiz et al. (2002) and, more recently, in Demirkol, Ersoy, and Alagöz (2006). Furthermore, an interesting overview of some of the protocols (and a detailed description of others) can be found in Sohrabi et al. (2000).

In the following, we will briefly present a couple of representative MAC protocols for sensor networks. Second, we present the recently adopted IEEE 802.15.1 and IEEE 802.15.4 standards as the industry standards for *wireless personal area networks* (WPANs) that hold great potential for *wireless sensor network* (WSN) application. We agree with the opinion expressed by Callaway (2004) that "the success of wireless sensor networks as a technology rests on the success of the standardization efforts to unify the market and avoiding the proliferation of proprietary, incompatible protocols that, although, perhaps optimal in their individual market niches, will limit the size of overall wireless sensor market."

Adaptive Rate Control with CSMA

Woo and Culler (2001) have augmented the basic CSMA protocol with adaptive rate control to improve energy efficiency and so-called multihop fairness—that is, in multihop scenarios, fairness can be measured as the balance between the traffic that originates in the node itself and the traffic that the node has to relay on behalf of others (route-thru traffic). Note that the traffic generated by any given node is, in fact, the route-thru traffic for all other nodes it has to pass through.

In the ideal case—a network with symmetric traffic—the knowledge about the total number of size of the network would help the node to estimate the allowed amount of its contribution. Because this information is hard or even impossible to obtain, an adaptive mechanism with a linear increase and multiplicative decrease is used to control the transmission rate of an application. The goal for any given node is to ensure fairness among all nodes whose traffic it routes. Moreover, because dropping the route-thru traffic is a waste of resources (it will have to go through that same route again or be lost), such traffic is given preference over the locally generated one. Measures are also taken to reduce the impact of the hidden node problem for pairs of nodes that are two hops away yet able to hear one another, for which Woo and Culler (2001) use the terms *child* and *grandparent nodes*.

When used in conjunction with a random delay before transmission, the adaptive rate mechanism provides an effective control mechanism without explicit control packets, in particular in scenarios where traffic loads are low, which is common in many sensor network applications.

S-MAC

The S-MAC protocol is designed for deployment in multihop WSN, where packet destinations are uniformly distributed (Ye, Heidemann, and Estrin 2004). Each device alternates between active periods, in which it communicates with other nodes, and inactive or sleep periods that ensure energy efficiency by keeping the duty cycle low. A complete cycle that includes an active and an inactive period is referred to as a *frame*. The information about the identity of the sensor node and its next scheduled sleep time is called an *activity schedule*.

The main feature of S-MAC is the self-synchronization of activity schedules for different sensor nodes, which is achieved as follows. On returning from sleep, a node listens to the medium for a specified time to check for other nodes' schedules. If a schedule is received, it is immediately adopted and followed; if not, the node chooses a schedule of its own and starts to follow it. Either way, it will start broadcasting SYNC packets with the information about the adopted schedule. However, if a node receives a different schedule at a later time, it may drop its own schedule and adopt the new one. Alternatively, it can add the new schedule to its own, which facilitates multihop communications between nodes that cannot communicate directly because of limited transmission range. A node periodically listens to SYNC broadcasts from other nodes; this lessens the risk of neighbor nodes following completely different schedules (and missing each other's active periods) and facilitates schedule reconfiguration when the network topology changes. In this manner, nodes collect information about their neighbors and build local tables of active periods of neighbors. Each node uses its table to schedule its sleep periods with the goal of being awake simultaneously

with some of its neighbors so that packet exchange can take place. In fact, *virtual clusters* are formed by the nodes that follow the same schedule; this facilitates communication among them.

During the active period, a node either listens or transmits a data packet. The listening period is organized in two phases that accommodate SYNC broadcasts and data transmissions, respectively. Packet transmission is achieved using slotted CSMA with RTS–CTS handshake. Before the node sends a SYNC or RTS packet of its own, it listens to the medium for a random period of time. After the transmission of a RTS packet, node waits for the CTS packet and then transmits the data packet. If the CTS response does not arrive during the listening period, then the node goes back to sleep and tries again in the next active period.

The delay in multihop transmissions can be extremely long if each node along the route receives the packet in one active period and transmits it in the next one. To reduce this delay, nodes in S-MAC use the so-called adaptive listening: A node that overhears the neighbor's transmission (RTS or CTS) will stay awake beyond the end of the active period. In this manner, if this node is the next destination of the packet, it can receive it immediately rather than after going through the sleep period.

