**Unit – Iii**

**MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS**

1. Introduction
2. Issues in Designing a Routing Protocol for Ad Hoc Wireless Networks
3. Classification of Routing Protocols
4. Table – Driven Routing Protocols
5. On – Demand Routing Protocols
6. Hybrid Routing Protocols
7. Routing Protocols with Efficient Flooding Mechanisms
8. Hierarchical Routing Protocols
9. Power – Aware Routing Protocols
10. Proactive Routing.

**MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS**

**4.1 INTRODUCTION**

Nodes in an ad hoc wireless network share a common broadcast radio channel. Since the radio spectrum is limited, the bandwidth available for communication in such networks is also limited. Access to this shared medium should be controlled in such a manner that all nodes receive a fair share of the available bandwidth, and that the bandwidth is utilized efficiently. Since the characteristics of the wireless medium are completely different from those of the wired medium, and since ad hoc wireless networks need to address unique issues (such as node mobility, limited bandwidth availability, error-prone broadcast channel, hidden and exposed terminal problems, and power constraints) that are not applicable to wired networks, a different set of protocols is required for controlling access to the shared medium in such networks. This chapter focuses on media access protocols for ad hoc wireless networks. First, the issues involved in designing a medium access control (MAC) protocol for ad hoc wireless networks are presented, followed by several classifications of the currently existing MAC protocols. This chapter then provides detailed descriptions of several existing MAC protocols.

**4.2 ISSUES IN DESIGNING A MAC PROTOCOL FOR AD HOC WIRELESS NETWORKS**

The following are the main issues that need to be addressed while designing a MAC protocol for ad hoc wireless networks.

**4.2.1 Bandwidth Efficiency**

As mentioned earlier, since the radio spectrum is limited, the bandwidth available for communication is also very limited. The MAC protocol must be designed in such a way that the scarce bandwidth is utilized in an efficient manner. The control overhead involved must be kept as minimal as possible. Bandwidth efficiency can be defined as the ratio of the bandwidth used for actual data transmission to the total available bandwidth. The MAC protocol must try to maximize this bandwidth efficiency.

**4.2.2 Quality of Service Support**

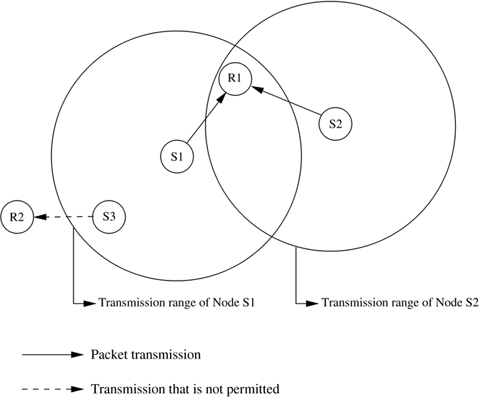
Due to the inherent nature of the ad hoc wireless network, where nodes are usually mobile most of the time, providing quality of service (QoS) support to data sessions in such networks is very difficult. Bandwidth reservation made at one point of time may become invalid once the node moves out of the region where the reservation was made. QoS support is essential for supporting time-critical traffic sessions such as in military communications. The MAC protocol for ad hoc wireless networks that are to be used in such real-time applications must have some kind of a resource reservation mechanism that takes into consideration the nature of the wireless channel and the mobility of nodes.

**4.2.3 Synchronization**

The MAC protocol must take into consideration the synchronization between nodes in the network. Synchronization is very important for bandwidth (time slot) reservations by nodes. Exchange of control packets may be required for achieving time synchronization among nodes. The control packets must not consume too much of network bandwidth.

**4.2.4 Hidden and Exposed Terminal Problems**

The hidden and exposed terminal problems are unique to wireless networks. The hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver. Collision occurs when both nodes transmit packets at the same time without knowing about the transmission of each other. For example, consider [Figure 4.1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig01). Here, if both node S1 and node S2 transmit to node R1 at the same time, their packets collide at node R1. This is because both nodes S1 and S2 are hidden from each other as they are not within the direct transmission range of each other and hence do not know about the presence of each other.



***Figure 4.1. Hidden and exposed terminal problems.***

The exposed terminal problem refers to the inability of a node, which is blocked due to transmission by a nearby transmitting node, to transmit to another node. Consider the example in [Figure 4.1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig01). Here, if a transmission from node S1 to another node R1 is already in progress, node S3 cannot transmit to node R2, as it concludes that its neighbor node S1 is in transmitting mode and hence it should not interfere with the on-going transmission.

The hidden and exposed terminal problems significantly reduce the throughput of a network when the traffic load is high. It is therefore desirable that the MAC protocol be free from the hidden and exposed terminal problems.

**4.2.5 Error-Prone Shared Broadcast Channel**

Another important factor in the design of a MAC protocol is the broadcast nature of the radio channel, that is, transmissions made by a node are received by all nodes within its direct transmission range. When a node is receiving data, no other node in its neighborhood, apart from the sender, should transmit. A node should get access to the shared medium only when its transmissions do not affect any ongoing session. Since multiple nodes may contend for the channel simultaneously, the possibility of packet collisions is quite high in wireless networks. A MAC protocol should grant channel access to nodes in such a manner that collisions are minimized. Also, the protocol should ensure that all nodes are treated fairly with respect to bandwidth allocation.

**4.2.6 Distributed Nature/Lack of Central Coordination**

Ad hoc wireless networks do not have centralized coordinators. In cellular networks, for example, the base stations act as central coordinating nodes and allocate bandwidth to the mobile terminals. But this is not possible in an ad hoc network, where nodes keep moving continuously. Therefore, nodes must be scheduled in a distributed fashion for gaining access to the channel. This may require exchange of control information. The MAC protocol must make sure that the additional overhead, in terms of bandwidth consumption, incurred due to this control information exchange is not very high.

**4.2.7 Mobility of Nodes**

This is a very important factor affecting the performance (throughput) of the protocol. Nodes in an ad hoc wireless network are mobile most of the time. The bandwidth reservations made or the control information exchanged may end up being of no use if the node mobility is very high. The MAC protocol obviously has no role to play in influencing the mobility of the nodes. The protocol design must take this mobility factor into consideration so that the performance of the system is not significantly affected due to node mobility.

**4.3 DESIGN GOALS OF A** MAC **PROTOCOL FOR AD HOC WIRELESS NETWORKS**

The following are the important goals to be met while designing a medium access control (MAC) protocol for ad hoc wireless networks:

• The operation of the protocol should be distributed.

• The protocol should provide QoS support for real-time traffic.

• The access delay, which refers to the average delay experienced by any packet to get transmitted, must be kept low.

• The available bandwidth must be utilized efficiently.

• The protocol should ensure fair allocation (either equal allocation or weighted allocation) of bandwidth to nodes.

• Control overhead must be kept as low as possible.

• The protocol should minimize the effects of hidden and exposed terminal problems.

• The protocol must be scalable to large networks.

• It should have power control mechanisms in order to efficiently manage energy consumption of the nodes.

• The protocol should have mechanisms for adaptive data rate control (adaptive rate control refers to the ability to control the rate of outgoing traffic from a node after taking into consideration such factors as load in the network and the status of neighbor nodes).

• It should try to use directional antennas which can provide advantages such as reduced interference, increased spectrum reuse, and reduced power consumption.

• Since synchronization among nodes is very important for bandwidth reservations, the protocol should provide time synchronization among nodes.

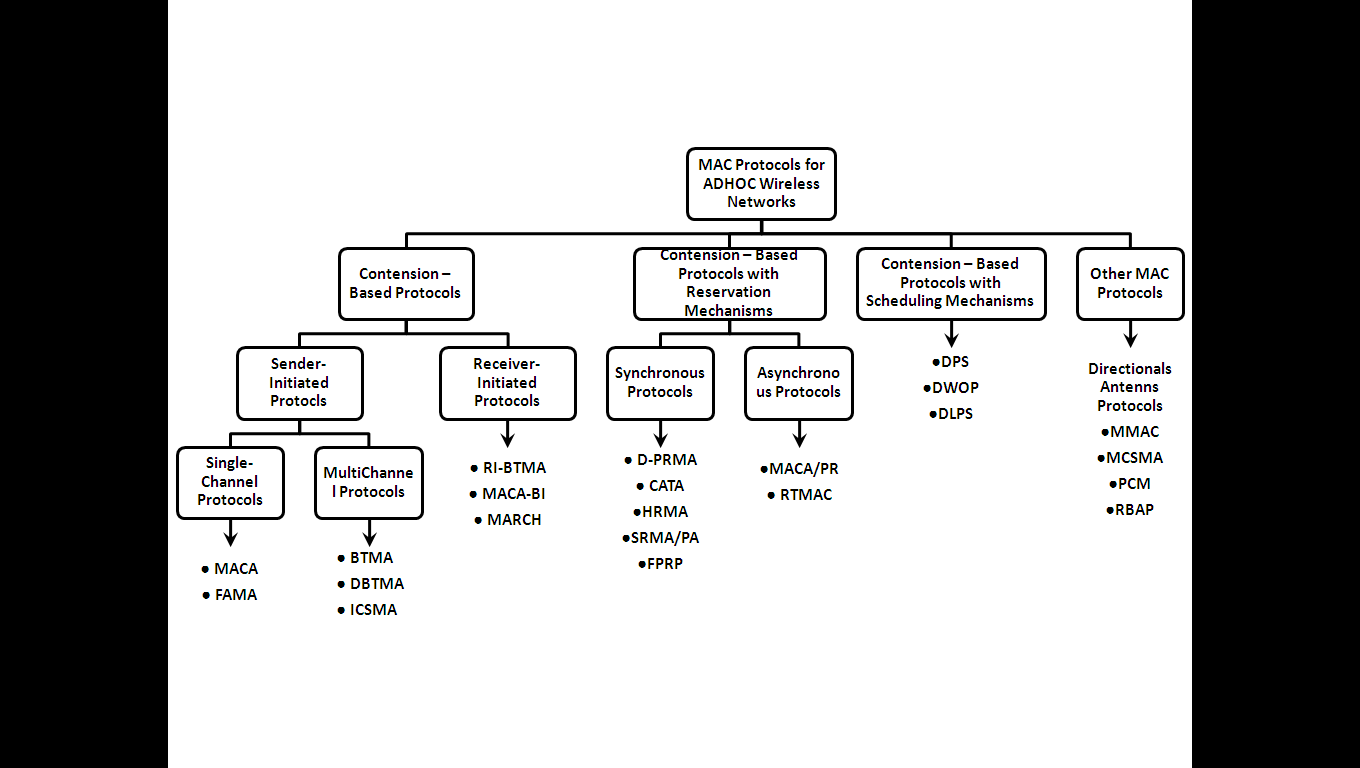
**4.4 CLASSIFICATIONS OF** MAC **PROTOCOLS**

MAC protocols for ad hoc wireless networks can be classified into several categories based on various criteria such as initiation approach, time synchronization, and reservation approaches. [Figure 4.2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig02) provides a detailed classification tree. In this section, some of the classifications of MAC protocols are briefly discussed. Ad hoc network MAC protocols can be classified into three basic types:

• Contention-based protocols

• Contention-based protocols with reservation mechanisms

• Contention-based protocols with scheduling mechanisms

******

***Figure 4.2. Classifications of MAC protocols.***

Apart from these three major types, there exist other MAC protocols that cannot be classified clearly under any one of the above three types of protocols.

**4.4.1 Contention-Based Protocols**

These protocols follow a contention-based channel access policy. A node does not make any resource reservation a priori. Whenever it receives a packet to be transmitted, it contends with its neighbor nodes for access to the shared channel. Contention-based protocols cannot provide QoS guarantees to sessions since nodes are not guaranteed regular access to the channel. Random access protocols can be further divided into two types:

• Sender-initiated protocols: Packet transmissions are initiated by the sender node.

• Receiver-initiated protocols: The receiver node initiates the contention resolution protocol.

Sender-initiated protocols can be further divided into two types:

• Single-channel sender-initiated protocols: In these protocols, the total available bandwidth is used as it is, without being divided. A node that wins the contention to the channel can make use of the entire bandwidth.

• Multichannel sender-initiated protocols: In multichannel protocols, the available bandwidth is divided into multiple channels. This enables several nodes to simultaneously transmit data, each using a separate channel. Some protocols dedicate a frequency channel exclusively for transmitting control information.

Several contention-based MAC protocols are discussed in detail in [Section 4.5](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06lev1sec5).

**4.4.2 Contention-Based Protocols with Reservation Mechanisms**

Ad hoc wireless networks sometimes may need to support real-time traffic, which requires QoS guarantees to be provided. In contention-based protocols, nodes are not guaranteed periodic access to the channel. Hence they cannot support real-time traffic. In order to support such traffic, certain protocols have mechanisms for reserving bandwidth a priori. Such protocols can provide QoS support to time-sensitive traffic sessions. These protocols can be further classified into two types:

• Synchronous protocols: Synchronous protocols require time synchronization among all nodes in the network, so that reservations made by a node are known to other nodes in its neighborhood. Global time synchronization is generally difficult to achieve.

• Asynchronous protocols: They do not require any global synchronization among nodes in the network. These protocols usually use relative time information for effecting reservations.

Various reservation-based random access protocols are discussed in detail in [Section 4.6](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06lev1sec6).

**4.4.3 Contention-Based Protocols with Scheduling Mechanisms**

As mentioned earlier, these protocols focus on packet scheduling at nodes, and also scheduling nodes for access to the channel. Node scheduling is done in a manner so that all nodes are treated fairly and no node is starved of bandwidth. Scheduling-based schemes are also used for enforcing priorities among flows whose packets are queued at nodes. Some scheduling schemes also take into consideration battery characteristics, such as remaining battery power, while scheduling nodes for access to the channel. Some of the existing scheduling-based protocols are discussed in [Section 4.7](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06lev1sec7).

**4.4.4 Other Protocols**

There are several other MAC protocols that do not strictly fall under the above categories. Some of these protocols are discussed in [Section 4.9](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06lev1sec9).

**4.5 CONTENTION-BASED PROTOCOLS**

As mentioned earlier, contention-based protocols do not have any bandwidth reservation mechanisms. All ready nodes contend for the channel simultaneously, and the winning node gains access to the channel. Since nodes are not guaranteed bandwidth, these protocols cannot be used for transmitting real-time traffic, which requires QoS guarantees from the system. In this section, several contention-based MAC protocols are described in detail.

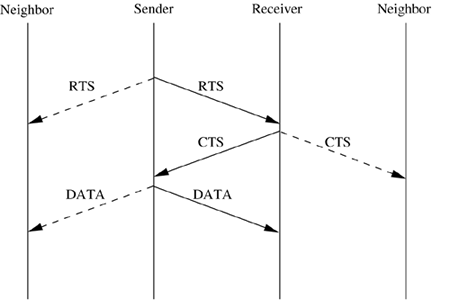
**4.5.1 MACAW: A Media Access Protocol for Wireless LANs**

This protocol is based on the multiple access collision avoidance protocol (MACA) proposed by Karn [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)]. MACA was proposed due to the shortcomings of CSMA protocols when used for wireless networks. In what follows, a brief description on why CSMA protocols fail in wireless networks is given. This is followed by detailed descriptions of the MACA protocol and the MACAW protocol.

**MACA Protocol**

The MACA protocol was proposed as an alternative to the traditional carrier sense multiple access (CSMA) protocols used in wired networks. In CSMA protocols, the sender first senses the channel for the carrier signal. If the carrier is present, it retries after a random period of time. Otherwise, it transmits the packet. CSMA senses the state of the channel only at the transmitter. This protocol does not overcome the hidden terminal problem. In a typical ad hoc wireless network, the transmitter and receiver may not be near each other at all times. In such situations, the packets transmitted by a node are prone to collisions at the receiver due to simultaneous transmissions by the hidden terminals. Also, the bandwidth utilization in CSMA protocols is less because of the exposed terminal problem.

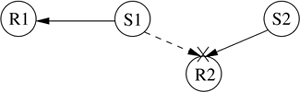
MACA does not make use of carrier-sensing for channel access. It uses two additional signaling packets: the request-to-send (RTS) packet and the clear-to-send (CTS) packet. When a node wants to transmit a data packet, it first transmits an RTS packet. The receiver node, on receiving the RTS packet, if it is ready to receive the data packet, transmits a CTS packet. Once the sender receives the CTS packet without any error, it starts transmitting the data packet. This data transmission mechanism is depicted in [Figure 4.3](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig03). If a packet transmitted by a node is lost, the node uses the binary exponential back-off (BEB) algorithm to back off for a random interval of time before retrying. In the binary exponential backoff mechanism, each time a collision is detected, the node doubles its maximum back-off window. Neighbor nodes near the sender that hear the RTS packet do not transmit for a long enough period of time so that the sender could receive the CTS packet. Both the RTS and the CTS packets carry the expected duration of the data packet transmission. A node near the receiver, upon hearing the CTS packet, defers its transmission till the receiver receives the data packet. Thus, MACA overcomes the hidden node problem. Similarly, a node receiving an RTS defers only for a short period of time till the sender could receive the CTS. If no CTS is heard by the node during its waiting period, it is free to transmit packets once the waiting interval is over. Thus, a node that hears only the RTS packet is free to transmit simultaneously when the sender of the RTS is transmitting data packets. Hence, the exposed terminal problem is also overcome in MACA. But MACA still has certain problems, which was why MACAW, described below, was proposed.



***Figure 4.3. Packet transmission in MACA.***

**MACAW Protocol**

The binary exponential back-off mechanism used in MACA at times starves flows. For example, consider [Figure 4.4](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig04). Here both nodes S1 and S2 keep generating a high volume of traffic. The node that first captures the channel (say, node S1) starts transmitting packets. The packets transmitted by the other node S2 get collided, and the node keeps incrementing its back-off window according to the BEB algorithm. As a result, the probability of node S2 acquiring the channel keeps decreasing, and over a period of time it gets completely blocked. To overcome this problem, the back-off algorithm has been modified in MACAW [[2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry02)]. The packet header now has an additional field carrying the current back-off counter value of the transmitting node. A node receiving the packet copies this value into its own back-off counter. This mechanism allocates bandwidth in a fair manner. Another problem with BEB algorithm is that it adjusts the back-off counter value very rapidly, both when a node successfully transmits a packet and when a collision is detected by the node. The back-off counter is reset to the minimum value after every successful transmission. In the modified back-off process, this would require a period of contention to be repeated after each successful transmission in order to build up the back-off timer values. To prevent such large variations in the back-off values, a multiplicative increase and linear decrease (MILD) back-off mechanism is used in MACAW. Here, upon a collision, the back-off is increased by a multiplicative factor (1.5), and upon a successful transmission, it is decremented by one. This eliminates contention and hence long contention periods after every successful transmission, at the same time providing a reasonably quick escalation in the back-off values when the contention is high.