To reduce transmission latency, a node that has more than one packet to send (a burst) can use only one control handshake per burst. This feature is referred to as *message passing* (Ye, Heidemann, and Estrin 2004). However, explicit acknowledgments are required after each successfully transmitted packet, which somewhat offsets the power savings obtained by spreading the cost of the handshake over all packets in the burst. If an ACK packet is not received, the sender will retry the transmission of the last packet; if the maximum number of retries is reached, the sender will abort the transmission and start over. Acknowledgments ensure reliable transmission but also inform other nodes about the transmission in progress and thus help avoid the hidden terminal problem. In other words, both data and ACK packets have a duration field that specifies the total transmission time for the burst, including the ACK packets. Therefore, nodes that hear either a data packet or an ACK packet from an ongoing burst transmission may go to sleep until its scheduled finish time.

Although simple and efficient, S-MAC suffers from scalability problems. As the size of the network increases, it becomes increasingly difficult for a node to maintain a coordinated schedule, and the power consumption of relaying nodes—which have to receive and transmit packets—increases. As a result, energy efficiency deteriorates. Decreasing the duty cycle may counter this trend, but it increases the delays and reduces the throughput.

Time-Out MAC

The main drawback of the S-MAC is the fixed duty cycle—that is, the ratio of the active period to the entire frame time, which does not depend on the traffic in the network. To reduce the power consumption even more, active periods should be as short as possible under the given traffic pattern. This is the main concept of the *time-out MAC* (T-MAC) proposed by Van Dam and Langendoen (2003). The T-MAC borrows the concept of distributed synchronization from the S-MAC, except that nodes that hear a schedule different from their own must follow both (this is not mandatory in S-MAC). This provision ensures that neighboring nodes will always be able to communicate.

T-MAC nodes can optionally go to sleep when they hear an ongoing transmission of which they are not a party. Although this forced sleep (which is mandatory in S-MAC) may save energy, it also leads to increased collision overhead and reduced throughput (Van Dam and Langendoen 2003).

In T-MAC, a node keeps listening and (perhaps) transmitting as long as it is active. It will switch to sleep if no activation event has occurred for a specified time *TA*. Activation events include activity on the medium, including end of own transmission or acknowledgment, and the firing of an activation timer (which brings a sleeping node back to the active state). The time-out value *TA* is chosen to ensure that the node does not miss any communication directed to it. Nodes that have data packets must attempt transmission as soon as their active period starts; other nodes in its neighborhood will be awake and, hopefully, able to receive the data. A potential sender listens to the medium for a random time within a fixed contention interval and then undertakes the RTS–CTS handshake.

As noted earlier, many sensor networks exhibit asymmetric traffic patterns: The bulk of the traffic is directed toward the network sink. In this case, it is possible that some transmissions will be delayed until the next active period simply because one of the relaying nodes was unaware of it and went to sleep (the early sleeping problem) as shown in Figure 21. To overcome this problem, the node that overhears a CTS packet destined for another node can send a *future request to send* (FRTS) packet to inform the future destination node about the forthcoming data transmission. The FRTS packet contains the duration of the transmission copied from the CTS packet; the destination node can then stay awake for that long. This mechanism, shown in Figure 22, reduces the delay and increases the overall throughput of the network.

Another possible problem in multihop relaying is that an intermediate node that has a nearly full buffer with data to forward receives the RTS packet announcing a forthcoming data transmission from another sender. Accepting that data increases the risk of buffer overflow and data loss; in addition, it may increase the delay because of the early sleeping problem. The T-MAC protocol allows such a node to give priority to its own transmission by sending a RTS packet to the proper destination rather

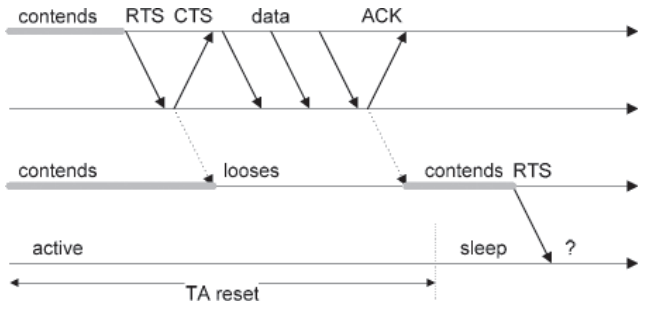

Figure 21: Early sleeping problem in multihop communication

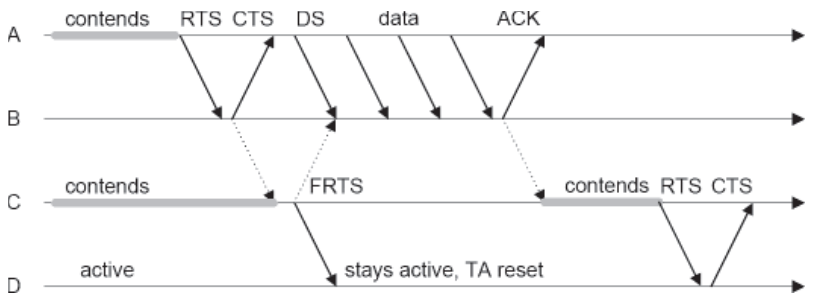

Figure 22: The use of FRTS packet avoids early sleeping

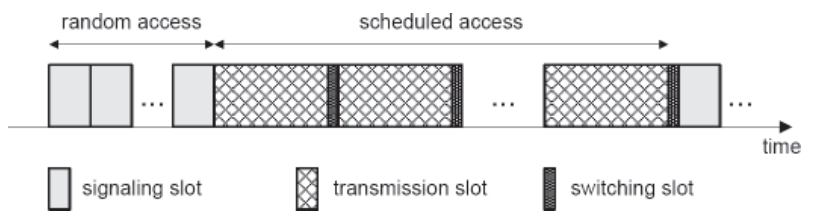

Figure 23: Time slot organization in TRAMA

that responding with the CTS packet as usual. In fact, such a node takes over the medium, which it has effectively won anyway, only for a different purpose.

Although this technique, known as *full-buffer priority*, can offer substantial improvement in performance, it may be risky to use under heavy loads: If too many nodes start taking over the medium, the risk of collisions rapidly increases. To avoid such situations, the node is allowed to use full-buffer priority only if the number of times it has lost contention exceeds the predefined threshold.

Traffic-Adaptive Medium Access

Although the T-MAC protocol is able to offer considerable power savings compared to S-MAC under variable workloads, it suffers from similar problems related to scalability, especially under heavy loads. A possible solution to keep the energy efficiency under control is to reduce the timeout value, but this increases delays and impairs the ability to react to changing network conditions. The *traffic-adaptive medium access* (TRAMA) protocol proposed by Rajendran, Obraczka, and Garcia-Luna-Aceves (2006) attempts to overcome this problem through a traffic-adaptive scheme that selects receivers according to announced transmitter schedules. It is based on an earlier protocol known as *node-activation multiple access* (Bao and Garcia-Luna-Aceves 2001) has employed a similar approach but without concern for energy efficiency.

TRAMA assumes a single, time-slotted channel for both data and control transmissions. Time is divided into periods of random access, interleaved with periods of scheduled access, as shown in Figure 23.

Three subprotocols are used to exchange neighbor information (*neighbor protocol*, or NP), exchange schedule information (schedule exchange protocol), and elect the transmitters and receivers for the current time slot (adaptive election algorithm). The network starts in random access period wherein each node transmits basic signaling information in a randomly selected slot. This period essentially serves to construct and subsequently maintain the information about the node's neighbors. Although collisions of signaling packets are possible, the duration of the random access period is chosen to guarantee consistent neighbor information with a high degree of confidence.

Each node calculates its schedule interval based on the traffic pattern of the application and the neighbor information obtained in the NP phase. In addition, it calculates the number of slots in the subsequent schedule interval for which it has the highest priority among its two-hop neighbors (the winning slots). This information, together with the designated receivers for those slots (i.e., the packets to be transmitted therein), is announced through schedule packets. Priority is determined according to node identity and the time slot in question; ideally, this process assigns unique (and globally known) priorities to each node and time slot combination. A node may give up its slot if there are no data packets to transmit; such vacant slots can be used by other nodes in the neighborhood. Some of its winning slots will go unused if the number of such slots exceeds the number necessary to transmit all the data in that node's queue. The ChangeOver slot denotes the slot after which all winning slots are unused; it is used for announcing the next schedule, and all of the nodes must listen to it. A node that has no data to send and has not been designated as the receiver may switch to low-power mode until the next ChangeOver slot.

Although the scheduling mechanism used in TRAMA ensures collision-free data transmissions and allows for longer sleep times than in other protocols such as S-MAC, some inefficiency is caused by the fixed size of the transmission slots, which imposes a lower limit on the duty cycle. Another problem is that the priority calculations must be repeated for each slot in every scheduled access period, which may cause problems in a network implemented with resource-constrained nodes.

Other MAC Protocols for Sensor Networks

In the following, we will briefly mention several other protocols for wireless sensor networks, both those that focus entirely on MAC layer issues and those that include issues related to other layers as well. The reader should note that a several subsequent chapters are devoted to

issues of broadcasting, routing, data gathering, and localization in sensor networks.