***Figure 4.4. Example topology.***

In MACAW another modification related to the back-off mechanism has been made. MACAW implements per flow fairness as opposed to the per node fairness in MACA. This is done by maintaining multiple queues at every node, one each for each data stream, and running the back-off algorithm independently for each queue. A node that is ready to transmit packets first determines how long it needs to wait before it could transmit an RTS packet to each of the destination nodes corresponding to the top-most packets in the node's queues. It then selects the packet for which the waiting time is minimal.

In addition to the RTS and CTS control packets used in MACA, MACAW uses another new control packet called acknowledgment (ACK) packet. The need for using this additional packet arises because of the following reason. In MACA, the responsibility of recovering from transmission errors lies with the transport layer. As many TCP implementations have a minimum timeout period of about 0.5 sec, significant delay is involved while recovering from errors. But in MACAW, the error recovery responsibility is given to the data link layer (DLL). In DLL, the recovery process can be made quicker as the timeout periods can be modified in order to suit the physical media being employed. In MACAW, after successful reception of each data packet, the receiver node transmits an ACK packet. If the sender does not receive the ACK packet, it reschedules the same data packet for transmission. The back-off counter is incremented if the ACK packet is not received by the sender. If the ACK packet got lost in transmission, the sender would retry by transmitting an RTS for the same packet. But now the receiver, instead of sending back a CTS, sends an ACK for the packet received, and the sender moves on to transmit the next data packet.

In MACA, an exposed node (which received only the RTS and not the CTS packet) is free to transmit simultaneously when the source node is transmitting packets. For example, in [Figure 4.5](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig05), when a transmission is going on between nodes S1 and R1, node S2 is free to transmit. RTS transmissions by node S2 are of no use, as it can proceed further only if it can receive a CTS from R2. But this is not possible as CTS packets from R2 get collided at node S2 with packets transmitted by node S1. As a result, the back-off counter at node S2 builds up unnecessarily. So an exposed node should not be allowed to transmit. But an exposed node, since it can hear only the RTS sent by the source node and not the CTS sent by the receiver, does not know for sure whether the RTS-CTS exchange was successful. To overcome this problem, MACAW uses another small (30-bytes) control packet called the data-sending (DS) packet. Before transmitting the actual data packet, the source node transmits this DS packet. The DS packet carries information such as the duration of the data packet transmission, which could be used by the exposed nodes for updating information they hold regarding the duration of the data packet transmission. An exposed node, overhearing the DS packet, understands that the previous RTS-CTS exchange was successful, and so defers its transmissions until the expected duration of the DATA-ACK exchange. If the DS packet was not used, the exposed node (node S2) would retransmit after waiting for random intervals of time, and with a high probability the data transmission (between nodes S1 and R1) would be still going on when the exposed node retransmits. This would result in a collision and the back-off period being further incremented, which affects the node even more.

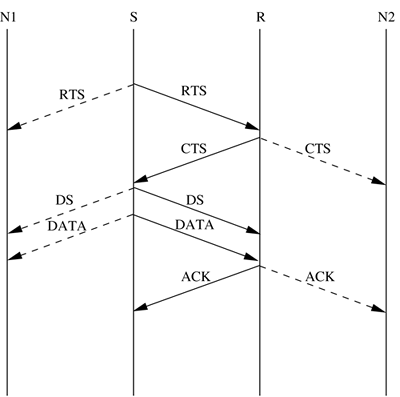
C:\Users\Pavan\Desktop\WSN\wsnmaterials\wsn\Chapter 6. MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS - Ad Hoc Wireless Networks Architectures and Protocols_files\06fig05.gif

***Figure 4.5. Example topology.***

The MACAW protocol uses one more control packet called the request-for-request-to-send (RRTS) packet. The following example shows how this RRTS packet proves to be useful. Consider [Figure 4.6](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig06). Here assume transmission is going on between nodes S1 and R1. Now node S2 wants to transmit to node R2. But since R2 is a neighbor of R1, it receives CTS packets from node R1, and therefore it defers its own transmissions. Node S2 has no way to learn about the contention periods during which it can contend for the channel, and so it keeps on trying, incrementing its back-off counter after each failed attempt. Hence the main reason for this problem is the lack of synchronization information at source S2. MACAW overcomes this problem by using the RRTS packet. In the same example shown in [Figure 4.6](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig06), receiver node R2 contends for the channel on behalf of source S2. If node R2 had received an RTS previously for which it was not able to respond immediately because of the on-going transmission between nodes S1 and R1, then node R2 waits for the next contention period and transmits the RRTS packet. Neighbor nodes that hear the RRTS packet (including node R1) are made to wait for two successive slots (for the RTS-CTS exchange to take place). The source node S2, on receiving the RRTS from node R2, transmits the regular RTS packet to node R2, and the normal packet exchange (RTS-CTS-Data-ACK) continues from here. [Figure 4.7](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig07) shows the operation of the MACAW protocol. In the figure, S is the source node and R denotes the receiver node. N1 and N2 are neighbor nodes. When RTS transmitted by node S is overheard by node N1, it refrains from transmitting until node S receives the CTS. Similarly, when the CTS transmitted by node R is heard by neighbor node N2, it defers its transmissions until the data packet is received by receiver R. On receiving this CTS packet, node S immediately transmits the DS message carrying the expected duration of the data packet transmission. On hearing this packet, node N1 backs off until the data packet is transmitted. Finally, after receiving the data packet, node R acknowledges the reception by sending node S an ACK packet.

C:\Users\Pavan\Desktop\WSN\wsnmaterials\wsn\Chapter 6. MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS - Ad Hoc Wireless Networks Architectures and Protocols_files\06fig06.gif

***Figure 4.6. Example topology.***



***Figure 4.7. Packet exchange in MACAW.***

To summarize, the MACAW protocol has been designed based on four main observations. The first is that the relevant congestion occurs at the receiver node and not at the sender. This realization makes CSMA protocols unsuitable for ad hoc wireless networks, and therefore the RTS-CTS-DATA exchange mechanism of MACA becomes necessary. MACAW further improves upon this scheme using the RTS-CTS-DS-DATA-ACK exchange mechanism. The second observation is that congestion is dependent on the location of the receiver. Therefore, instead of characterizing back-off by a single back-off parameter, separate back-off parameters have been introduced for each flow. The third is that learning about congestion at various nodes must be a collective enterprise. Therefore, the notion of copying back-off values from overheard packets has been introduced in MACA. And the final observation is that in order that nodes contend effectively for the channel, the synchronization information needs to be propagated to the concerned nodes at appropriate times. This is done in MACAW through the DS and RRTS packets. Because of the various changes described above, the performance of MACAW is significantly improved when compared to the MACA protocol.

**4.5.2 Floor Acquisition Multiple Access Protocols**

The floor acquisition multiple access (FAMA) protocols [[3](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry03)] are based on a channel access discipline which consists of a carrier-sensing operation and a collision-avoidance dialog between the sender and the intended receiver of a packet. Floor acquisition refers to the process of gaining control of the channel. At any given point of time, the control of the channel is assigned to only one node, and this node is guaranteed to transmit one or more data packets to different destinations without suffering from packet collisions. Carrier-sensing by the sender, followed by the RTS-CTS control packet exchange, enables the protocol to perform as efficiently as MACA [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)] in the presence of hidden terminals, and as efficiently as CSMA otherwise. FAMA requires a node that wishes to transmit packets to first acquire the floor (channel) before starting to transmit the packets. The floor is acquired by means of exchanging control packets. Though the control packets themselves may collide with other control packets, it is ensured that data packets sent by the node that has acquired the channel are always transmitted without any collisions. Any single-channel MAC protocol that does not require a transmitting node to sense the channel can be adapted for performing floor acquisition tasks. Floor acquisition using the RTS-CTS exchange is advantageous as the mechanism also tries to provide a solution for the hidden terminal problem.

Two FAMA protocol variants are discussed in this section: RTS-CTS exchange with no carrier-sensing, and RTS-CTS exchange with non-persistent carrier-sensing. The first variant uses the ALOHA protocol for transmitting RTS packets, while the second variant uses non-persistent CSMA for the same purpose.

**Multiple Access Collision Avoidance**

Multiple access collision avoidance (MACA) [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)], which was discussed earlier in this chapter, belongs to the category of FAMA protocols. In MACA, a ready node transmits an RTS packet. A neighbor node receiving the RTS defers its transmissions for the period specified in the RTS. On receiving the RTS, the receiver node responds by sending back a CTS packet, and waits for a long enough period of time in order to receive a data packet. Neighbor nodes of the receiver which hear this CTS packet defer their transmissions for the time duration of the impending data transfer. In MACA, nodes do not sense the channel. A node defers its transmissions only if it receives an RTS or CTS packet. In MACA, data packets are prone to collisions with RTS packets.

According to the FAMA principle, in order for data transmissions to be collision-free, the duration of an RTS must be at least twice the maximum channel propagation delay. Transmission of bursts of packets is not possible in MACA. In FAMA-NTR (discussed below) the MACA protocol is modified to permit transmission of packet bursts by enforcing waiting periods on nodes, which are proportional to the channel propagation time.

**FAMA – Non-Persistent Transmit Request**

This variant of FAMA, called FAMA – non-persistent transmit request (FAMA-NTR), combines non-persistent carrier-sensing along with the RTS-CTS control packet exchange mechanism. Before sending a packet, the sender node senses the channel. If the channel is found to be busy, then the node backs off for a random time period and retries later. If the channel is found to be free, it transmits the RTS packet. After transmitting the RTS, the sender listens to the channel for one round-trip time in addition to the time required by the receiver node to transmit a CTS. If it does not receive the CTS within this time period or if the CTS received is found to be corrupted, then the node takes a random back-off and retries later. Once the sender node receives the CTS packet without any error, it can start transmitting its data packet burst. The burst is limited to a maximum number of data packets, after which the node releases the channel, and contends with other nodes to again acquire the channel.

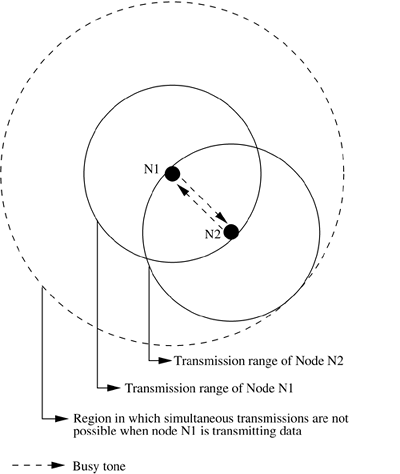
In order to allow the sender node to send a burst of packets once it acquires the floor, the receiver node is made to wait for a time duration of τ seconds after processing each data packet received. Here, τ denotes the maximum channel propagation time. A waiting period of 2τ seconds is enforced on a transmitting node after transmitting any control packet. This is done to allow the RTS-CTS exchange to take place without any error. A node transmitting an RTS is required to wait for 2τ seconds after transmitting the RTS in order to enable the receiver node to receive the RTS and transmit the corresponding CTS packet. After sending the final data packet, a sender node is made to wait for τseconds in order to allow the destination to receive the data packet and to account for the enforced waiting time at the destination node.

**4.5.3 Busy Tone Multiple Access Protocols**

**Busy Tone Multiple Access**

The busy tone multiple access (BTMA) protocol [[4](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry04)] is one of the earliest protocols proposed for overcoming the hidden terminal problem faced in wireless environments. The transmission channel is split into two: a data channel and a control channel. The data channel is used for data packet transmissions, while the control channel is used to transmit the busy tone signal.

When a node is ready for transmission, it senses the channel to check whether the busy tone is active. If not, it turns on the busy tone signal and starts data transmission; otherwise, it reschedules the packet for transmission after some random rescheduling delay. Any other node which senses the carrier on the incoming data channel also transmits the busy tone signal on the control channel. Thus, when a node is transmitting, no other node in the two-hop neighborhood of the transmitting node is permitted to simultaneously transmit. Though the probability of collisions is very low in BTMA, the bandwidth utilization is very poor. [Figure 4.8](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig08) shows the worst-case scenario where the node density is very high; the dotted circle shows the region in which nodes are blocked from simultaneously transmitting when node N1 is transmitting packets.



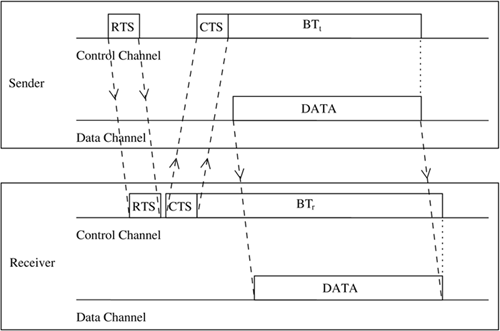
***Figure 4.8. Transmission in BTMA.***

**Dual Busy Tone Multiple Access Protocol**

The dual busy tone multiple access protocol (DBTMA) [[5](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry05)] is an extension of the BTMA scheme. Here again, the transmission channel is divided into two: the data channel and the control channel.

As in BTMA, the data channel is used for data packet transmissions. The control channel is used for control packet transmissions (RTS and CTS packets) and also for transmitting the busy tones. DBTMAuses two busy tones on the control channel, BTt and BTr. The BTt tone is used by the node to indicate that it is transmitting on the data channel. The BTr tone is turned on by a node when it is receiving data on the data channel. The two busy tone signals are two sine waves at different well-separated frequencies.

When a node is ready to transmit a data packet, it first senses the channel to determine whether the BTr signal is active. An active BTr signal indicates that a node in the neighborhood of the ready node is currently receiving packets. If the ready node finds that there is no BTr signal, it transmits the RTS packet on the control channel. On receiving the RTS packets, the node to which the RTS was destined checks whether the BTt tone is active in its neighborhood. An active BTt implies that some other node in its neighborhood is transmitting packets and so it cannot receive packets for the moment. If the node finds no BTt signal, it responds by sending a CTSpacket and then turns on the BTr signal (which informs other nodes in its neighborhood that it is receiving). The sender node, on receiving this CTS packet, turns on the BTt signal (to inform nodes in its neighborhood that it is transmitting) and starts transmitting data packets. After completing transmission, the sender node turns off the BTt signal. The receiver node, after receiving all data packets, turns off the BTr signal. The above process is depicted in [Figure 4.9](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig09).



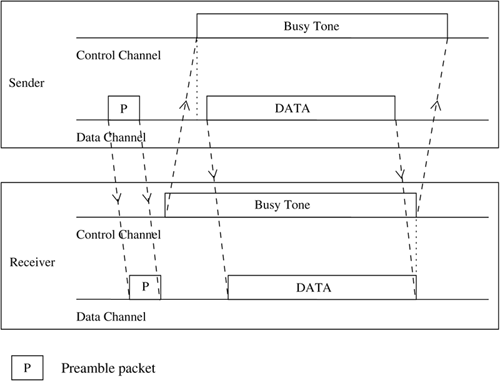
***Figure 4.9. Packet transmission in DBTMA.***

When compared to other RTS/CTS-based medium access control schemes (such as MACA [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)] and MACAW [[2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry02)]), DBTMA exhibits better network utilization. This is because the other schemes block both the forward and reverse transmissions on the data channel when they reserve the channel through their RTS or CTS packets. But in DBTMA, when a node is transmitting or receiving, only the reverse (receive) or forward (transmit) channels, respectively, are blocked. Hence the bandwidth utilization of DBTMA is nearly twice that of other RTS/CTS-based schemes.

**Receiver-Initiated Busy Tone Multiple Access Protocol**

In the receiver-initiated busy tone multiple access protocol (RI-BTMA) [[6](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry06)], similar to BTMA, the available bandwidth is divided into two channels: a data channel for transmitting data packets and a control channel. The control channel is used by a node to transmit the busy tone signal. A node can transmit on the data channel only if it finds the busy tone to be absent on the control channel.

The data packet is divided into two portions: a preamble and the actual data packet. The preamble carries the identification of the intended destination node. Both the data channel and the control channel are slotted, with each slot equal to the length of the preamble. Data transmission consists of two steps. First, the preamble needs to be transmitted by the sender. Once the receiver node acknowledges the reception of this preamble by transmitting the busy tone signal on the control channel, the actual data packet is transmitted. A sender node that needs to transmit a data packet first waits for a free slot, that is, a slot in which the busy tone signal is absent on the control channel. Once it finds such a slot, it transmits the preamble packet on the data channel. If the destination node receives this preamble packet correctly without any error, it transmits the busy tone on the control channel. It continues transmitting the busy tone signal as long as it is receiving data from the sender. If preamble transmission fails, the receiver does not acknowledge with the busy tone, and the sender node waits for the next free slot and tries again. The operation of the RI-BTMA protocol is shown in [Figure 4.10](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig10). The busy tone serves two purposes. First, it acknowledges the sender about the successful reception of the preamble. Second, it informs the nearby hidden nodes about the impending transmission so that they do not transmit at the same time.



***Figure 4.10. Packet transmission in RI-BTMA.***

There are two types of RI-BTMA protocols: the basic protocol and the controlled protocol. The basic packet transmission mechanism is the same in both protocols. In the basic protocol, nodes do not have backlog buffers to store data packets. Hence packets that suffer collisions cannot be retransmitted. Also, when the network load increases, packets cannot be queued at the nodes. This protocol would work only when the network load is not high; when network load starts increasing, the protocol becomes unstable.

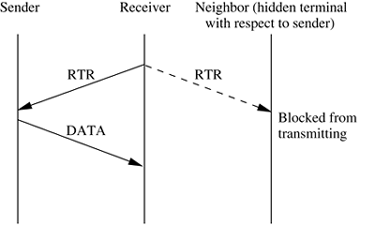
The controlled protocol overcomes this problem. This protocol is the same as the basic protocol, the only difference being the availability of backlog buffers at nodes. Therefore, packets that suffer collisions, and those that are generated during busy slots, can be queued at nodes. A node is said to be in the backlogged mode if its backlog buffer is non-empty. When a node in the backlogged mode receives a packet from its higher layers, the packet is put into the buffer and transmitted later. Suppose the packet arrives at a node when it is not in the backlogged mode, then if the current slot is free, the preamble for the packet is transmitted with probability p in the current slot itself (not transmitted in the same slot with probability (1 - p)). If the packet was received during a busy slot, the packet is just put into the backlog buffer, where it waits until the next free slot. A backlogged node transmits a backlogged packet in the next idle slot with a probability q. All other packets in the backlog buffer just keep waiting until this transmission succeeds.

This protocol can work for multi-hop radio networks as well as for single-hop fully connected networks.

**4.5.4 MACA-By Invitation**

MACA-by invitation (MACA-BI) [[7](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry07)] is a receiver-initiated MAC protocol. It reduces the number of control packets used in the MACA [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)] protocol. MACA, which is a sender-initiated protocol, uses the three-way handshake mechanism (which was shown in [Figure 4.3](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig03)), where first the RTS and CTS control packets are exchanged, followed by the actual DATA packet transmission. MACA-BI eliminates the need for the RTS packet.

In MACA-BI the receiver node initiates data transmission by transmitting a ready to receive (RTR) control packet to the sender ([Figure 4.11](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig11)). If it is ready to transmit, the sender node responds by sending a DATA packet. Thus data transmission in MACA-BI occurs through a two-way handshake mechanism.

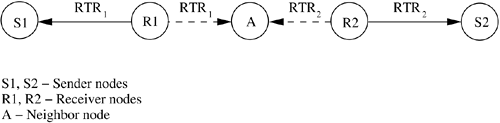


***Figure 4.11. Packet transmission in MACA-BI.***

The receiver node may not have an exact knowledge about the traffic arrival rates at its neighboring sender nodes. It needs to estimate the average arrival rate of packets. For providing necessary information to the receiver node for this estimation, the DATA packets are modified to carry control information regarding the backlogged flows at the transmitter node, number of packets queued, and packet lengths. Once this information is available at the receiver node, the average rate of the flows can be easily estimated. Suppose the estimation is incorrect or is not possible (when the first data packet of the session is to be transmitted), the MACA-BI protocol can be extended by allowing the sender node to declare its backlog through an RTS control packet, if an RTR packet is not received within a given timeout period.

In MACA, the CTS packet was used to inform the hidden terminals (nodes) about the impending DATA packet transmission, so that they do not transmit at the same time and disrupt the session. This role is played in MACA-BI by the RTR packets. An RTR packet carries information about the time interval during which the DATA packet would be transmitted. When a node hears RTR packets transmitted by its neighbors, it can obtain information about the duration of DATA packet transmissions by nodes that may be either its direct one-hop neighbors or its two-hop neighbors, that is, hidden terminals. Since it has information about transmissions by the hidden terminals, it refrains from transmitting during those periods ([Figure 4.11](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig11)). Hence the hidden terminal problem is overcome in MACA-BI. Collision among DATA packets is impossible.

However, the hidden terminal problem still affects the control packet transmissions. This leads to protocol failure, as in certain cases the RTR packets can collide with DATA packets. One such scenario is depicted in [Figure 4.12](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig12). Here, RTR packets transmitted by receiver nodes R1 and R2 collide at node A. So node A is not aware of the transmissions from nodes S1 and S2. When node A transmits RTR packets, they collide with DATA packets at receiver nodes R1 and R2.



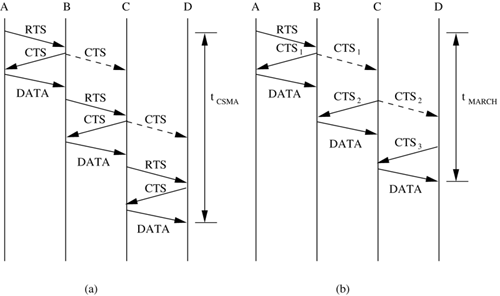
***Figure 4.12. Hidden terminal problem in MACA-BI.***

The efficiency of the MACA-BI scheme is mainly dependent on the ability of the receiver node to predict accurately the arrival rates of traffic at the sender nodes.

**4.5.5 Media Access with Reduced Handshake**

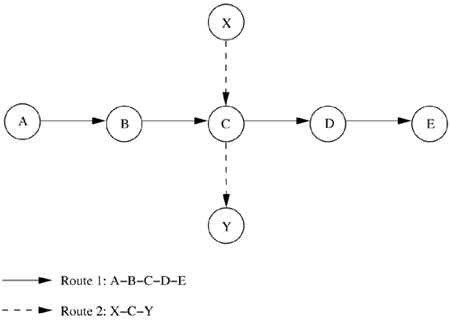
The media access with reduced handshake protocol (MARCH) [[8](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry08)] is a receiver-initiated protocol. MARCH, unlike MACA-BI [[7](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry07)], does not require any traffic prediction mechanism. The protocol exploits the broadcast nature of traffic from omnidirectional antennas to reduce the number of handshakes involved in data transmission. In MACA [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)], the RTS-CTS control packets exchange takes place before the transmission of every data packet. But in MARCH, the RTS packet is used only for the first packet of the stream. From the second packet onward, only the CTS packet is used.

A node obtains information about data packet arrivals at its neighboring nodes by overhearing the CTS packets transmitted by them. It then sends a CTS packet to the concerned neighbor node for relaying data from that node. This mechanism is illustrated in [Figure 4.13](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig13). [Figure 4.13 (a)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig13) depicts the packet exchange mechanism of MACA. Here two control packets RTS and CTS need to be exchanged before each data packet is transmitted. It can be seen from this figure that node C, for example, can hear both the CTS and the RTS packets transmitted by node B. MARCH uses this property of the broadcast channel to reduce the two-way handshake into a single CTS-only handshake. [Figure 4.13 (b)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig13) shows the handshake mechanism of MARCH. Here, when node B transmits the CTS1 packet, this packet is also heard by node C. A CTS packet carries information regarding the duration of the next data packet. Node C therefore determines the time at which the next data packet would be available at node B. It sends the CTS2 packet at that point of time. On receiving the CTS2 packet, node B sends the data packet directly to node C. It can be observed from the figure that the time taken for a packet transmitted by node A to reach node D in MARCH, that is, tMARCH, is less compared to the time taken in MACA, tMACA.



***Figure 4.13. Handshake mechanism in (a) MACA and (b) MARCH.***

The CTS packet carries the MAC addresses of the sender and the receiver node, and the route identification number (RTid) for that flow. The RTid is used by nodes in order to avoid misinterpretation of CTS packets and initiation of false CTS-only handshakes. Consider [Figure 4.14](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig14). Here there are two routes – Route 1: A-B-C-D-E and Route 2: X-C-Y. When node C hears a CTS packet transmitted by node B, by means of the RTid field on the packet, it understands that the CTS was transmitted by its upstream node (upstream node refers to the next hop neighbor node on the path from the current node to the source node of the data session) on Route 1. It invokes a timer T which is set to expire after a certain period of time, long enough for node B to receive a packet from node A. A CTS packet is transmitted by node C once the timer expires. This CTS is overheard by node Y also, but since the RTid carried on the CTS is different from the RTid corresponding to Route 2, node Y does not respond. In MARCH, the MAC layer has access to tables that maintain routing information (such as RTid), but the protocol as such does not get involved in routing.



***Figure 4.14. Example topology.***

The throughput of MARCH is significantly high when compared to MACA, while the control overhead is much less. When the network is heavily loaded, the average end-to-end delay in packet delivery for MARCH is very low compared to that of MACA. All the above advantages are mainly due to the fact that MARCH has a lower number of control packet handshakes compared to MACA. The lower number of control packets transmitted reduces the control overhead while improving the throughput, since less bandwidth is being consumed for control traffic.

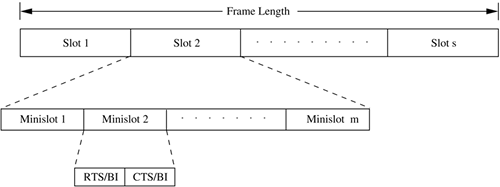
**4.6 CONTENTION-BASED PROTOCOLS WITH RESERVATION MECHANISMS**

Protocols described in this section have certain mechanisms that aid the nodes in effecting bandwidth reservations. Though these protocols are contention-based, contention occurs only during the resource (bandwidth) reservation phase. Once the bandwidth is reserved, the node gets exclusive access to the reserved bandwidth. Hence, QoS support can be provided for real-time traffic.

**4.4.1 Distributed Packet Reservation Multiple Access Protocol**

The distributed packet reservation multiple access protocol (D-PRMA) [[9](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry09)] extends the earlier centralized packet reservation multiple access (PRMA) [[10](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry10)] scheme into a distributed scheme that can be used in ad hoc wireless networks. PRMA was proposed for voice support in a wireless LAN with a base station, where the base station serves as the fixed entity for the MAC operation. D-PRMA extends this protocol for providing voice support in ad hoc wireless networks.

D-PRMA is a TDMA-based scheme. The channel is divided into fixed- and equal-sized frames along the time axis ([Figure 4.15](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig15)). Each frame is composed of s slots, and each slot consists of m minislots. Each minislot can be further divided into two control fields, RTS/BI and CTS/BI (BI stands for busy indication), as shown in the figure. These control fields are used for slot reservation and for overcoming the hidden terminal problem. Details on how this is done will be explained later in this section. All nodes having packets ready for transmission contend for the first minislot of each slot. The remaining (m - 1) minislots are granted to the node that wins the contention. Also, the same slot in each subsequent frame can be reserved for this winning terminal until it completes its packet transmission session. If no node wins the first minislot, then the remaining minislots are continuously used for contention, until a contending node wins any minislot. Within a reserved slot, communication between the source and receiver nodes takes place by means of either time division duplexing (TDD) or frequency division duplexing (FDD). Any node that wants to transmit packets has to first reserve slots, if they have not been reserved already. A certain period at the beginning of each minislot is reserved for carrier-sensing. If a sender node detects the channel to be idle at the beginning of a slot (minislot 1), it transmits an RTS packet (slot reservation request) to the intended destination through the RTS/BI part of the current minislot. On successfully receiving this RTS packet, the receiver node responds by sending a CTS packet through the CTS/BI of the same minislot. If the sender node receives this CTS successfully, then it gets the reservation for the current slot and can use the remaining minislots, that is, minislots 2 to m. Otherwise, it continues the contention process through the subsequent minislots of the same slot.



***Figure 4.15. Frame structure in D-PRMA.***

In order to prioritize nodes transmitting voice traffic (voice nodes) over nodes transmitting normal data traffic (data nodes), two rules are followed in D-PRMA. According to the first rule, the voice nodes are allowed to start contending from minislot 1 with probability p = 1; data nodes can start contending only with probability p < 1. For the remaining (m - 1) minislots, both the voice nodes and the data nodes are allowed to contend with probability p < 1. This is because the reservation process for a voice node is triggered only after the arrival of voice traffic at the node; this avoids unnecessary reservation of slots. According to the second rule, only if the node winning the minislot contention is a voice node, is it permitted to reserve the same slot in each subsequent frame until the end of the session. If a data node wins the contention, then it is allowed to use only one slot, that is, the current slot, and it has to make fresh reservations for each subsequent slot.

Nodes that are located within the radio coverage of the receiver should not be permitted to transmit simultaneously when the receiver is receiving packets. If permitted, packets transmitted by them may collide with the packets of the on-going traffic being received at the receiver. Though a node which is located outside the range of a receiver is able to hear packets transmitted by the sender, it should still be allowed to transmit simultaneously. The above requirements, in essence, mean that the protocol must be free of the hidden terminal and exposed terminal problems. In D-PRMA, when a node wins the contention in minislot 1, other terminals must be prevented from using any of the remaining (m - 1) minislots in the same slot for contention (requirement 1). Also, when a slot is reserved in subsequent frames, other nodes should be prevented from contending for those reserved slots (requirement 2). The RTS-CTS exchange mechanism taking place in the reservation process helps in trying to satisfy requirement 1. A node that wins the contention in minislot 1 starts transmitting immediately from minislot 2. Any other node that wants to transmit will find the channel to be busy from minislot 2. Since an RTS can be sent only when the channel is idle, other neighboring nodes would not contend for the channel until the on-going transmission gets completed. A node sends an RTS in the RTS/BI part of a minislot. Only a node that receives an RTS destined to it is allowed to use the CTS/BI part of the slot for transmitting the CTS. So the CTS packet does not suffer any collision due to simultaneous RTS packet transmissions. This improves the probability for a successful reservation. In order to avoid the hidden terminal problem, all nodes hearing the CTS sent by the receiver are not allowed to transmit during the remaining period of that same slot. In order to avoid the exposed terminal problem, a node hearing the RTS but not the CTS (sender node's neighbor) is still allowed to transmit. But, if the communication is duplex in nature, where a node may transmit and receive simultaneously, even such exposed nodes (that hear RTS alone) should not be allowed to transmit. Therefore, D-PRMA makes such a node defer its transmissions for the remaining time period of the same slot. If an RTS or CTS packet collides, and a successful reservation cannot be made in the first minislot, then the subsequent (m - 1) minislots of the same slot are used for contention.

For satisfying requirement 2, the following is done. The receiver of the reserved slot transmits a busy indication (BI) signal through the RTS/BI part of minislot 1 of the same slot in each of the subsequent frames, without performing a carrier-sense. The sender also performs a similar function, transmitting the BI through the CTS/BI part of minislot 1 of the same slot in each subsequent frame. When any node hears a BI signal, it does not further contend for that slot in the current frame. Because of this, the reserved slot in each subsequent frame is made free of contention. Also, making the receiver transmit the BI signal helps in eliminating the hidden terminal problem, since not all neighbors of the receiver can hear from the sender. Finally, after a node that had made the reservation completes its data transmission and does not anymore require a reserved slot, it just stops transmitting the BI signal. D-PRMA is more suited for voice traffic than for data traffic applications.

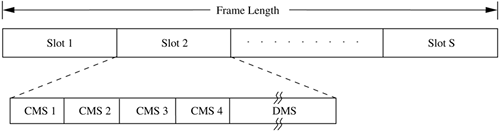
**4.4.2 Collision Avoidance Time Allocation Protocol**

The collision avoidance time allocation protocol (CATA) [[11](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry11)] is based on dynamic topology-dependent transmission scheduling. Nodes contend for and reserve time slots by means of a distributed reservation and handshake mechanism. CATA supports broadcast, unicast, and multicast transmissions simultaneously. The operation of CATA is based on two basic principles:

• The receiver(s) of a flow must inform the potential source nodes about the reserved slot on which it is currently receiving packets. Similarly, the source node must inform the potential destination node(s) about interferences in the slot.

• Usage of negative acknowledgments for reservation requests, and control packet transmissions at the beginning of each slot, for distributing slot reservation information to senders of broadcast or multicast sessions.

Time is divided into equal-sized frames, and each frame consists of S slots ([Figure 4.16](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig16)). Each slot is further divided into five minislots. The first four minislots are used for transmitting control packets and are called control minislots (CMS1, CMS2, CMS3, and CMS4). The fifth and last minislot, called data minislot (DMS), is meant for data transmission. The data minislot is much longer than the control minislots as the control packets are much smaller in size compared to data packets.



***Figure 4.16. Frame format in CATA.***

Each node that receives data during the DMS of the current slot transmits a slot reservation (SR) packet during the CMS1 of the slot. This serves to inform other neighboring potential sender nodes about the currently active reservation. The SR packet is either received without error at the neighbor nodes or causes noise at those nodes, in both cases preventing such neighbor nodes from attempting to reserve the current slot. Every node that transmits data during the DMS of the current slot transmits a request-to-send (RTS) packet during CMS2 of the slot. This RTS packet, when received by other neighbor nodes or when it collides with other RTS packets at the neighbor nodes, causes the neighbor nodes to understand that the source node is scheduled to transmit during the DMS of the current slot. Hence they defer their transmissions during the current slot.

The control minislots CMS3 and CMS4 are used as follows. The sender of an intended reservation, if it senses the channel to be idle during CMS1, transmits an RTS packet during CMS2. The receiver node of a unicast session transmits a clear-to-send (CTS) packet during CMS3. On receiving this packet, the source node understands that the reservation was successful and transmits data during the DMS of that slot, and during the same slot in subsequent frames, until the unicast flow gets terminated. Once the reservation has been made successfully in a slot, from the next slot onward, both the sender and receiver do not transmit anything during CMS3, and during CMS4 the sender node alone transmits a not-to-send (NTS) packet. The purpose of the NTS packet is explained below. If a node receives an RTS packet for broadcast or multicast during CMS2, or if it finds the channel to be free during CMS2, it remains idle and does not transmit anything during CMS3 and CMS4. Otherwise, it sends a not-to-send (NTS) packet during CMS4. The NTS packet serves as a negative acknowledgment; a potential multicast or broadcast source node that receives the NTS packet during CMS4, or that detects noise during CMS4, understands that its reservation request had failed, and it does not transmit during the DMS of the current slot. If it finds the channel to be free during CMS4, which implies that its reservation request was successful, it starts transmitting the multicast or broadcast packets during the DMS of the slot.

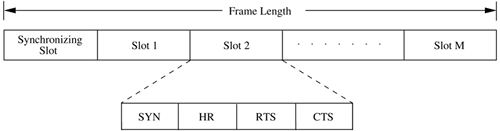
The length of the frame is very important in CATA. For any node (say, node A) to broadcast successfully, there must be no other node (say, node B) in its two-hop neighborhood that transmits simultaneously. If such a node B exists, then if node B is within node A's one-hop neighborhood, node A and node B cannot hear the packets transmitted by each other. If node B is within the two-hop neighborhood of node A, then the packets transmitted by nodes A and B would collide at their common neighbor nodes. Therefore, for any node to transmit successfully during one slot in every frame, the number of slots in each frame must be larger than the number of two-hop neighbor nodes of the transmitting node. The worst-case value of the frame length, that is, the number of slots in the frame, would be Min(d2 + 1, N), where d is the maximum degree (degree of a node refers to the count of one-hop neighbors of the node) of a node in the network, and N is the total number of nodes in the network.