Low Energy Adaptive Clustering Hierarchy

Heinzelman, Chandrakasan, and Balakrishnan (2000) have attempted to minimize energy consumption in the scenario in which the sensor networks consists of several identical, resource-constrained nodes that generate unidirectional traffic sent to a single network sink. Although this protocol, known as *low energy adaptive clustering hierarchy* (LEACH), focuses mostly on routing issues, it does integrate elements from both network and MAC layer, which is why a brief overview is included here.

The main idea of the LEACH algorithm is that the sensors should autonomously organize themselves into clusters, with one node acting as the cluster-head. Sensors elect themselves to be the cluster-heads with a certain probability, which depends on the remaining energy of the sensor and the number of sensors in the entire network. Becoming the cluster-head is a unilateral decision and no negotiation is involved. To balance the power consumption, the role of the cluster-head is randomly rotated among the sensors in the cluster.

Once elected, the cluster-head advertises its presence. All sensor nodes must listen to the advertisements. They decide which cluster they will join and inform the corresponding cluster- head. All transmissions so far use CSMA. When the cluster-head gathers sufficient knowledge about the sensors in its cluster, it creates a TDMA transmission schedule and broadcasts it. (The original proposal assumes that CDMA coding is used for intracluster communication so that collisions can be avoided.) Once a sensor knows its cluster and its transmission schedule, it may go to sleep until the time comes for its transmission, thus conserving the least amount of power.

When the cluster-head receives the data from all the sensors in the cluster, it aggregates it and sends it to the network sink. A new round of clustering then follows.

Ideally, the LEACH algorithm should result in energy levels of individual sensors being used up at about the same rate. Of course, the cluster-head must remain active throughout the entire cycle, but the rotation of the cluster-head role should ensure that the load is evenly balanced among all nodes.

Wave Scheduling

Wave scheduling, originally proposed by Trigoni et al. (2004), is an example of an integrated protocol that spans the MAC, network, and even portions of the transport layer of the traditional networking protocol stack (Stallings 2002). It partitions the network into cells such that the nodes in a cell are assumed to be able to communicate only with the nodes in their own cell and its immediate neighbors. In fact, the assumption is that the cells are near rectangular, although this is not necessary for the proper functioning of the algorithm.

Transmissions are then scheduled utilizing a sequence of activations of edges that connect two cells. Each activation period includes a contention-based period followed by a contention-free period. In the contention period, the GAF scheduling protocol (Xu, Heidemann, and Estrin 2001) is run locally in each cell: The nodes attempt to determine whether the current leader has sufficient energy to continue its leadership role. If this is not the case, then a new leader is elected and messages queued for delivery are handed over by the previous leader, together with intercell routing information. The remaining nodes send the sensed data to the leader. Ordinary nodes can then go to sleep until the next activation period. In the contention-free period, accumulated messages are transferred from one cell leader to the next one. A special message informs the receiver if there are no messages to deliver, in which case both leaders can go to sleep to conserve energy.

Two distinct wave schedules are defined. In the *simple wave* schedule, messages are coordinated in east–west and north–south directions. This allows simultaneous transmissions from the leaders when the distance between them exceeds the transmission range. North–south and east–west schedules occur in interleaved fashion. In the *pipelined wave*, the graph obtained by the cell-to-cell edges is partitioned into a collection of maximal independent sets that can then be activated simultaneously without interference. Both wave schedules avoid interference, and they can be tuned to achieve minimum energy or minimum delay routing.

Stationary MAC and Startup

The *stationary MAC and startup* (SMACS) procedure, originally proposed by Sohrabi et al. (2000), deals with the creation of a suitable TDMA-like schedule without a centralized authority. In SMACS, neighbor discovery and channel assignment phases are lumped together, and channels for node-to-node communication are assigned immediately after the two nodes involved learn about each other's existence. By the time all nodes find out about all of their neighbors, an operational network with a flat topology is formed. *Channels*, in the SMACS terminology, are in fact time slots within a superframe of sufficient duration. To reduce contention, it is advisable to assign different frequencies to different channels whenever possible.

Berkeley MAC

A rather different approach was proposed by Polastre, Hill, and Culler (2004). Instead of trying to optimize the behavior of the MAC protocol, the *Berkeley MAC* (B-MAC) protocol optimizes the low-power listening and clear channel sensing on a popular Mica2 hardware (Hill et al. 2000). Other medium access mechanisms that use these primitives can then be implemented on top of the B-MAC platform.