CATA works well with simple single-channel half-duplex radios. It is simple and provides support for collision-free broadcast and multicast traffic.

**4.4.3 Hop Reservation Multiple Access Protocol**

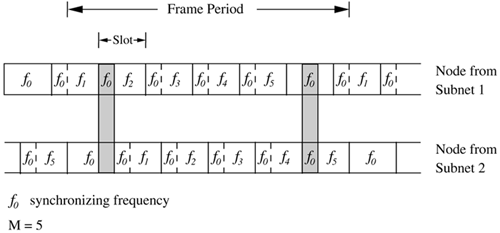
The hop reservation multiple access protocol (HRMA) [12] is a multichannel MAC protocol which is based on simple half-duplex, very slow frequency-hopping spread spectrum (FHSS) radios. It uses a reservation and handshake mechanism to enable a pair of communicating nodes to reserve a frequency hop, thereby guaranteeing collision-free data transmission even in the presence of hidden terminals. HRMA can be viewed as a time slot reservation protocol where each time slot is assigned a separate frequency channel.

Out of the available L frequency channels, HRMA uses one frequency channel, denoted by f0, as a dedicated synchronizing channel. The nodes exchange synchronization information on f0. The remaining L - 1 frequencies are divided into frequency pairs (denoted by **(f*i,* fi\* )**, i = 1, 2, 3, .., M), thereby restricting the length of the hopping sequence to M. fi is used for transmitting and receiving hop-reservation (HR) packets, request-to-send (RTS) packets, clear-to-send (CTS) packets, and data packets. **fi**\* is used for sending and receiving acknowledgment (ACK) packets for the data packets received or transmitted on frequency ***fi.*** In HRMA, time is slotted, and each slot is assigned a separate frequency hop, which is one among the M frequency hops in the hopping sequence. Each time slot is divided into four periods, namely, synchronizing period, HR period, RTS period, and CTS period, each period meant for transmitting or receiving the synchronizing packet, FR packet, RTS packet, and CTS packet, respectively. All idle nodes, that is, nodes that do not transmit or receive packets currently, hop together. During the synchronizing period of each slot, all idle nodes hop to the synchronizing frequency f0 and exchange synchronization information. During the HR, RTS, and CTS periods, they just stay idle, dwelling on the common frequency hop assigned to each slot. In addition to the synchronization period used for synchronization purposes, an exclusive synchronization slot is also defined at the beginning of each HRMA frame (Figure 4.17). This slot is of the same size as that of the other normal slots. All idle nodes dwell on the synchronizing frequency f0 during the synchronizing slot and exchange synchronization information that may be used to identify the beginning of a frequency hop in the common hopping sequence, and also the frequency to be used in the immediately following hop. Thus the HRMA frame, as depicted in Figure 4.17, is composed of the single synchronizing slot, followed by M consecutive normal slots.



***Figure 4.17. Frame format in HRMA.***

When a new node enters the network, it remains on the synchronizing frequency f0 for a long enough period of time so as to gather synchronization information such as the hopping pattern and the timing of the system. If it receives no synchronization information, it assumes that it is the only node in the network, broadcasts its own synchronization information, and forms a one-node system. Since synchronization information is exchanged during every synchronization slot, new nodes entering the system can easily join the network. If μ is the length of each slot and μs the length of the synchronization period on each slot, then the dwell time of fo at the beginning of each frame would be μ + μs. Consider the case where nodes from two different disconnected network partitions come nearby. Figure 4.18 depicts the worst-case frequency overlap scenario. In the figure, the maximum number of frequency hops M = 5. It is evident from the figure that within any time period equal to the duration of a HRMA frame, any two nodes from the two disconnected partitions always have at least two overlapping time periods of length μs on the synchronizing frequency f0. Therefore, nodes belonging to disconnected network components can easily merge into a single network.



***Figure 4.18. Merging of subnets.***

When a node receives data to be transmitted, it first listens to the HR period of the immediately following slot. If it hears an HR packet, it backs off for a randomly chosen period (which is a multiple of slot time). If it finds the channel to be free during the SR period, it transmits an RTS packet to the destination during the RTS period of the slot and waits for the CTS packet. On receiving the RTS, the destination node transmits the CTS packet during the CTS period of the same slot, stays on the same frequency currently being used, and waits for the data packet. If the source node receives the CTS packet correctly, it implies that the source and receiver nodes have successfully reserved the current hop. In case the source node does not receive any CTS packet, it backs off for a random number of time slots and repeats the entire process again. The source and receiver nodes dwell on the same reserved frequency throughout the data transmission process, which starts immediately after the CTS period. As mentioned earlier, a separate frequency (image, i = 1, 2, ..., M) is used for transmitting acknowledgments. After transmitting each data packet, the source node hops onto this acknowledgment frequency. The receiver sends an acknowledgment (ACK) packet back to the source on this acknowledgment frequency. Once the ACK packet transmission/reception is over, both the source and receiver hop back to the reserved frequency to continue with the data transmission.

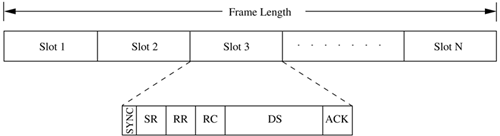
After the CTS period of a slot, the idle nodes that do not transmit or receive packets hop onto the synchronization frequency f0 and exchange synchronization information. They dwell on f0 for a time period of μs and then hop onto the next frequency hop in the common hopping sequence.

The data packets transmitted can be of any size. Data transmission can take place through a single packet or through a train of packets. A maximum dwell period has been defined in order to prevent nodes from hogging onto a particular frequency channel. Therefore, the transmission time for the data packet, or the train of data packets, should not exceed this maximum dwell time. Suppose the sender needs to transmit data packets across multiple frames, then it informs the receiver node through the header of the data packet it transmits. On reading this information, the receiver node transmits an HR packet during the HR period of the same slot in the next frame. The neighbor nodes of the receiver on hearing this HR packet refrain from using the frequency hop reserved. On receiving the HR packet, the source node of the session sends an RTS packet during the RTS period and jams other RTS packets (if any) destined to its neighbors, so that the neighbor nodes do not interfere on the reserved frequency hop. Both the sender and the receiver remain silent during the CTS period, and data transmission resumes once this CTS period gets over.

**4.4.4 Soft Reservation Multiple Access with Priority Assignment**

Soft reservation multiple access protocol with priority assignment (SRMA/PA) [13] was developed with the main objective of supporting integrated services of real-time and non-real-time applications in ad hoc wireless networks, at the same time maximizing the statistical multiplexing gain. Nodes use a collision-avoidance handshake mechanism and a soft reservation mechanism in order to contend for and effect reservation of time slots. The soft reservation mechanism allows any urgent node, transmitting packets generated by a real-time application, to take over the radio resource from another node of a non-real-time application on an on-demand basis. SRMA/PA is a TDMA-based protocol in which nodes are allocated different time slots so that the transmissions are collision-free. The main features of SRMA/PA are a unique frame structure and soft reservation capability for distributed and dynamic slot scheduling, dynamic and distributed access priority assignment and update policies, and a time-constrained back-off algorithm.

Time is divided into frames, with each frame consisting of a fixed number (N) of time slots. The frame structure is shown in Figure 4.19. Each slot is further divided into six different fields, SYNC, soft reservation (SR), reservation request (RR), reservation confirm (RC), data sending (DS), and acknowledgment (ACK). The SYNC field is used for synchronization purposes. The SR, RR, RC, and ACK fields are used for transmitting and receiving the corresponding control packets. The DS field is used for data transmission. The SR packet serves as a busy tone. It informs the nodes in the neighborhood of the transmitting node about the reservation of the slot. The SR packet also carries the access priority value assigned to the node that has reserved the slot. When an idle node receives a data packet for transmission, the node waits for a free slot and transmits the RR packet in the RR field of that slot. A node determines whether or not a slot is free through the SR field of that slot. In case of a voice terminal node, the node tries to take control of the slot already reserved by a data terminal if it finds its priority level to be higher than that of the data terminal. This process is called soft reservation. This makes the SRMA/PA different from other protocols where even if a node has lower access priority compared to other ready nodes, it proceeds to complete the transmission of the entire data burst once it has reserved the channel.



***Figure 4.19. Frame structure in SRMA/PA.***

Priority levels are initially assigned to nodes based on the service classes (real-time or non-real-time) in a static manner. Once the node acquires the channel, the corresponding slot stays reserved for the node until the node completes transmitting the entire data burst. The node is assigned a prespecified priority, image or image, respectively, for voice and data terminals. R denotes that the node is a reserved node, that is, a node that has successfully reserved the slot. It is required that image, such that delay-sensitive voice applications get preference over normal data applications. Whenever the reservation attempt fails due to collision, the access priority of the node is updated based on the urgency of its packets.

A node that is currently transmitting is said to be in the active state. A node is said to be in the idle state if it does not have any packet to be transmitted. In the active state itself, nodes can be in one of the two states: access state and reserved state. Access state is one in which the node is backlogged and is trying to reserve a slot for transmission. The node is said to be in the reserved state if it has already reserved the slot for transmission. Whenever the access priority level of a voice terminal in the access state becomes greater than that of a data terminal in the reserved state, which could be known from the SR field, the corresponding slot is taken over by the prioritized voice terminal. In order to effect this mechanism, the values of priority levels must be such that image, where image and image are the access priority values of a voice terminal and data terminal, respectively, after its nth reservation attempt results in a collision. This soft reservation feature of SRMA/PA, where a voice terminal can take over the slots reserved by a data terminal whenever, due to the urgent nature of its traffic, its access priority becomes higher than that of the data terminal, helps in maximizing the statistical multiplexing gain for voice-data integrated services.

The RR-RC-DS-ACK exchange mechanism of SRMA/PA is similar to the RTS-CTS-DATA-ACK exchange mechanism of MACAW [2]. The RR and RC packets help in eliminating the hidden terminal problem. The major difference between SRMA/PA and CATA protocol [11] is that, while in CATA the slot reservation (SR) packet is transmitted by the receiver of the session, in SRMA/PA it is sent by the source node. Also, the soft reservation feature of SRMA/PA is absent in CATA.

The access priorities are assigned to nodes and updated in a distributed and dynamic manner. This allows dynamic sharing of the shared channel. On receiving a new packet for transmission, an idle node becomes active. Now, transition to the access state is made with the initial access priority assigned a value **Pv(o)**or **Pd(o)**, depending on whether it is a voice or data terminal. If the random access attempt to effect the reservation by transmitting an RR packet ends up in a collision, then the access priorities of the node concerned are increased as follows:

C:\Documents and Settings\User\My Documents\06equ01.gif

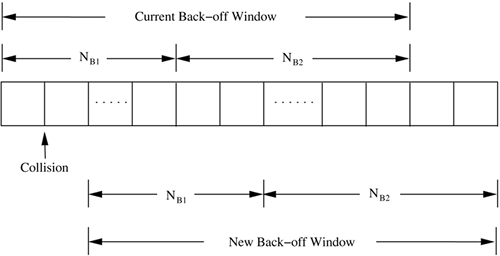
where Δpν and Δpd are the incremental access priorities for voice and data services, respectively. They reflect the urgency of the traffic queued at the two types of terminals, and are given as below:



where τS is the slot duration, τr is the residual lifetime for the voice service, lQ is the queue length, and α is a scaling coefficient. In order that the access priority of a voice terminal is always higher than that of a data terminal, the following constraint is followed:



Though dynamic assignment and update of access priority values are followed in SRMA/PA, collisions among nodes with the same priority and carrying traffic of the same service types cannot be avoided completely. Collisions occur during the RR field of the slot. In order to avoid collisions, a binary exponential back-off algorithm is used for non-real-time connections, and a modified binary exponential back-off algorithm is used for real-time connections. The modified algorithm implements a priority access policy in order to meet the delay requirements of real-time sessions. Here the back-off window is divided into two different regions, each region having a length of NB1 and NB2, respectively, for real-time and non-real-time traffic. Each node checks the laxity of its head-of-line packet (laxity is the difference between the maximum access delay allowed and the residual lifetime of the packet). If the laxity exceeds the threshold Tlimit slots, one slot out of the NB1 slots is selected randomly. Otherwise, one out of the NB2 slots is chosen randomly, that is, if the node is unable to make a reservation within the given time (Tlimit slots), it is given a higher priority compared to other non-real-time nodes by choosing a slot from NB1; otherwise, the node selects a slot from NB2. The RR packet is transmitted on this chosen slot. Here again, if more than one node selects the same random slot and their RR packets collide again, a new back-off window starts immediately after the current slot. This is shown in Figure 4.20. The above back-off mechanism, which gives high preference to nodes transmitting delay-sensitive traffic, helps in guaranteeing the QoS requirements of real-time services in the network. The parameters NB1, NB2, and Tlimit significantly affect the performance of the protocol, and must be chosen carefully based on the traffic load expected on the network.

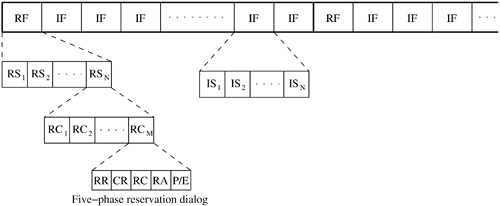


***Figure 4.20. Back-off windows.***

**4.4.5 Five-Phase Reservation Protocol**

The five-phase reservation protocol (FPRP) [[14](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry14)] is a single-channel time division multiple access (TDMA)-based broadcast scheduling protocol. Nodes use a contention mechanism in order to acquire time slots. The protocol is fully distributed, that is, multiple reservations can be simultaneously made throughout the network. No ordering among nodes is followed; nodes need not wait for making time slot reservations. The slot reservations are made using a five-phase reservation process. The reservation process is localized; it involves only the nodes located within the two-hop radius of the node concerned. Because of this, the protocol is insensitive to the network size, that is, it is scalable. FPRP also ensures that no collisions occur due to the hidden terminal problem.

Time is divided into frames. There are two types of frames: reservation frame (RF) and information frame (IF). Each RF is followed by a sequence of IFs. Each RF has N reservation slots (RS), and each IF has N information slots (IS). In order to reserve an IS, a node needs to contend during the corresponding RS. Based on these contentions, a TDMA schedule is generated in the RF and is used in the subsequent IFs until the next RF. The structure of the frames is shown in [Figure 4.21](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig21). Each RS is composed of M reservation cycles (RC). Within each RC, a five-phase dialog takes place, using which a node reserves slots. If a node wins the contention in an RC, it is said to have reserved the IS corresponding to the current RS in the subsequent IFs of the current frame. Otherwise, the node contends during the subsequent RCs of the current RS until itself or any other node (1-hop or 2-hop neighbor) succeeds. During the corresponding IS, a node would be in one of the following three states: transmit (T), receive (R), or blocked (B). The five-phase dialog ensures that the protocol is free from the hidden node problem, and also ensures that once a reservation is made by a node with a high probability, it gets sole access to the slot within its neighborhood.



***Figure 4.21. Frame structure in FPRP.***

The protocol assumes the availability of global time at all nodes. Each node therefore knows when a five-phase cycle would start. The five phases of the reservation process are as follows:

1. Reservation request phase: Nodes that need to transmit packets send reservation request (RR) packets to their destination nodes.
2. Collision report phase: If a collision is detected by any node during the reservation request phase, then that node broadcasts a collision report (CR) packet. The corresponding source nodes, upon receiving the CR packet, take necessary action.
3. Reservation confirmation phase: A source node is said to have won the contention for a slot if it does not receive any CR messages in the previous phase. In order to confirm the reservation request made in the reservation request phase, it sends a reservation confirmation (RC) message to the destination node in this phase.
4. Reservation acknowledgment phase: In this phase, the destination node acknowledges reception of the RC by sending back a reservation acknowledgment (RA) message to the source. The hidden nodes that receive this message defer their transmissions during the reserved slot.
5. Packing and elimination (P/E) phase: Two types of packets are transmitted during this phase: packing packet and elimination packet. The details regarding the use of these packets will be described later in this section.

Each of the above five phases is described below.

Reservation request phase:

In this phase, each node that needs to transmit packets sends an RR packet to the intended destination node with a contention probability p, in order to reserve an IS. Such nodes that send RR packets are called requesting nodes (RN). Other nodes just keep listening during this phase.

Collision report phase:

If any of the listening nodes detects collision of RR packets transmitted in the previous phase, it broadcasts a collision report (CR) packet. By listening for CR packets in this phase, an RN comes to know about collision of the RR packet it had sent. If no CR is heard by the RN in this phase, then it assumes that the RR packet did not collide in its neighborhood. It then becomes a transmitting node (TN). Once it becomes a transmitting node, the node proceeds to the next phase, the reservation confirmation phase. On the other hand, if it hears a CR packet in this phase, it waits until the next reservation request phase, and then tries again. Thus, if two RNs are hidden from each other, their RR packets collide, both receive CR packets, and no reservation is made, thereby eliminating the hidden terminal problem.

Reservation confirmation phase:

An RN that does not receive any CR packet in the previous phase, that is, a TN, sends an RC packet to the destination node. Each neighbor node that receives this packet understands that the slot has been reserved, and defers its transmission during the corresponding information slots in the subsequent information frames until the next reservation frame.

Reservation acknowledgment phase:

On receiving the RC packet, the intended receiver node responds by sending an RA packet back to the TN. This is used to inform the TN that the reservation has been established. In case the TN is isolated and is not connected to any other node in the network, then it would not receive the RA packet, and thus becomes aware of the fact that it is isolated. Thus the RC packet prevents such isolated nodes from transmitting further. The reservation acknowledgment phase also serves another purpose. Other two-hop neighbor nodes that receive this RA packet get blocked from transmitting. Therefore, they do not disturb the transmission that is to take place in the reserved slots.