Zebra MAC

The *zebra MAC* or Z-MAC (Rhee et al. 2005) combines the strengths of the CSMA and TDMA approaches. A node can transmit in any time slot after successful carrier sensing, as in CSMA, but the designated owners of that particular slot take precedence. This concept reduces the risk of collision while still allowing other nodes to transmit in that slot. In addition, lightweight synchronization schemes are utilized to provide the network with some resilience to topology changes and control information loss. However, the ZMAC protocol does not consider the situation in

which individual nodes go to sleep, which severely limits its usability in several sensing applications.

IEEE 802.15.4

The recent IEEE 802.15.4 standard (IEEE 2003b) is not specifically intended for use in wireless sensor networks—it was originally designed as a *low data rate WPAN* (LR-WPAN)—but its simplicity and low data rate make it an attractive choice for sensing applications as well (Callaway 2004). The IEEE 802.15.4 protocol is a full-function MAC and PHY protocol that requires other layers above it to function properly, in contrast with most other proposals, which simplify several layers into a single, integrated one.

An IEEE 802.15.4 network can operate in two modes: (1) the beacon-enabled, slotted CSMA/CA mode in which a dedicated coordinator must be present; and (2) beaconless, unslotted CSMA/CA similar to the 802.11 DCF in its basic form (without the RTS–CTS handshake). In the former case, the interval between two successive beacon frames is split between an active and inactive period, which allows the entire network to conserve energy by going to sleep. Also, a portion of the active part of the superframe can be reserved for scheduled access, which is allocated on request. However, requests to allocate such time must be made through packets that undergo contention. The performance of IEEE 802.15.4 networks at the MAC level is not fully explored yet, especially their suitability for low-power applications, although some results have recently been reported (Mišić, Shafi, and Mišić 2006).

CONCLUSION

As the discussion above shows, the area of MAC protocols for ad hoc and sensor networks does not suffer from any shortage of research results. Nonetheless, there are still many open issues that need to be addressed, mainly in the domain of sensor networks.

First, the selection of the actual protocol is still largely unsolved, even though most proposals use some variant of contention-based access derived from the ubiquitous CSMA approach. In addition, the problems of hidden and exposed terminals still plague most, if not all, of the proposed protocols.

Second, the simultaneous minimization of energy consumption and optimization of different performance metrics of the MAC protocol requires more research. Although some proposals do assume a specific structure of the sensor field and try to find the optimum solution for that particular case, more generic solutions are needed.

Third, synchronization problems are also difficult, in particular when sensor nodes sleep for prolonged intervals and wake up only to find that their clock has drifted away from that of the nodes that are active. Although several algorithms exist that tackle this problem, solutions that would be generic enough to fit most sensing applications have yet to be found.

From the practical perspective, we can expect further reductions in the physical size and consumption of sensor devices. Advances in integrated circuit technology will allow more complex protocols to be implemented. However, fundamental problems related to transmission range and power will still remain. In particular, severe resource constraints mean that tight integration and cross-layer design that encompasses the functionality of PHY, MAC, routing, and (possibly) data aggregation layers will be a practical necessity.

These and many other questions will most likely be answered with further applications of wireless sensor devices.

GLOSSARY

Backoff: Random delay introduced after a node or device that wants to transmit data has detected that the medium is not idle. Often used synonymously with binary exponential backoff.

Bandwidth Reservation: A procedure through which a portion of bandwidth is reserved for exclusive use by a link (i.e., for the transmission from one node to another) before the actual transmission takes place.

Beacon: A special packet transmitted periodically by the network coordinator in order to facilitate synchronization of other nodes and devices in the network.

Binary Exponential Backoff: Backoff procedure in which the range to choose the actual backoff duration from doubles after every unsuccessful clear channel assessment (in some protocols) or attempt to transmit data (in others).

Burst: A sequence of packets transmitted in short succession, often generated by segmenting a longer packet from a higher protocol layer.

Channel Hopping: Switching through available channels (in most cases, different frequencies in the designated radio frequency band) in a defined, often pseudorandom manner, with the goal of reducing the impact of interference and noise.

Clear-to-Send (CTS): Control packet sent by the designated data receiver to indicate its readiness to receive the data transmission from the potential transmitter. Usually sent in response to the RTS packet sent by the potential transmitter.

Clear Channel Assessment or Clear Channel Sensing: Listening to the channel before transmission to detect whether the medium is idle (i.e., used by another node or device).

Collision: Situation when two or more packet transmissions overlap in time and cannot be successfully received.

Contention: Situation in which two or more nodes or devices compete for medium access.

Coordinator: Node in the network that has special responsibilities in terms of monitoring or managing network operation.

Cluster: Group of nodes or devices that work together under the control of a central controller or coordinator referred to as a *cluster-head*. Commonly used for sensor networks.

Distributed Coordinator Function (DCF): One operational mode in an IEEE 802.11 compliant network in which contention arbitration is performed without a central controller or coordinator.

Duty Cycle: Ratio of the period in which a node is active to the period in which the node is inactive (or sometimes to the sum of active and inactive period durations).

Exposed Terminal: Node or device whose transmission may prevent a potential transmitter from sending the data even though there would be no collision at the receiver.

Fairness: Ability of the network to provide sufficient (and possibly comparable) level of service to all of its nodes.

Frame: Periodic time unit containing one or more packet transmissions and other activities by participants in a communications link; sometimes grouped to form a superframe and sometimes used to designate a packet or, more precisely, any data unit defined by a particular protocol.

Handshake: Exchange of control signals in order to make proper arrangement for subsequent data exchange.

Hidden Terminal: Node or device that is beyond the reach of the transmitter node, but whose transmissions can reach the receiving node and thus cause collision.

Interframe Spacing (IFS): Prescribed time intervals between specific transmissions in an IEEE 802.11-compliant network. Several different IFS intervals exist in the 802.11 standard.

Multihop: Communication between two nodes or devices through one or more intermediary nodes.

Out-of-Band: Separate channel or, in general, any mechanism distinctly different from the regular data channel; often used for the exchange of control information.

Piggybacking: Adding a small amount of extra information to a packet that has a well-defined purpose (e.g., data or control).

Point Coordinator Function (PCF): One operational mode in an IEEE 802.11-compliant network in which one of the nodes acts as a central controller or coordinator.

Polling: Procedure whereby a coordinator node queries other nodes in the network it controls to find out the amount of data they want to transmit and thus allocate bandwidth for those transmissions.

Request-to-Send (RTS): Control packet through which a node that wants to send data informs the designated receiver (and other nodes within its transmission range) about the intended data transmission. The receiver should respond with a CTS packet.

Round-Trip Time: Time interval from the beginning of a data packet transmission to the end of the subsequent acknowledgment packet.

Scalability: Ability of the network to operate without significant changes in performance in a wide range of network sizes.

Scheduling: General procedure of allocating resources (usually time) to different links (and corresponding nodes or devices). Also used to designate the mechanism whereby allocation is performed according to some predefined criteria, including but not limited to the volume and other characteristics of traffic.

Self-Healing: Ability of the network to operate when some of its nodes suddenly cease to function.

Signaling: Generic label for communication wherein only control information is exchanged (as opposed to data transmission).

Single-Hop: Direct communication between two nodes or devices without an intermediary.

Starvation: Situation in which a node has data to transmit but cannot gain access to the medium.

Superframe: Periodic time interval (sometimes referred to as a *cycle*) that contains several smaller units, often designated as *slots* or *slices*.

CROSS REFERENCES

See *Emerging Trends in Routing Protocols in Mobile Wireless Ad Hoc and Sensor Networks*; *Network Middleware*; *Network QoS*; *Principles and Applications of Ad Hoc and Sensor Networks*.

REFERENCES

Achir, M., and L. Ouvry. 2004. Power consumption prediction in wireless sensor networks. In *Proceedings of the Sixteenth ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Aug. 31–Sept. 2, Antwerp, Belgium.

Akan, Ö. B., and I. F. Akyildiz. 2005. ESRT: Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 13(5): 1003–16.

Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: A survey. *Computer Networks*, 38: 393–422.

ANSI/IEEE. 1999. *Standard for part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*. New York: IEEE.

Bahl, P., R. Chandra, and J. Duganan. 2004. SSCH: Slotter seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking* (ACM MobiCom'04), Sept. 26–Oct. 1, Philadelphia. pp. 216–30.

Bao, L., and J. J. Garcia-Luna-Aceves. 2001. A new approach to channel access scheduling for ad hoc networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking* (MobiCOM'01), July 16–21, Rome. pp. 210–21.

Bertsekas, D. P., and R. Gallager. 1991. *Data networks*. 2d ed. Englewood Cliffs, NJ: Prentice-Hall.

Bhardwaj, M., and A. Chandrakasan. 2002. Bounding the lifetime of sensor networks via optimal role assignments. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies* (IEEE INFOCOM 2002), June 23–7, New York. Vol. 3: 1587–96.

Bharghavan, V., A. Demers, S. Shenker, and L. Zhang. 1994. MACAW: A media access protocol for wireless LANs. In *Proceedings of the Conference on Communications Architectures, Protocols, and Applications* (ACM SIGCOMM '94), Aug. 31–Sept. 2, London. pp. 212–25.