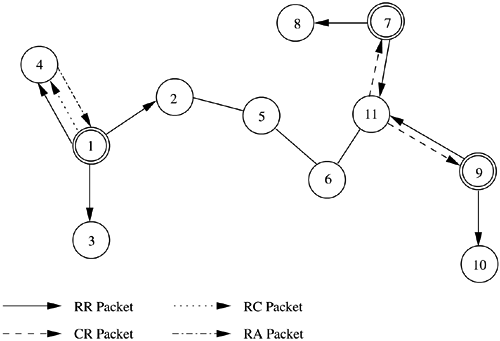
When more than two TNs are located nearby, it results in a deadlock condition. Such situations may occur when there is no common neighbor node present when the RNs transmit RR packets. Collisions are not reported in the next phase, and so each node claims success and becomes a TN. Deadlocks are of two types: isolated and non-isolated. An isolated deadlock is a condition where none of the deadlocked nodes is connected to any non-deadlocked node. In the non-isolated deadlock situation, at least one deadlocked node is connected to a non-deadlocked neighbor node. The RA phase can resolve isolated deadlocks. None of the nodes transmits RA, and hence the TNs abort their transmissions.

Packing/elimination (P/E) phase:

In this phase, a packing packet (PP) is sent by each node that is located within two hops from a TN, and that had made a reservation since the previous P/E phase. A node receiving a PP understands that there has been a recent success in slot reservation three hops away from it, and because of this some of its neighbors would have been blocked during this slot. The node can take advantage of this and adjust its contention probability p, so that convergence is faster.

In an attempt to resolve a non-isolated deadlock, each TN is required to transmit an elimination packet (EP) in this phase, with a probability 0.5. A deadlocked TN, on receiving an EP before transmitting its own EP, gets to know about the deadlock. It backs off by marking the slot as reserved and does not transmit further during the slot.

Consider [Figure 4.22](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig22). Here nodes 1, 7, and 9 have packets ready to be transmitted to nodes 4, 8, and 10, respectively. During the reservation request phase, all three nodes transmit RR packets. Since no other node in the two-hop neighborhood of node 1 transmits simultaneously, node 1 does not receive any CR message in the collision report phase. So node 1 transmits an RC message in the next phase, for which node 4 sends back an RA message, and the reservation is established. Node 7 and node 9 both transmit the RR packet in the reservation request phase. Here node 9 is within two hops from node 7. So if both nodes 7 and 9 transmit simultaneously, their RR packets collide at common neighbor node 11. Node 11 sends a CR packet which is heard by nodes 7 and 9. On receiving the CR packet, nodes 7 and 9 stop contending for the current slot.



***Figure 4.22. FPRP - Example.***

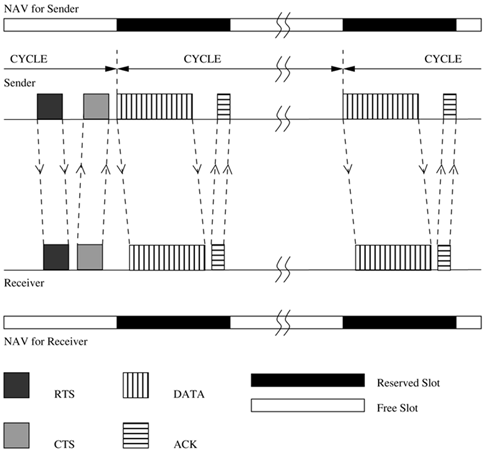
The reservation process in FPRP is simple. No information needs to be distributed to nodes other than the one-hop neighbor nodes before the reservation becomes successful.

**4.4.6 MACA with Piggy-Backed Reservation**

MACA with piggy-backed reservation (MACA/PR) [[15](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry15)] is a protocol used to provide real-time traffic support in multi-hop wireless networks. The MAC protocol used is based on the MACAW protocol [[2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry02)], with the provisioning of non-persistent CSMA (as in FAMA [[3](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry03)]). The main components of MACA/PR are: a MAC protocol, a reservation protocol, and a QoS routing protocol. MACA/PR differentiates real-time packets from the best-effort packets. While providing guaranteed bandwidth support for real-time packets, at the same time it provides reliable transmission of best-effort packets. Time is divided into slots. The slots are defined by the reservations made at nodes, and hence are asynchronous in nature with varying lengths. Each node in the network maintains a reservation table (RT) that records all the reserved transmit and receive slots/windows of all nodes within its transmission range.

In order to transmit a non-real-time packet, a MACAW [[2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry02)]-based MAC protocol is used. The ready node (a node which has packets ready for transmission) first waits for a free slot in the RT. Once it finds a free slot, it again waits for an additional random time of the order of a single-hop round-trip delay time, after which it senses the channel. If the channel is found to be still free, the node transmits an RTS packet, for which the receiver, if it is ready to receive packets, responds with a CTS packet. On receiving the CTS packet, the source node sends a DATA packet, and the receiver, on receiving the packet without any error, finally sends an ACK packet back to the source. The RTS and CTS control packets contain, in them, the time duration in which the DATA packet is to be transmitted. A nearby node that hears these packets avoids transmission during that time. If, after the random waiting time, the channel is found to be busy, the node waits for the channel to become idle again, and then repeats the same procedure.

For real-time traffic, the reservation protocol of MACA/PR functions as follows. The sender is assumed to transmit real-time packets at certain regular intervals, say, every CYCLE time period. The first data packet of the session is transmitted in the usual manner just as a best-effort packet would be transmitted. The source node first sends an RTS packet, for which the receiver node responds with a CTS packet. Now the source node sends the first DATA packet of the real-time session. Reservation information for the next DATA packet to be transmitted (which is scheduled to be transmitted after CYCLE time period) is piggy-backed on this current DATA packet. On receiving this DATA packet, the receiver node updates its reservation table with the piggy-backed reservation information. It then sends an ACK packet back to the source. The receiver node uses the ACK packet to confirm the reservation request that was piggy-backed on the previous DATA packet. It piggy-backs the reservation confirmation information on the ACK packet. Neighbor nodes that hear the DATA and ACK packets update their reservation tables with the reservation information carried by them, and refrain from transmitting when the next packet is to be transmitted. Unlike MACAW [[2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry02)], MACA/PR does not make use of RTS/CTS packets for transmission of the subsequent DATA packets. After receiving the ACK, the source node directly transmits the next DATA packet at its scheduled transmission time in the next CYCLE. This DATA packet in turn would carry reservation information for the next DATA packet. Real-time data transmission, hence, occurs as a series of DATA-ACK packet exchanges. The real-time packets (except for the first packet of the session that is used to initiate the reservation process) are transmitted only once. If an ACK packet is not received for a DATA packet, the source node just drops the packet. The ACK packet therefore serves the purpose of renewing the reservation, in addition to recovering from packet loss. If the source node fails to receive ACK packets for a certain number of consecutive DATA packets, it then assumes the reservation to have been lost. It restarts the real-time session again with an RTS-CTS control packet exchange, either on a different slot on the same link, or on a different link in case of a path break. In order to transmit an RTS to the receiver node, the source needs to find a slot that is free at both the nodes. For maintaining consistent information regarding free slots at all nodes, MACA/PR uses periodic exchange of reservation tables. This periodic table exchange automatically overcomes the hidden terminal problem. When a hidden terminal receives a reservation table from a node, it refrains from transmitting in the reserved slots of that node. Slot reservation information maintained in the reservation tables is refreshed every cycle. If the reservation is not refreshed for a certain number of consecutive cycles, it is then dropped. The transmission of packets in MACA/PR is depicted in [Figure 4.23](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig23). It can be seen from the figure that the RTS-CTS exchange is used only for transmitting the first packet of the session. Since each DATA packet carries reservation information for the next DATA packet that would be transmitted after CYCLE time period, RTS-CTS exchange is not required for the subsequent DATA packets. Neighbor nodes that receive DATA packets update their reservation tables accordingly and do not contend for the channel during the reserved slots. The network allocation vector (NAV) at each node reflects the current and future state of the channel as perceived by the node.



***Figure 4.23. Packet transmission in MACA/PR.***

Best-effort packet transmissions and real-time packet transmissions can be interleaved at nodes, with higher priority being given to real-time packets. For real-time packets, MACA/PR effectively works as a TDM system, with a superframe time of CYCLE. The best-effort packets are transmitted in the empty slots (which have not been reserved) of the cycle.

When a new node joins the network, it initially remains in the listening mode during which it receives reservation tables from each of its neighbors and learns about the reservations made in the network. After this initial period, the node shifts to its normal mode of operation.

The QoS routing protocol used with MACA/PR is the destination sequenced distance vector (DSDV) routing protocol [[16](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry16)] (described in detail in [Chapter 7](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch07.html)). Bandwidth constraint has been introduced in the routing process. Each node periodically broadcasts to its neighbors the (bandwidth, hop distance) pairs for each preferred path, that is, for each bandwidth value, to each destination. The number of preferred paths is equal to the maximum number of slots in a cycle. After this is done, if a node receives a real-time packet with a certain bandwidth requirement that cannot be satisfied using the current available paths, the packet is dropped and no ACK packet is sent. The sender node would eventually reroute the packet.

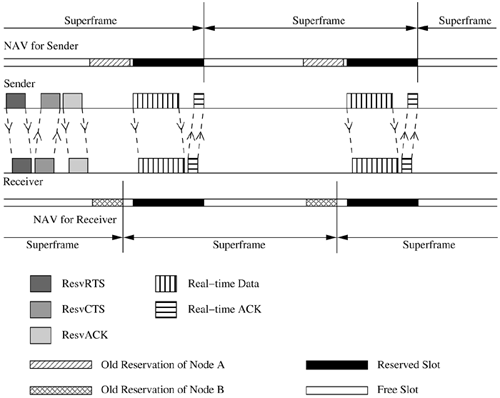
Thus, MACA/PR is an efficient bandwidth reservation protocol that can support real-time traffic sessions. One of the important advantages of MACA/PR is that it does not require global synchronization among nodes. A drawback of MACA/PR is that a free slot can be reserved only if it can fit in the entire RTS-CTS-DATA-ACK exchange. Therefore, there is a possibility of many fragmented free slots not being used at all, reducing the bandwidth efficiency of the protocol.

**4.4.7 Real-Time Medium Access Control Protocol**

The real-time medium access control protocol (RTMAC) [[17](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry17)] provides a bandwidth reservation mechanism for supporting real-time traffic in ad hoc wireless networks. RTMAC consists of two components, a MAC layer protocol and a QoS routing protocol. The MAC layer protocol is a real-time extension of the IEEE 802.11 DCF. The QoS routing protocol is responsible for end-to-end reservation and release of bandwidth resources. The MAC layer protocol has two parts: a medium-access protocol for best-effort traffic and a reservation protocol for real-time traffic.

A separate set of control packets, consisting of ResvRTS, ResvRTSResvCTS, and ResvACK, is used for effecting bandwidth reservation for real-time packets. RTS, CTS, and ACK control packets are used for transmitting best-effort packets. In order to give higher priority for real-time packets, the wait time for transmitting a ResvRTS packet is reduced to half of DCF inter-frame space (DIFS), which is the wait time used for best-effort packets.

Time is divided into superframes. As can be seen from [Figure 4.24](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig24), the superframe for each node may not strictly align with the other nodes. Bandwidth reservations can be made by a node by reserving variable-length time slots on superframes, which are sufficient enough to carry the traffic generated by the node. The core concept of RTMAC is the flexibility of slot placement in the superframe. Each superframe consists of a number of reservation-slots (resv-slots). The time duration of each resv-slot is twice the maximum propagation delay. Data transmission normally requires a block of resv-slots. A node that needs to transmit real-time packets first reserves a set of resv-slots. The set of resv-slots reserved by a node for a connection on a superframe is called a connection-slot. A node that has made reservations on the current superframe makes use of the same connection-slot in the successive superframes for transmitting packets. Each node maintains a reservation table containing information such as the sender id, receiver id, and starting and ending times of reservations that are currently active within its direct transmission range.



***Figure 4.24. Reservation mechanism in RTMAC.***

In RTMAC, no time synchronization is assumed. The protocol uses relative time for all reservation purposes. When a node receives this relative-time-based information, it converts the relative time to absolute time by adding its current time maintained in its clock. A three-way handshake protocol is used for effecting the reservation. For example, node A, which wants to reserve a slot with node B, sends a ResvRTS packet which contains the relative time information of starting and ending of the connection-slot (a number of resv-slots) to be reserved. Node B, on receiving this packet, first checks its reservation table to see whether it can receive on those resv-slots. If it can, it replies by sending a ResvCTS packet containing the relative time information of the same resv-slots to be reserved. Neighbor nodes of the receiver, on receiving the ResvCTS, update their reservation tables accordingly. Source node A, on receiving the ResvCTS packet, responds by sending a ResvACK packet. This packet also carries relative time information regarding the reserved slots. The ResvACK packet is meant for the neighbor nodes of the source node (node A) which are not aware of the reservation as they may not receive the ResvCTS packet. Such nodes update their reservation tables on receiving the ResvACK packet. Transmission of the ResvACK packet completes the reservation process. Once the reservation is made, real-time packets are transmitted in these reserved slots. Transmission of each real-time packet is followed by the transmission of a real-time ACK (RTACK) packet by the receiver.

The bandwidth reservation process is illustrated in [Figure 4.24](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig24). In the figure, NAV indicates the network allocation vector maintained at each node. As mentioned earlier in [Section 4.4.6](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06lev2sec22), the NAV at a node reflects the current and future state of the channel as perceived by the node. The sender node first transmits a ResvRTS packet indicating the connection-slot (represented by the offset time from the current time for the beginning and end of the connection-slot) to be reserved. The receiver node on receiving this packet checks its NAV and finds that the requested connection-slot is free. So it responds by sending a ResvCTS packet carrying the same connection-slot information. The sender node, on receiving this packet, completes the reservation process by sending a ResvACK packet. The corresponding connection-slot is marked as reserved at both the sender and the receiver. This is indicated in [Figure 4.24](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig24) by the dark-shaded regions in the NAVs of the sender and receiver. Once the reservation is made, the real-time session gets started, and packets are transmitted in the reserved connection-slot by means of Real-timeData – Real-timeACK exchanges.

If the receiver node receives the ResvRTS packet on a slot which has already been reserved by one of its neighbor nodes, it does not respond with a ResvCTS packet. It just discards the received ResvRTSpacket. This is because, if the node responds with a negative or positive ACK, the ACK packet may cause collisions with the reservations made by its neighbor. The sender node times out and retries later. In case the ResvRTS is received on a free slot, but the requested connection-slot is not free at the receiver node, the receiver sends a negative CTS (ResvNCTS) back to the sender node. On receiving this, the sender node reattempts following the same procedure but with another free connection-slot.

If the real-time session gets finished, or a route break is detected by the sender node, the node releases the resources reserved for that session by sending a reservation release RTS (ResvRelRTS) packet. The ResvRelRTS packet is a broadcast packet. Nodes hearing this packet update their reservation tables in order to free the corresponding connection slots. In case the receiver node receives this (ResvRelRTS) packet, it responds by broadcasting a (ResvRelCTS) packet. The receiver's neighbor nodes, on receiving this (ResvRelCTS) packet, free up the corresponding reservation slots. In case the downstream node of the broken link does not receive the ResvRelRTS packet, since it also does not receive any DATA packet belonging to the corresponding connection, it times out and releases the reservations made.

A QoS routing protocol is used with RTMAC to find an end-to-end path that matches the QoS requirements (bandwidth requirements) of the user. The QoS routing protocol used here is an extension of the destination sequenced distance vector (DSDV) routing protocol. When a node receives a data packet for a new connection, the node reserves bandwidth on the forward link and forwards the packet to the next node on the path to the destination. In order to maintain a consistent view of reservation tables of neighboring nodes at each node, each node transmits its reservation information along with the route update packet, which is defined as part of the DSDV protocol. The routing protocol can specify a specific connection-slot to be reserved for a particular connection; this gives flexibility for the routing protocol to decide on the positioning of the connection-slot. But generally, the first available connection-slot is used.

One of the main advantages of RTMAC is its bandwidth efficiency. Since nodes operate in the asynchronous mode, successive reservation slots may not strictly align with each other. Hence small fragments of free slots may occur in between reservation slots. If the free slot is just enough to accommodate a DATA and ACK packet, then RTMAC can make use of the free slot by transmitting ResvRTS-ResvCTS-ResvACK in some other free slot. Such small free slots cannot be made use of in other protocols such as MACA/PR, which require the free slot to accommodate the entire RTS-CTS-DATA-ACK exchange. Another advantage of RTMAC is its asynchronous mode of operation where nodes do not require any global time synchronization.

**4.7 CONTENTION-BASED MAC PROTOCOLS WITH SCHEDULING MECHANISMS**

Protocols that fall under this category focus on packet scheduling at the nodes and transmission scheduling of the nodes. Scheduling decisions may take into consideration various factors such as delay targets of packets, laxities of packets, traffic load at nodes, and remaining battery power at nodes. In this section, some of the scheduling-based MAC protocols are described.

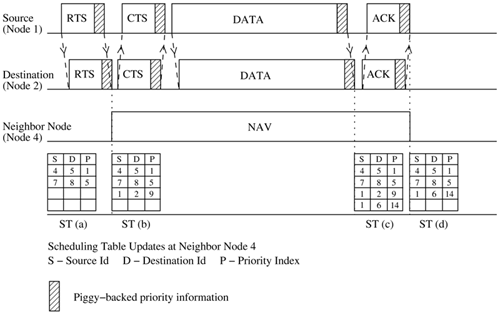
**4.7.1 Distributed Priority Scheduling and Medium Access in Ad Hoc Networks**

This work, proposed in [[18](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry18)], presents two mechanisms for providing quality of service (QoS) support for connections in ad hoc wireless networks. The first technique, called distributed priority scheduling (DPS), piggy-backs the priority tag of a node's current and head-of-line packets on the control and data packets. By retrieving information from such packets transmitted in its neighborhood, a node builds a scheduling table from which it determines its rank (information regarding its position as per the priority of the packet to be transmitted next) compared to other nodes in its neighborhood. This rank is incorporated into the back-off calculation mechanism in order to provide an approximate schedule based on the ranks of the nodes. The second scheme, called multi-hop coordination, extends the DPS scheme to carry out scheduling over multi-hop paths. The downstream nodes in the path to the destination increase the relative priority of a packet in order to compensate for the excessive delays incurred by the packet at the upstream nodes.