Bluetooth SIG. 2004. Draft specification of the Bluetooth system, version 2.0.

Callaway, E. H. Jr. 2004. *Wireless sensor networks, architecture and protocols*. Boca Raton, FL: Auerbach Publications.

Deng, J., and Z. J. Haas. 1998. Dual busy tone multiple access DBTMA: A new medium access control for packet radio networks. In *Proceedings of International Conference on Universal Personal Communications* (IEEE ICUPC 1998), Oct. 5–9, Florence, Italy. pp. 973–7.

Demirkol, I., C. Ersoy, and F. Alagöz. 2006. MAC protocols for wireless sensor networks: A survey. *IEEE Communications Magazine,* 44(4): 115–21.

Fullmer, C. L., and J. J. Garcia-Luna-Aceves. 1995. Floor acquisition multiple access (FAMA) for packet radio networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (ACM SIGCOMM 1995), Aug. 22–6, Philadelphia. pp. 262–73.

Goodman, D. J., R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi. 1989. Packet reservation multiple access for local wireless communications. *IEEE Transactions on Communications*, 37(8): 885–90.

Heinzelman, W. R., A. Chandrakasan, and H. Balakrishnan. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Thirty-Third Annual Hawaii International Conference on System Sciences* (CD-ROM), Jan. 4–7, Maui, HI, USA.

Hill, J., R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. 2000. System architecture directions for networked sensors. In Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems, Nov. 12–5, Cambridge, MA, USA. pp. 93–104.

IEEE. 2002. *Standard for part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPAN)*. New York: Author.

———. 2003a. *Standard for part 15.3: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPAN).* New York: Author.

———. 2003b. *Standard for part 15.4: Wireless MAC and PHY specifications for low rate WPAN*. New York: Author.

Intanagonwiwat, C., R. Govindan, and D. Estrin. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking* (MobiCOM '00), Aug. 6–11, Boston. pp. 56–67.

Jain, N., S. R. Das, and A. Nasipuri. 2001. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the Tenth International Conference on Computer Communications and Networks* (IC3N), October, Phoenix. pp. 432–9.

Jian, S., J. Rao, D. He, and C. C. Ko. 2002. A simple distributed PRMA for MANETs. *IEEE Transactions on Vehicular Technology*, 51(2): 293–305.

Johansson, N., U. Körner, and L. Tassiulas. 2001. A distributed scheduling algorithm for a Bluetooth scatternet. In *Proceedings of the Seventeenth International Teletraffic Congress* (ITC'17), Sept. 24–8, Salvador da Bahia, Brazil. pp. 61–72.

Jones, C. E., K. M. Sivalingam, P. Agrawal, and J. C. Chen. 2001. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4): 343–58.

Jung, E.-S., and N. H. Vaidya. 2005. A power control MAC protocol for ad hoc networks. *Wireless Networks*, 11 (1–2): 55–66.

Kanodia, V., C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. 2004. Distributed priority scheduling and medium access in ad hoc networks. *Wireless Networks*, 8(5): 455–66.

Karn, P. 1990. MACA: A new channel access method for packet radio. In *Proceedings of the ARRL/CRRL Amateur Radio Computer Networking Conference*, September. pp. 134–40.

Ko, Y. B., V. Shankarkumar, and N. H. Vaidya. 2000. Medium access control protocols using directional antennas in ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications* (INFOCOM 2000), March 26–7, Tel Aviv. Vol. 1: 13–21.

Korakis, T., G. Jakllari, and L. Tassiulas. 2003. A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc'03), June 1–3, Annapolis, MD, USA. pp. 98–107.

Lin, C. R., and M. Gerla. 1999. Real-time support in multihop wireless networks. *Wireless Networks*, 5(2): 125–35.

Mišić, J., and V. B. Mišić. 2005. *Performance modeling and analysis of Bluetooth networks: Network formation, polling, scheduling, and traffic control*. Boca Raton, FL: CRC Press.

Mišić, J., S. Shafi, and V. B. Mišić. 2006. Cross-layer activity management in a 802.15.4 sensor network. *IEEE Communications Magazine*, 44(1): 131–6.

Nasipuri, A., S. Ye, J. You, and R. E. Hiromoto. 2000. A MAC protocol for mobile ad-hoc networks using directional antennas. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, Sept. 23–8, Chicago. Vol. 1: 1214–9.

Nasipuri, A., J. Zhuang, and S. R. Das. 1999. A multichannel CSMA MAC protocol for multihop wireless networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, September, New Orleans. pp. 1402–6.