**Distributed Priority Scheduling**

The distributed priority scheduling scheme (DPS) is based on the IEEE 802.11 distributed coordination function. DPS uses the same basic RTS-CTS-DATA-ACK packet exchange mechanism. The RTS packet transmitted by a ready node carries the priority tag/priority index for the current DATA packet to be transmitted. The priority tag can be the delay target for the DATA packet. On receiving the RTS packet, the intended receiver node responds with a CTS packet. The receiver node copies the priority tag from the received RTS packet and piggy-backs it along with the source node id, on the CTS packet. Neighbor nodes receiving the RTS or CTS packets (including the hidden nodes) retrieve the piggy-backed priority tag information and make a corresponding entry for the packet to be transmitted, in their scheduling tables (STs). Each node maintains an ST holding information about packets, which were originally piggy-backed on control and data packets. The entries in the ST are ordered according to their priority tag values. When the source node transmits a DATA packet, its head-of-line packet information (consisting of the destination and source ids along with the priority tag) is piggy-backed on the DATA packet (head-of-line packet of a node refers to the packet to be transmitted next by the node). This information is copied by the receiver onto the ACK packet it sends in response to the received DATA packet. Neighbor nodes receiving the DATA or ACK packets retrieve the piggy-backed information and update their STs accordingly. When a node hears an ACK packet, it removes from its ST any entry made earlier for the corresponding DATA packet.

Figure 4.25 illustrates the piggy-backing and table update mechanism. Node 1 needs to transmit a DATA packet (with priority index value 9) to node 2. It first transmits an RTS packet carrying piggy-backed information about this DATA packet. The initial state of the ST of node 4 which is a neighbor of nodes 1 and 2 is shown in ST (a). Node 4, on hearing this RTS packet, retrieves the piggybacked priority information and makes a corresponding entry in its ST, as shown in ST (b). The destination node 2 responds by sending a CTS packet. The actual DATA packet is sent by the source node once it receives the CTS packet. This DATA packet carries piggy-backed priority information regarding the head-of-line packet at node 1. On hearing this DATA packet, neighbor node 4 makes a corresponding entry for the head-of-line packet of node 1, in its ST. ST (c) shows the new updated status of the ST at node 4. Finally, the receiver node sends an ACK packet to node 1. When this packet is heard by node 4, it removes the entry made for the corresponding DATA packet from its ST. The state of the scheduling table at the end of this data transfer session is depicted in ST (d).



***Figure 4.25. Piggy-backing and scheduling table update mechanism in DPS.***

In essence, each node's scheduling table gives the rank of the node with respect to other nodes in its neighborhood. This rank information is used to determine the back-off period to be taken by the node. The back-off distribution is given by



whereCWmin is the minimum size of the contention window. r is the rank in the scheduling table of the node's highest priority packet; n is the current number of transmission attempts made by the node; nmax is the maximum number of retransmissions permitted; α is a constant; and γ is a constant that is used to control the congestion in the second attempt for the highest ranked nodes.

**Multi-Hop Coordination**

By means of the multi-hop coordination mechanism, the excess delay incurred by a packet at the upstream nodes is compensated for at the downstream nodes. When a node receives a packet, it would have already received the priority index of the packet piggy-backed on the previous RTS packet. In case the node is an intermediate node which has to further forward the packet, the node calculates the new priority index of the DATA packet in a recursive fashion, based on the received value of the priority index. If is the priority index assigned to the kth packet of flow i with size at its jth hop, and if is the time at which the kth packet of flow i arrives at its first hop (the next hop node to the source node on the path to the destination), then the new priority index assigned to the received packet at intermediate node j is given as



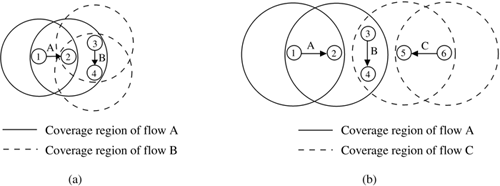
where the increment of the priority index is a non-negative function of i, j,**, and . Because of this mechanism, if a packet suffers due to excess delay at the upstream nodes, the downstream nodes increase the priority of the packet so that the packet is able to meet its end-to-end delay target. Similarly, if a packet arrives very early due to lack of contention at the upstream nodes, then the priority of that packet would be reduced at the downstream nodes. Any suitable scheme can be used for obtaining the values for . One simple scheme, called uniform delay budget allocation, works as follows. For a flow i with an end-to-end delay target of D, the increment in priority index value for a packet belonging to that flow, at hop j, is given as, , where K is the length of the flow's path.

The distributed priority scheduling and multi-hop coordination schemes described above are fully distributed schemes. They can be utilized for carrying time-sensitive traffic on ad hoc wireless networks.

**4.7.2 Distributed Wireless Ordering Protocol**

The distributed wireless ordering protocol (DWOP) [[19](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry19)] consists of a media access scheme along with a scheduling mechanism. It is based on the distributed priority scheduling scheme proposed in [[18](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry18)]. DWOP ensures that packets access the medium according to the order specified by an ideal reference scheduler such as first-in-first-out (FIFO), virtual clock, or earliest deadline first. In this discussion, FIFO is chosen as the reference scheduler. In FIFO, packet priority indices are set to the arrival times of packets. Similar to DPS [[18](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry18)], control packets are used in DWOP to piggy-back priority information regarding head-of-line packets of nodes. As the targeted FIFO schedule would transmit packets in order of the arrival times, each node builds up a scheduling table (ST) ordered according to the overheard arrival times.

The key concept in DWOP is that a node is made eligible to contend for the channel only if its locally queued packet has a smaller arrival time compared to all other arrival times in its ST (all other packets queued at its neighbor nodes), that is, only if the node finds that it holds the next region-wise packet in the hypothetical FIFO schedule. Two additional table management techniques, receiver participation and stale entry elimination, are used in order to keep the actual schedule close to the reference FIFO schedule. DWOP may not suffer due to information asymmetry. Since in most networks all nodes are not within the radio range of each other, a transmitting node might not be aware of the arrival times of packets queued at another node which is not within its direct transmission range. This information asymmetry might affect the fair sharing of bandwidth. For example, in [Figure 4.26 (a)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig26), the sender of flow B would be aware of the packets to be transmitted by the sender of flow A, and so it defers its transmission whenever a higher priority packet is queued at the sender of flow A. But the sender of flow A is not aware of the arrival times of packets queued at the sender of flow B and hence it concludes that it has the highest priority packet in its neighborhood. Therefore, node 1 unsuccessfully tries to gain access to the channel continuously. This would result in flow B receiving an unfair higher share of the available bandwidth. In order to overcome this information asymmetry problem, the receiver participation mechanism is used.



***Figure 4.24. (a) Information asymmetry. (b) Perceived collisions.***

In the receiver participation mechanism, a receiver node, when using its ST information, finds that the sender is transmitting out of order, that is, the reference FIFO schedule is being violated, anout-of-order notification is piggy-backed by the receiver on the control packets (CTS/ACK) it sends to the sender. In essence, information regarding the transmissions taking place in the two-hop neighborhood of the sender is propagated by the receiver node whenever it detects a FIFO schedule violation. Since the notification is sent only when a FIFO violation is detected, the actual transmission may not strictly follow the FIFO schedule; rather, it approximates the FIFO schedule. On receiving an out-of-order packet from a sender node, the receiver node transmits a notification to the sender node carrying the actual rank R of the sender with respect to the receiver's local ST. On receiving this out-of-order notification, the sender node goes into a back-off state after completing the transmission of its current packet. The back-off period Tback−off is given by



whereTsuccess is the longest possible time required to transmit a packet successfully, including the RTS-CTS-DATA-ACK handshake. Thus the node backs off, allowing higher priority packets in the neighborhood of the receiver to get transmitted first. In order to obtain a perfect FIFO schedule, the receiver can very well be made not to reply to the out-of-order requests (RTS) of the sender. This would cause the sender to time out and back off, thereby avoiding any out-of-order transmission. But since the sender has already expended its resources in transmitting the RTS successfully, it is allowed to complete the transmission of its current packet. This is a trade-off between achieving perfect FIFO scheduling and high system utilization. Since in DWOP a node's access to the medium is dependent on its rank in the receiver node's ST (the rank of a node denotes the position of the node's entry in the receiver node's ST as per the priority of the corresponding packet), information maintained in the ST must be consistent with the actual network scenario. The stale entry elimination mechanism makes sure that the STs are free of stale entries. An entry is deleted from the ST only after an ACK packet for the corresponding entry is heard by the node. In case the ACK packet collides at the node, the corresponding entry in the ST will never be removed. This may cause a large deviation from the ideal FIFO schedule. [Figure 4.26 (b)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig26) shows an example-perceived collisions scenario. The sender and receiver of flow B might have stale entries because of collisions caused by packets belonging to flow A and flow C at the sender and receiver of flow B. It can be observed that, in case there is a stale entry in the ST of a node, the node's own head-of-line packet position remains fixed, while other entries below the head-of-line entry keep changing. The above observation is used as a stale entry detection method. Thus, when a node observes that its rank remains fixed while packets whose priorities are below the priority of its head-of-line packet are being transmitted, it concludes that it may have one or more stale entries in its ST. The node simply deletes the oldest entry from its ST, assuming it to be the stale entry. This mechanism thus eliminates stale entries from the STs of nodes.

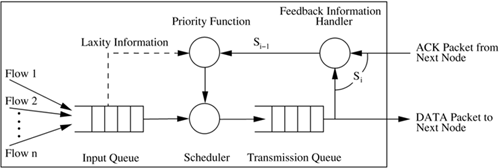
In summary, DWOP tries to ensure that packets get access to the channel according to the order defined by a reference scheduler. The above discussion was with respect to the FIFO scheduler. Though the actual schedule deviates from the ideal FIFO schedule due to information asymmetry and stale information in STs, the receiver participation and the stale entry elimination mechanisms try to keep the actual schedule as close as possible to the ideal schedule.

**4.7.3 Distributed Laxity-Based Priority Scheduling Scheme**

The distributed laxity-based priority scheduling (DLPS) scheme [[20](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry20)] is a packet scheduling scheme, where scheduling decisions are made taking into consideration the states of neighboring nodes and the feedback from destination nodes regarding packet losses. Packets are reordered based on their uniform laxity budgets (ULBs) and the packet delivery ratios of the flows to which they belong.

Each node maintains two tables: scheduling table (ST) and packet delivery ratio table (PDT). The ST contains information about packets to be transmitted by the node and packets overheard by the node, sorted according to their priority index values. Priority index expresses the priority of a packet. The lower the priority index, the higher the packet's priority. The PDT contains the count of packets transmitted and the count of acknowledgment (ACK) packets received for every flow passing through the node. This information is used for calculating current packet delivery ratio of flows (explained later in this section).

A node keeps track of packet delivery ratios (used for calculating priority index of packets) of all flows it is aware of by means of a feedback mechanism. [Figure 4.27](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig27) depicts the overall functioning of the feedback mechanism. Incoming packets to a node are queued in the node's input queue according to their arrival times. The scheduler sorts them according to their priority values and inserts them into the transmission queue. The highest priority packet from this queue is selected for transmission. The node, after transmitting a packet, updates the count of packets transmitted so far in its PDT. The destination node of a flow, on receiving data packets, initiates a feedback by means of which the count of DATA packets received by it is conveyed to the source through ACK packets traversing the reverse path. These two pieces of information, together denoted by Si in [Figure 4.27](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig27), are received by the feedback information handler (FIH). The FIH, in parallel, also sends the previous state information Si−1 to the priority function module (PFM). The ULB of each packet in ST is available at the node (ULB calculation will be explained later in this section). This information is also sent to PFM, which uses the information fed to it to calculate the priority indices of packets in the ST.



***Figure 4.27. Feedback mechanism.* Reproduced with permission from [**[**20**](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry20)**], © Elsevier, 2004.**

Using the count of DATA packets transmitted (pktsSent) and count information carried by ACK packets (acksRcvd), available in PDT, packet delivery ratio (PDR) of the flow at any given time is computed as

C:\Users\Pavan\Desktop\WSN\wsnmaterials\wsn\Chapter 6. MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS - Ad Hoc Wireless Networks Architectures and Protocols_files\06equ01.gif(4.7.1)

Priority index of a packet (PI) is defined as

C:\Users\Pavan\Desktop\WSN\wsnmaterials\wsn\Chapter 6. MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS - Ad Hoc Wireless Networks Architectures and Protocols_files\06equ02.gif(4.7.2)

Here, is the uniform laxity budget of the packet, and M is a user-defined parameter representing the desired packet delivery ratio for the flow. deadline is the end-to-end deadline target of the packet and is equal to (packet creation time + end-to-end delay target). currentTime denotes the current time according to the node's local clock.

When greater numbers of packets belonging to a flow meet their delay targets, the term would have a high value. Hence priority index would be high for packets of that flow, and therefore the actual priority of the packets would be low. When very few packets of a flow meet their delay targets, the value of would be much less, thereby lowering the priority index and increasing the priority of packets of that flow. ULB also plays an equally important role. Since remHops, the number of hops remaining to be traversed, is in the denominator of the expression for ULB, when a packet is near its source and needs to traverse several hops to reach its destination, its priority index value will be lowered, thereby increasing its priority. When it nears its destination, the fewer number of hops to be traversed tends to increase the priority index, thereby lowering its priority.

RTS and CTS packets transmitted by a node are modified to carry piggy-backed information regarding the highest priority packet queued at the node. Similarly, DATA and ACK packets transmitted by a node carry piggy-backed information corresponding to the highest priority packet entry in theSTof the node. A node hearing any packet retrieves the piggy-backed priority information, calculates the priority index of the corresponding packet, and adds a corresponding entry in its ST.

A DATA packet also carries information about itself. The end-to-end delay target, remaining number of hops, actual source ID, and the flow ID constitute this information. A node, on receiving a DATA packet, using the above information and the information maintained in the PDT, can obtain the priority index of the packet (PI), as given in [Equation 4.7.2](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06equ02). Since the priority index of a packet keeps changing with time, it needs to be updated constantly. Each node, before calculating its backoff period and before inserting a new entry into its ST, recalculates and updates the priority index of each entry in its ST.

When a node hears a DATA packet, if an entry for the corresponding packet exists in its ST, then that entry is deleted. The sender node deletes its entry from the ST only when an ACK for the transmitted DATA packet is received. It may happen that a DATA packet transmitted is not heard by a node which had previously been located within the transmission range of a sender node holding the highest priority packet in its locality. This might be because of reasons such as node mobility and channel errors. In such cases, the stale entries might affect the desired scheduling of packets. Another reason for stale entries in the ST is that, when the network load is high, some of the packets would miss their deadline targets while waiting in the node's queue itself. Such packets will never be transmitted. In order to remove stale entries, whenever table updates are performed, entries whose deadline targets have been missed already are deleted from the ST.

The objective of the back-off mechanism used in DLPS is to reflect the priority of the node's highest priority packet on the back-off period to be taken by the node. If r is the rank (rank of an entry is the position of that entry in the scheduling table of the node), in ST of the node, of the current packet to be sent, n is the number of retransmission attempts made for the packet, and nmax is the maximum number of retransmission attempts permitted, then the back-off interval is given by



(4.7.3)(4.7.4)(4.7.5)

whereCWmin is the minimum size of the contention window, and M is the desired packet delivery ratio.

This means that if the packet has the highest rank in the broadcast region of the node, then it has the lowest back-off period according to [Equation 4.7.3](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06equ03) and faces much less contention. Else, if it is the first time the packet is being transmitted, then the back-off distribution follows the second scheme as in [Equation 4.7.4](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06equ03), where the back-off is more than that for the first case. Here the current PDR of the flow affects the back-off period. If PDR is considerably less, then the first term would be less, and if it is high, then the first term would be high and the node would have to wait for a longer time. Finally, if the packet does not fit into these two categories, then the back-off value is as per the third scheme in [Equation 4.7.5](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06equ03), and is the longest of the three cases. The higher the value of ULB, the longer the back-off period.

DLPS delivers a higher percentage of packets within their delay targets and has lower average end-to-end delay in packet delivery when compared to the 802.11 DCF and the DPS [[18](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry18)] schemes..

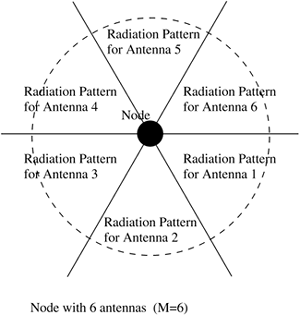
**4.8 MAC PROTOCOLS THAT USE DIRECTIONAL ANTENNAS**

MAC protocols that use directional antennas for transmissions have several advantages over those that use omnidirectional transmissions. The advantages include reduced signal interference, increase in the system throughput, and improved channel reuse that leads to an increase in the overall capacity of the channel. In this section, some of the MAC layer protocols that make use of directional antennas are discussed.

**4.8.1 MAC Protocol Using Directional Antennas**

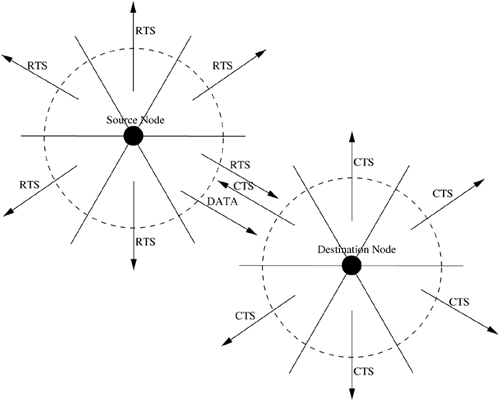
The MAC protocol for mobile ad hoc networks using directional antennas that was proposed in [[21](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry21)] makes use of directional antennas to improve the throughput in ad hoc wireless networks. The mobile nodes do not have any location information by means of which the direction of the receiver and sender nodes could be determined. The protocol makes use of an RTS/CTS exchange mechanism, which is similar to the one used in MACA [[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry01)]. The nodes use directional antennas for transmitting and receiving data packets, thereby reducing their interference to other neighbor nodes. This leads to an increase in the throughput of the system.

Each node is assumed to have only one radio transceiver, which can transmit and receive only one packet at any given time. The transceiver is assumed to be equipped with M directional antennas, each antenna having a conical radiation pattern, spanning an angle of radians ([Figure 4.28](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig28)). It is assumed that the transmissions by adjacent antennas never overlap, that is, the complete attenuation of the transmitted signal occurs outside the conical pattern of the directional antenna. The MAC protocol is assumed to be able to switch every antenna individually or all the antennas together to the active or passive modes. The radio transceiver uses only the antennas that are in the active mode. If a node transmits when all its antennas are active, then the transmission's radiation pattern is similar to that of an omnidirectional antenna. The receiver node uses receiver diversity while receiving on all antennas. This means that the receiver node uses the signal from the antenna which receives the incoming signal at maximum power. In the normal case, this selected antenna would be the one whose conical pattern is directed toward the source node whose signal it is receiving. It is assumed that the radio range is the same for all directional antennas of the nodes. In order to detect the presence of a signal, a threshold signal power value is used. A node concludes that the channel is active only if the received signal strength is higher than this threshold value.



***Figure 4.28. Radiation patterns of directional antennas.***

The protocol works as follows. The packet-exchange mechanism followed for transmitting each data packet is depicted in [Figure 4.29](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig29). In the example shown in the figure, each node is assumed to have six directional antennas. The main concept in this protocol is the mechanism used by the transmitting and receiving nodes to determine the directions of each other. The MAC layer at the source node must be able to find the direction of the intended next-hop receiver node so that the data packet could be transmitted through a directional antenna. It is the same case with the receiver. The receiver node must be able to determine the direction of the sender node before starting to receive data packets. This is performed in the following manner. An idle node is assumed to be listening to the on-going transmissions on all its antennas. The sender node first transmits an RTS packet addressed to the receiver. This RTS is transmitted through all the antennas of the node (omnidirectional transmission). The intended receiver node, on receiving this RTS packet, responds by transmitting a CTS packet, again on all its antennas (omnidirectional transmission). The receiver node also notes down the direction of the sender by identifying the antenna that received the RTS packet with maximum power. The source, on receiving the CTS packet, determines the direction of the receiver node in a similar manner. The neighbor nodes that receive the RTS or CTS packets defer their transmissions for appropriate periods of time. After receiving the CTS, the source node transmits the next data packet through the chosen directional antenna. All other antennas are switched off and remain idle. The receiver node receives this data packet only through its selected antenna.



***Figure 4.29. Packet transmission.***

Since a node transmits packets only through directional antennas, the interference caused to nodes in its direct transmission range is reduced considerably, which in turn leads to an increase in the overall throughput of the system.

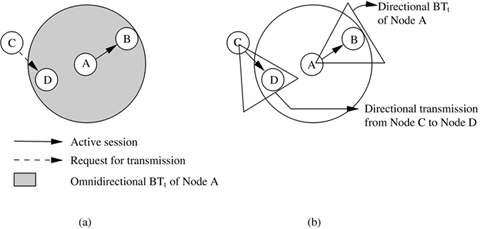
**4.8.2 Directional Busy Tone-Based MAC Protocol**

The directional busy tone-based MAC protocol [[22](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry22)] adapts the DBTMA protocol [[5](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry05)] for use with directional antennas. It uses directional antennas for transmitting the RTS, CTS, and data frames, as well as the busy tones. By doing so, collisions are reduced significantly. Also, spatial reuse of the channel improves, thereby increasing the capacity of the channel.

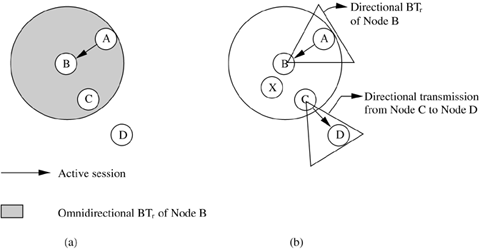
Each node has a directional antenna which consists of N antenna elements, each covering a fixed sector spanning an angle of (360/N) degrees. For a unicast transmission, only a single antenna element is used. For broadcast transmission, all the N antenna elements transmit simultaneously. When a node is idle (not transmitting packets), all antenna elements of the node keep sensing the channel. The node is assumed to be capable of identifying the antenna element on which the incoming signal is received with maximum power. Therefore, while receiving, exactly one antenna element collects the signals. In an ad hoc wireless network, nodes may be mobile most of the time. It is assumed that the orientation of sectors of each antenna element remains fixed. The protocol uses the same two busy tones BTt and BTr used in the DBTMA protocol. The purpose of the busy tones is the same. Before transmitting an RTS packet, the sender makes sure that the BTr tone is not active in its neighborhood, so that its transmissions do not interfere with packets being received at a neighboring receiver node. Similarly, a receiver node, before transmitting a CTS, verifies that a BTt is not active in its neighborhood. This is done to make sure that the data the node is expected to receive does not collide with any other on-going transmission. The modified directional DBTMA protocol operates as follows.

A node that receives a data packet for transmission first transmits an RTS destined to the intended receiver in all directions (omnidirectional transmission). On receiving this RTS, the receiver node determines the antenna element on which the RTS is received with maximum gain. The node then sends back a directional CTS to the source using the selected antenna element (which points toward the direction of the sender). It also turns on the busy tone BTr in the direction toward the sender. On receiving the CTS packet, the sender node turns on the BTt busy tone in the direction of the receiver node. It then starts transmitting the data packet through the antenna element on which the previous CTS packet was received with maximum gain. Once the packet transmission is over, it turns off the BTt signal. The receiver node, after receiving the data packet, turns off the BTr signal.

The directional busy tones can permit simultaneous transmissions in the neighborhood of a transmitting or a receiving node. For example, in [Figure 4.30 (a)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig30), where omnidirectional busy tones are being used, when a transmission is going on from node A to node B, node D is not permitted to receive any data as it hears the BTt tone transmitted by node A. But when directional busy tone transmissions are used ([Figure 4.30 (b)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig30)), it can be seen that node D can simultaneously receive data from node C. Another example is shown in [Figure 4.31](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig31). When omnidirectional busy tones are used ([Figure 4.31 (a)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig31)), node C is not permitted to transmit to node D when node B is receiving data from node A. But when directional busy tones are used ([Figure 4.31 (b)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig31)), node C does not receive any BTr tone from node B and so it is free to transmit to node D even while the transmission between nodes A and B is going on.



***Figure 4.30. Directional DBTMA: Example 1.***



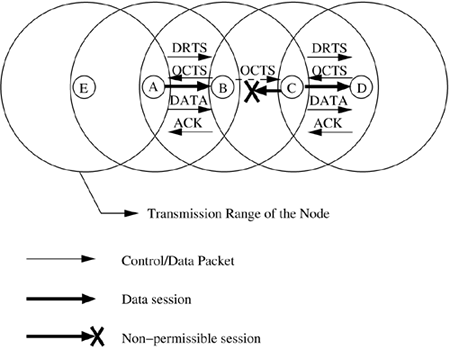
***Figure 4.31. Directional DBTMA: Example 2.***

But this scheme may cause collisions in certain scenarios. For example, in [Figure 4.31 (b)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig31), node C is free to transmit to node X when node B is receiving packets from node A. The packets transmitted to node X may collide with those being received at node B. Therefore, this protocol is not guaranteed to be collision-free. But the overall performance of the protocol (which uses directional busy tone transmissions) is better than that of the DBTMA protocol. In the case of vehicle-mounted nodes, the basic assumption that the orientation of sectors of each antenna element remains fixed may not be valid.

**4.8.3 Directional MAC Protocols for Ad Hoc Wireless Networks**

Two MAC schemes using directional antennas are proposed in [[23](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry23)]. It is assumed that each node knows about the location of its neighbors as well as its own location. The physical location information can be obtained by a node using the global positioning system (GPS). In the IEEE 802.11 DCF scheme, a node that is aware of a nearby on-going transmission will not participate in a transmission itself. In the directional MAC (D-MAC) protocols proposed in [[23](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry23)], a similar logic is applied on a per-antenna basis. If a node has received an RTS or CTS packet related to an on-going transmission on a particular antenna, then that particular antenna is not used by the node till the other transmission is completed. This antenna stays blocked for the duration of that transmission. The key concept here is, though a particular antenna of a node may remain blocked, the remaining antennas of the node can be used for transmissions. This improves the throughput of the system. An omnidirectional transmission is possible only if none of the antennas of the node is blocked.

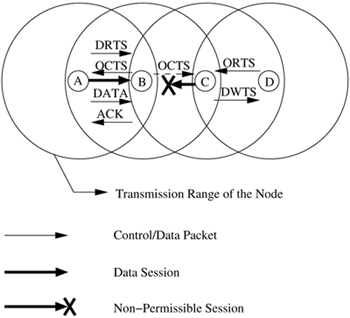
In the first directional MAC scheme (DMAC-1), a directional antenna is used for transmitting RTS packets. CTS packet transmissions are omnidirectional. Consider [Figure 4.32](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig32). Here node A, which needs to transmit a packet to node B, first transmits a directional RTS (DRTS) packet to node B. Node B, on receiving this packet, responds by transmitting an omnidirectional CTS (OCTS) packet. Once the OCTS is received without any error by node A, node A sends a data packet using a directional antenna. When node B receives the data packet, it immediately transmits a directional ACK (DACK) packet. Node C would receive the OCTS packet from node B. At node C, only the directional antenna pointing toward node B would be blocked due to this. Node C can freely transmit to node D using another directional antenna. Thus it can be seen that in DMAC-1, usage of directional antennas improves the performance by allowing simultaneous transmissions, which are not permitted when only omnidirectional antennas are used.



***Figure 4.32. Operation of DMAC protocol.***

In the second directional MAC scheme (DMAC-2) proposed in [[23](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry23)], both directional RTS (DRTS) as well as omnidirectional RTS (ORTS) transmissions are used. In DMAC-1, the usage of DRTS may increase the probability of control packet collisions. For example, consider [Figure 4.32](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig32). Here node A initiates a DRTS transmission to node B. The DRTS packet is not heard by node E, and so it is not aware of the transmission between nodes A and B. Suppose node E sends a packet to node A, then that packet may collide with the OCTS or DACK packets transmitted by node B. The probability of control packet collisions is reduced in DMAC-2. In DMAC-2, a node that wants to initiate a data transfer may send an ORTS or a DRTS as per the following two rules. (1) If none of the directional antennas at the node are blocked, then the node sends an ORTS packet. (2) Otherwise, the node sends a DRTS packet, provided the desired directional antenna is not blocked. Consider the same example in [Figure 4.32](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig32). Here when node A initiates data transfer to node B, assuming all its antennas are not blocked, it sends an ORTS packet to node B. Node E would now receive this packet, and the antenna on which the ORTS packet was received would remain blocked for the duration of the transmission from node A to node B. If node E wants to send a packet to node A, it needs to wait for the duration of the transmission between nodes A and B, so that its directional antenna pointing toward node A becomes unblocked; only then can it start transmitting packets to node A. Thus, the combination of ORTS and DRTS packets in DMAC-2 reduces collisions between control packets.

Consider [Figure 4.33](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig33). Node B sends an OCTS packet to node A after receiving DRTS from node A. Node C would be aware of the transmission and its antenna pointing toward node B would remain blocked for the duration of the transmission. Now suppose node D sends an ORTS to node C. Since one of node C's antennas is blocked currently, it would not respond to the ORTS packet. This results in node D timing out and unnecessary retransmissions of ORTS packets to node C. To avoid this situation, another packet called directional wait-to-send (DWTS) is used. On receiving the ORTS from node D, node C transmits the DWTS packet using a directional antenna toward node D. This DWTS packet carries the expected duration of the on-going transmission between nodes A and B. Node D, on receiving this packet, waits for the specified interval of time and then tries again.



***Figure 4.33. Operation of DMAC protocol.***

By enabling simultaneous collision-free transmissions, the directional MAC schemes proposed in [[23](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry23)] improve channel reuse, thereby increasing the capacity of the channel. This leads to a significant increase in the throughput of the system.

**4.9 OTHER MAC PROTOCOLS**

There are several other MAC protocols that do not strictly fall under the categories discussed above. Some of these MAC protocols are described in this section.

**4.9.1 Multichannel MAC Protocol**

The multichannel MAC protocol (MMAC) [[24](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry24)] uses multiple channels for data transmission. There is no dedicated control channel. N channels that have enough spectral separation between each other are available for data transmission.

Each node maintains a data structure called PreferableChannelList (PCL). The usage of the channels within the transmission range of the node is maintained in the PCL. Based on their usage, channels can be classified into three types.

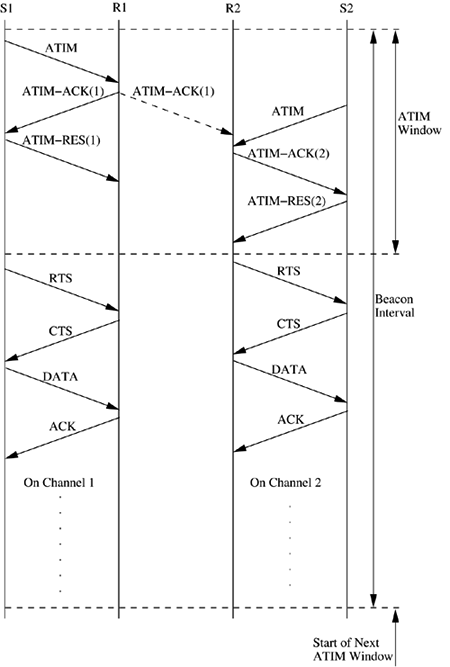
• High preference channel (HIGH): The channel has been selected by the current node and is being used by the node in the current beacon interval (beacon interval mechanism will be explained later). Since a node has only one transceiver, there can be only one HIGH channel at a time.

• Medium preference channel (MID): A channel which is free and is not being currently used in the transmission range of the node is said to be a medium preference channel. If there is no HIGH channel available, a MID channel would get the next preference.

• Low preference channel (LOW): Such a channel is already being used in the transmission range of the node by other neighboring nodes. A counter is associated with each LOW state channel. For each LOW state channel, the count of source-destination pairs which have chosen the channel for data transmission in the current beacon interval is maintained.

Time is divided into beacon intervals and every node is synchronized by periodic beacon transmissions. So, for every node, the beacon interval starts and ends almost at the same time. At the start of every beacon interval, there exists a time interval called the ad hoc traffic indication messages (ATIM) window. This window is used by the nodes to negotiate for channels for transmission during the current beacon interval. ATIM messages such as ATIM, ATIM-ACK (ATIM-acknowledgment), and ATIM-RES (ATIM-reservation) are used for this negotiation. The exchange of ATIM messages takes place on a particular channel called the default channel. The default channel is one of the multiple available channels. This channel is used for sending DATA packets outside the ATIM window, like any other channel. A node that wants to transmit in the current beacon interval sends an ATIM packet to the intended destination node. The ATIM message carries the PCL of the transmitting node. The destination node, upon receiving the packet, uses the PCL carried on the packet and its own PCL to select a channel. It includes this channel information in the ATIM-ACK packet it sends to the source node. The source node, on receiving the ATIM-ACK packet, determines whether it can transmit on the channel indicated in the ATIM-ACK message. If so, it responds by sending the destination node an ATIM-RES packet. The ATIM-ACK and ATIM-RES packets are also used to notify the neighbor nodes of the receiver and sender nodes, respectively, about the channel that is going to be used for transmission in the current beacon interval. The nodes that hear these packets update their PCLs accordingly. At the end of the ATIM window, the source and destination nodes switch to the agreed-upon channel and start communicating by exchanging RTS/CTS control packets. If the source node is not able to use the channel selected by the destination, it cannot transmit packets to that destination in the current beacon interval. It has to wait for the next beacon interval for again negotiating channels. The ATIM packets themselves may be lost due to collisions; in order to prevent this, each node waits for a randomly chosen back-off period (between 0 and CWmin) before transmitting the ATIM packet.

Operation of the MMAC protocol is illustrated in [Figure 4.34](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig34). At the beginning of the beacon interval, source node S1 sends an ATIM message to receiver R1. Receiver R1 responds by sending an ATIM-ACK packet (ATIM-ACK(1)) carrying the ID 1 of the channel it prefers (in [Figure 4.34](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig34) the number within parentheses indicates the ID of the preferred channel). Node S1, on receiving this packet, confirms the reservation by sending an ATIM-RES packet (ATIM-RES(1)) for channel 1. The ATIM-ACK(1) packet sent by receiver R1 is also overheard by node R2. When node R2 receives an ATIM packet from source S2, it chooses a different channel with ID 2, and sends the channel information to source S2 through the ATIM-ACK packet (ATIM-ACK(2)). Since channel 2 is agreeable to node S2, it responds by sending the ATIM-RES(2) packet, and the reservation gets established. Once the ATIM window finishes, the data transmission (through RTS-CTS-DATA-ACK packet exchange) between node pairs S1-R1 and S2-R2 takes place on the corresponding reserved channels, channel 1 and channel 2, respectively.



***Figure 4.34. Operation of MMAC protocol.***

In this protocol, it is the receiver node that plays a dominant role in channel selection. In case all channels are in use at the receiver, even then the receiver selects one of the channels. Since the actual data packet transmissions are protected by the RTS/CTS control packet exchange, the nodes transmitting packets on the same channel need to contend for the channel, as in IEEE 802.11 for transmitting packets. The protocol also employs a power-saving mode. In case a node realizes after the ATIM window that it is neither going to transmit packets nor going to receive packets, then the node goes into a power-saving doze mode.

Channel selection is done at the receiver in the following manner. The receiver node uses the PCL on the received ATIM packet and its own PCL for selecting the best possible channel for communication with the source node. The channel selection procedure tries to balance the network load on the channels. If a receiver node R receives an ATIM packet from a source node S, it selects a channel as below.

• If there exists a HIGH state channel in node R's PCL, then that channel is selected.