O'Hara, B., and A. Petrick. 1999. *IEEE 802.11 handbook: A designer's companion*. New York: IEEE Press.

Perkins, C. E., ed. 2001. *Ad hoc networking*. Boston: Addison-Wesley.

Polastre, J., J. Hill, and D. Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems* (SenSys'04), Nov. 3–5, Baltimore. pp. 95–107.

Rajendran, V., K. Obraczka, and J. J. Garcia-Luna-Aceves. 2006. Energy-efficient, collision- free medium access control for wireless sensor networks. *Wireless Networks*, 12(1): 63–78.

Ram Murthy, C. and B. Manoj. 2004. *Ad hoc wireless networks, architecture and protocols*. Upper Saddle River, NJ: Prentice Hall.

Rhee, I., A. Warrier, M. Aia, and J. Min. 2005. Z-MAC: A hybrid MAC for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems* (SenSys'05), Nov. 2–4, San Diego. pp. 90–101.

Sanchez, J., R. Martinez, R. and M. W. Marcellin. 1997. A survey of MAC protocols proposed for wireless ATM. *IEEE Network*, 11(6): 52–62.

Shi, J., T. Salonidis, and E. W. Knightly. 2006. Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks. In *Proceedings of the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc'06), May 22–5, Florence, Italy. pp. 214–25.

Singh, S., and Raghavendra, C. S. 1998. PAMAS: Power aware multi-access protocol with signaling for ad hoc networks. *ACM SIGCOMM Computer Communications Review*, 28(3): 5–26.

Sohrabi, K., J. Gao, V. Ailawadhi, and G. J. Pottie. 2000. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5): 16–27.

Stallings, W. 2002. *Wireless communications and networks*. Upper Saddle River, NJ: Prentice Hall.

Stemm, M., and R. H. Katz. 1997. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications: Special Issue on Mobile Computing*, E80-B(8): 1125–31.

Talluci, F., M. Gerla, and L. Fratta. 1997. MACA-BI (MACA by invitation): A wireless MAC protocol for high speed ad hoc networking. In *Proceedings of the IEEE Sixth International Conference on Universal Personal Communications* (ICUPC 1997), Oct. 12–6, San Diego. pp. 913–7.

Tang, Z., and J. J. Garcia-Luna-Aceves. 1999a. Hop-reservation multiple access (HRMA) for ad hoc networks. In *Proceedings of the Conference on Computer Communications* (IEEE INFOCOM 1999), March, New York. pp. 194–201.

———. 1999b. A protocol for topology-dependent transmission scheduling in wireless networks. In *Proceedings of the Wireless Communications and Networking Conference* (IEEE WCNC 1999), Sept. 21–4, New Orleans. pp. 1333–1337.

Tobagi, F. A., and L. Kleinrock. 1975. Packet switching in radio channels: Part II—The hidden terminal problem in carrier sense multiple access and busy tone solution. *IEEE Transactions on Communications*, pp. 1417–33.

Toh, C.-K. 2002. *Ad hoc mobile wireless networks: Protocols and systems*. Upper Saddle River, NJ: Prentice-Hall PTR.

———, V. Vassiliou, G. Guichal, and C. H. Shih. 2000. MARCH: A medium access control protocol or multi-hop wireless ad hoc networks. In *Proceedings of the Twenty-First Century Military Communications Conference* (IEEE MILCOM 2000), October, Los Angeles. Vol. 1: 512–6.

Trigoni, N., Y. Yao, A. Demers, J. Gehrke, and R. Rajaraman. 2004. WaveScheduling: Energy-efficient data dissemination for sensor networks. In *Proceedings of the First Workshop on Data Management for Sensor Networks* (DMSN'04), Aug. 30, Toronto. pp. 48–57.

Van Dam, T., and K. Langendoen. 2003. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Conference on Embedded Networked Sensor Systems* (ACM SenSys'03), Oct. 15, Los Angeles. pp. 171–80.

Woo, A., and D. Culler. 2001. A transmission control scheme for media access in sensor networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking* (MobiCOM'01), July 16–21, Rome. pp. 201–35.

Wu, C., and V. O. K. Li. 1987. Receiver initiated busy tone multiple access in packet radio networks. In *Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology* (ACM SIGCOMM'87 Workshop), Aug. 11–13, Stowe, VT, USA. Vol. 17(5): 336–42.

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

Xu, Y., J. Heidemann, and D. Estrin. 2001. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking* (MobiCOM'01), July 16–21, Rome. pp. 70–84.