• Else if there exists a HIGH state channel in the PCL of node S, then this channel is selected.

• Else if there exists a common MID state channel in the PCLs of both node S and node R, then that channel is selected. If many such channels exists, one of them is selected randomly.

• Else if there exists a channel which is in the MID state at only one of the two nodes, then that channel is chosen. If many such channels exist, one of them is selected randomly.

• If all channels in both PCLs are in the LOW state, the counters of the corresponding channels at nodes S and R are added, and the channel with the least count is selected. Ties are broken arbitrarily.

MMAC uses simple hardware. It requires only a single transceiver. It does not have any dedicated control channel. The throughput of MMAC is higher than that of IEEE 802.11 when the network load is high. This higher throughput is in spite of the fact that in MMAC only a single transceiver is used at each node. Unlike other protocols, the packet size in MMAC need not be increased in order to take advantage of the presence of an increased number of channels.

**4.9.2 Multichannel CSMAMAC Protocol**

In the multichannel CSMAMAC protocol (MCSMA) [[25](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry25)], the available bandwidth is divided into several channels. A node with a packet to be transmitted selects an idle channel randomly. The protocol also employs the notion of soft channel reservation, where preference is given to the channel that was used for the previous successful transmission. Though the principle used in MCSMA is similar to the frequency division multiple access (FDMA) schemes used in cellular networks, the major difference here is that there is no centralized infrastructure available, and channel assignment is done in a distributed fashion using carrier-sensing. The operation of the protocol is discussed below.

The total available bandwidth is divided into N non-overlapping channels (N is independent of the number of hosts in the network), each having a bandwidth of (W/N), where W is the total bandwidth available for communication. The channels may be created in the frequency domain (FDMA) or in the code domain (CDMA). Since global synchronization between nodes is not available in ad hoc wireless networks, channel division in the time domain (TDMA) is not used.

An idle node (which is not transmitting packets) continuously monitors all the N channels. A channel whose total received signal strength[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ftn.ch06fn01) (TRSS) is below the sensing threshold (ST) of the node is marked IDLE by the node. The time at which TRSS drops below ST is also noted for each IDLE channel. Such channels are put in the free-channels list.

[1](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fn01) The total received signal strength of a signal is calculated by the sum of contributions arising from the various individual multipath components of the signal.

When an idle node receives a packet to be transmitted, it does the following. If the free-channels list is empty, it waits for any channel to become IDLE. It then waits for an additional long inter-frame space (LIFS) time, and for another random access back-off period. If the channel remains idle for this entire wait period, then the node starts transmitting its packets on this channel. In case the free-channels list is non-empty, the node first checks whether the channel it used for its most recent successful transmission is included in the list. If so, the node uses this channel for its new transmission. Otherwise, one among the IDLE channels available in the free-channels list is randomly chosen (using a uniform random number generator).

Before the actual packet transmission, the node checks the TRSS of the chosen channel. If it had remained below ST for at least LIFS period of time, then the node immediately initiates packet transmission. Otherwise, the node initiates back-off delay after the LIFS time period. During the back-off period, if the TRSS of the chosen channel goes above ST, then the back-off is immediately canceled. A new back-off delay is scheduled when the TRSS again goes below the ST. After successfully transmitting a packet (indicated by an acknowledgment from the receiver), the sender node notes the ID of the channel used. This channel would be given preference when a new channel is to be selected for its next transmission.

When the number of channels N is sufficiently large, each node tends to reserve a channel for itself. This is because a node prefers the channel used in its last successful transmission for its next transmission also. This reduces the probability of two contending nodes choosing the same channel for transmission. Nodes are expected to dynamically select channels for transmissions in a mutually exclusive manner, so as to enable parallel interference-free transmissions. Even at high traffic loads, due to the tendency of every node to choose a reserved channel for itself, the chances of collisions are greatly reduced.

The number of channels into which the available bandwidth is split is a very important factor affecting the performance of the protocol. If the number of channels is very large, then the protocol results in very high packet transmission times.

**4.9.3 Power Control MAC Protocol for Ad Hoc Networks**

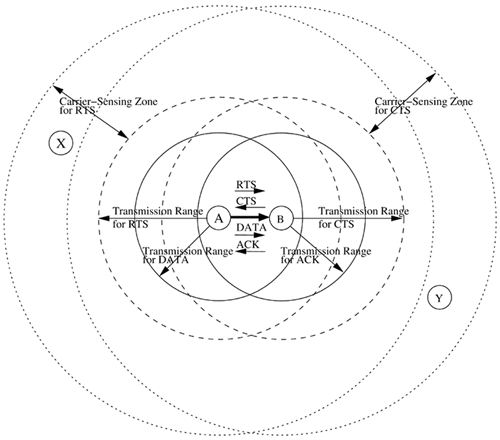
The power control MAC protocol (PCM) [[26](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry26)] allows nodes to vary their transmission power levels on a per-packet basis. The PCM protocol is based on the power control protocol used in [[27](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry27)], which is referred to as the BASIC protocol in this section. In what follows, the working of the BASIC power control protocol is briefly described. This is followed by a discussion of the PCM protocol.

In the BASIC scheme, the RTS and CTS packets are transmitted with maximum power pmax. The RTS-CTS handshake is used for deciding upon the transmission power for the subsequent DATA and ACK packet transmissions. This can be done using two methods. In the first method, source node A transmits the RTS with maximum power pmax. This RTS is received at the receiver with signal level pr. The receiver node B can calculate the minimum required transmission power level pdesiredfor the DATA packet, based on the received power level pr, the transmitted power level pmax, and the noise level at receiver B. Node B then specifies this pdesired in the CTS packet it transmits to node A. Node A transmits the DATA packet using power level pdesired. In the second method, when the receiver node B receives an RTS packet, it responds with a CTS packet at the usual maximum power level pmax. When the source node receives this CTS packet, it calculates pdesired based on the received power level pr and transmitted power level pmax as



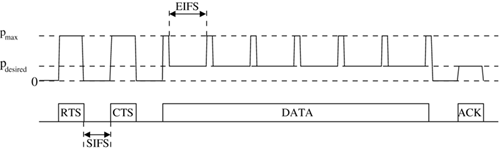
whereRxthresh is the minimum necessary received signal strength and c is a constant. The source node uses power level pdesired to transmit the DATA packet. Similarly, the receiver uses the signal power of the received RTS packet to determine the power level to be used, pdesired, for the ACK packet. This method assumes the attenuation between the sender and receiver nodes to be the same in both directions. It also assumes the noise level at the nodes to be below a certain predefined threshold value.

Thus, the BASIC scheme uses maximum transmit power for RTS and CTS packets, and only necessary power levels for the DATA and ACK packets. But this scheme has a drawback. Consider [Figure 4.35](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig35). Node A sends an RTS to node B, for which node B sends a CTS packet. Since these packets are sent at maximum power, nodes X and Y that are located in the carrier-sensing zones of nodes A and B, respectively (when a node N1 is in the carrier-sensing zone of node N2, node N1 can sense the signal from node N2, but the received signal strength is not high enough to decode it correctly), defer their transmissions for a sufficient enough period of time [extended inter-frame space (EIFS) period of time] so as to not interfere with the RTS-CTS exchange. But since the DATA and ACK transmissions use only the minimum necessary power, the DATA transmitted by node A cannot be sensed by node X, and the ACK packet transmitted by node B cannot be sensed by node Y. So if nodes X and Y transmit after the EIFS period (which is set in their NAVs on sensing the RTS or CTS packets), the packet transmitted by node X would collide at node A with the ACK packet from node B, and the packet transmitted by node Y would collide with the DATA packet at node B.



***Figure 4.35. Packet transmission in BASIC scheme.***

PCM modifies this scheme so as to minimize the probability of collisions. The source and receiver nodes transmit the RTS and CTS packets, as usual, with maximum power pmax. Nodes in the carrier-sensing zones of the source and receiver nodes set their NAVs for EIFS duration when they sense the signal but are not able to decode it. The source node generally transmits with minimum necessary power, as in the BASIC scheme. But, in order to avoid collisions with packets transmitted by the nodes in its carrier-sensing zone, the source node transmits the DATA packet at maximum power level pmax periodically. The duration of each such transmission must be larger than the time required for physical carrier-sensing. Since the nodes in the carrier-sensing zone defer their transmissions for EIFS duration if they are not able to decode the received signal, the transmit power for the DATA packet is increased (and brought down back to original level) every EIFS duration. The power level changes for RTS-CTS-DATA-ACK transmissions is depicted in [Figure 4.36](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig36). Thus this protocol prevents collisions of ACK packets at the sender node.



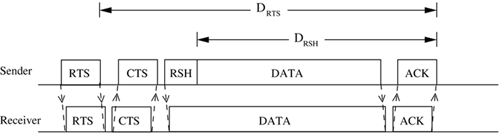
***Figure 4.36. Transmission power pattern in PCM.***

Hence with the above simple modification, the PCM protocol overcomes the problems faced in the BASIC scheme. PCM achieves throughput very close to that of the 802.11 protocol while using much less energy.

**4.9.4 Receiver-Based Autorate Protocol**

The receiver-based autorate protocol (RBAR) [[28](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry28)] uses a novel rate adaptation approach. The rate adaptation mechanism is at the receiver node instead of being located at the sender. Rate adaptation is the process of dynamically switching data rates in order to match the channel conditions so that optimum throughput for the given channel conditions is achieved. Rate adaptation consists of two processes, namely, channel quality estimation and rate selection. The accuracy of the channel quality estimates significantly influences the effectiveness of the rate adaptation process. Therefore, it is important that the best available channel quality estimates are used for rate selection. Since it is the channel quality at the receiver node which determines whether a packet can be received or not, it can be concluded that the best channel quality estimates are available at the receiver. The estimates must be used as early as possible before they get stale. If the sender is to implement the rate adaptation process, significant delay would be involved in communicating the channel quality estimates from the receiver to the sender, which may result in the estimates becoming stale before being used. Therefore, the RBAR protocol advocates for rate adaptation at the receiver node rather than at the sender.

Rate selection is done at the receiver on a per-packet basis during the RTS-CTS packet exchange. Since rate selection is done during the RTS-CTS exchange, the channel quality estimates are very close to the actual transmission times of the data packets. This improves the effectiveness of the rate selection process. The RTS and CTS packets carry the chosen modulation rate and the size of the data packet, instead of carrying the duration of the reservation. The packet transmission process is depicted in [Figure 4.37](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig37). The sender node chooses a data rate based on some heuristic and inserts the chosen data rate and the size of the data packet into the RTS. When a neighbor node receives this RTS, it calculates the duration of the reservation DRTS using the data rate and packet size carried on the RTS. The neighbor node then updates its NAV accordingly to reflect the reservation. While receiving the data packet, the receiver node generates an estimate of the channel conditions for the impending data transfer. Based on this estimate, it chooses an appropriate data rate. It stores the chosen data rate and the size of the packet on the CTS packet and transmits the CTS to the sender. Neighbor nodes receiving the CTS calculate the expected duration of the transmission and update their NAVs accordingly. The source node, on receiving the CTS packet, responds by transmitting the data packet at the rate chosen by the receiver node.



***Figure 4.37. Packet transmission in RBAR.***

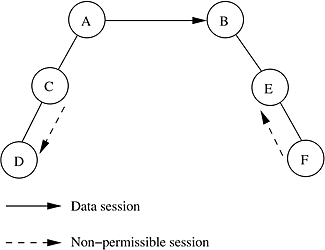
If the rates chosen by the sender and receiver are different, then the reservation duration DRTS calculated by the neighbor nodes of the sender would not be valid. DRTS time period, which is calculated based on the information carried initially by the RTS packet, is referred to as tentative reservation. In order to overcome this problem, the sender node sends the data packet with a special MAC header containing a reservation subheader (RSH). The RSH contains a subset of header fields already present in the IEEE 802.11 data frame, along with a check sequence for protecting the subheader. The fields in the RSH contain control information for determining the duration of the transmission. A neighbor node with tentative reservation entries in its NAV, on hearing the data packet, calculates DRSH, the new reservation period, and updates its NAV to account for the difference between DRTS and DRSH.

For the channel quality estimation and prediction algorithm, the receiver node uses a sample of the instantaneous received signal strength at the end of RTS reception. For the rate selection algorithm, a simple threshold-based technique is used. Here the rate is chosen by comparing the channel quality estimate [e.g., signal to noise ratio (SNR)] against a series of thresholds representing the desired performance bounds (e.g., a series of SNR thresholds). The modulation scheme with the highest data rate, satisfying the performance objective for the channel quality estimate, is chosen.

RBAR employs an efficient quality estimation mechanism, which leads to a high overall system throughput. RBAR can be easily incorporated into many existing medium access control protocols.

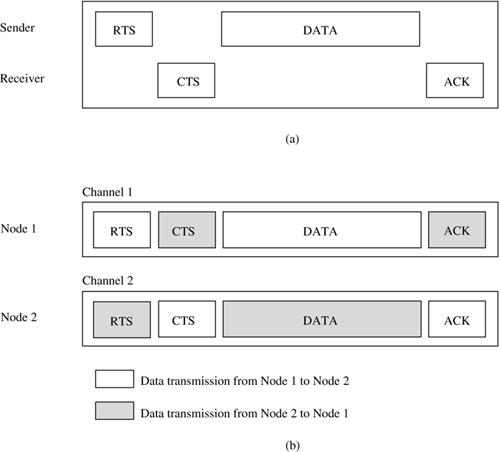
**4.9.5 Interleaved Carrier-Sense Multiple Access Protocol**

The interleaved carrier-sense multiple access protocol (ICSMA) [[29](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry29)] efficiently overcomes the exposed terminal problem faced in ad hoc wireless networks. The inability of a source node to transmit, even though its transmission may not affect other ongoing transmissions, is referred to as the exposed terminal problem. For example, consider the topology shown in [Figure 4.38](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig38). Here, when a transmission is going from node A to node B, nodes C and F would not be permitted to transmit to nodes D and E, respectively. Node C is called a sender-exposed node, and node E is called a receiver-exposed node. The exposed terminal problem reduces the bandwidth efficiency of the system.



***Figure 4.38. Exposed terminal problem.* Reproduced with permission from [**[**29**](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry29)**], © IEEE, 2003.**

In ICSMA, the total available bandwidth is split into two equal channels (say, channel 1 and channel 2). The handshaking process is interleaved between the two channels, hence the name interleaved carrier-sense multiple access. The working of ICSMA is very simple. It uses the basic RTS-CTS-DATA-ACK exchange mechanism used in IEEE 802.11 DCF. If the source node transmits the RTS packet on channel 1, the receiver node, if it is ready to accept packets from the sender, responds by transmitting the CTS packet on channel 2. Each node maintains a data structure called extended network allocation vector (E-NAV), which is analogous to the network allocation vector (NAV) used in IEEE 802.11 DCF. On receiving an RTS packet, the receiver node checks its E-NAV and finds out whether free time slots are available. It sends the CTS only if free slots are available. The source node, on receiving this CTS, transmits the DATA packet on channel 1. The receiver acknowledges the reception of the DATA packet by transmitting the ACK on channel 2. The ICSMA channel access mechanism is illustrated in [Figure 4.39](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig39). [Figure 4.39 (a)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig39) shows the RTS-CTS-DATA-ACK exchange of 802.11 DCF. [Figure 4.39 (b)](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig39) shows simultaneous data packet transmissions between two nodes in ICSMA. After transmitting an RTS or a DATA packet on a channel, the sender node waits on the other channel for the CTS or ACK packet. If it does not receive any packet on the other channel, it assumes that the RTS or DATA packet it transmitted was lost, and retries again. Similarly at the receiver, after transmitting a CTS frame on one of the channels, the receiver node waits on the other channel for the DATA packet. If the DATA packet is not received within the timeout period, it retransmits the CTS packet.



***Figure 4.39. Packet transmissions in (a) 802.11 DCF and (b) ICSMA.* Reproduced with permission from [**[**29**](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06biblio01entry29)**], © IEEE, 2003.**

The performance improvement of ICSMA is attributed to the following facts:

• Nodes that hear RTS in a particular channel (say, channel 1) and do not hear the corresponding CTS on the other channel (channel 2) conclude that they are only sender-exposed in channel 1. Therefore, if they have packets to send, they can use channel 1 to transmit RTS to other nodes. This would not have been possible in 802.11 DCF, where transmissions by a sender-exposed node would have collided with the corresponding currently active sender node.

• Nodes that hear only the CTS in a particular channel (say, channel 1) and had not heard the corresponding RTS on the other complementary channel (channel 2) realize that they are only receiver-exposed on channel 1 to the on-going transmission. If they receive any RTS on channel 2, they would not refrain from sending a CTS on channel 1 for the received RTS. This would also not have been possible in 802.11 DCF, where there would have been collision at the receiver of the on-going session between the CTS packet transmitted by this node and the DATA packets belonging to the on-going session. Also, if this CTS transmission is successful, then there might have been collisions between DATA packets belonging to the two sessions at both the receiver nodes.

The E-NAV used in ICSMA is implemented as two linked lists of blocks, namely, the SEList and the REList. Each block in each linked list has a start time and an end time. A typical list looks like s1, f1; s2, f2; ...; sk, fk where Si denotes the start time of the ith block in the list and fi denotes the finish time of the ith block in the list. The SEList is used to determine if the node would be sender-exposed at any given instant of time in the future. A node is predicted to be sender-exposed at any time t if there is a block sj, fj in the SEList such that sj< t <fj. Similarly, the REList tells if the node would be receiver-exposed at any time in the future. A node is predicted to be receiver-exposed at any time t if there exists a block sj, fj in the REList such that sj< t <fj. The SEList and the REList are updated whenever the RTS and CTS packets are received by the node. The SEList is modified when an RTS packet is received, and the REList is modified when a CTS packet is received by the node. The modification in the list might be adding a new block, modifying an existing block, or merging two or more existing blocks and modifying the resulting block.

ICSMA is a simple two-channel MAC protocol for ad hoc wireless networks that reduces the number of exposed terminals and tries to maximize the number of simultaneous sessions. ICSMA was found to perform better than the 802.11 DCF protocol is terms of metrics such as throughput and channel access delay.

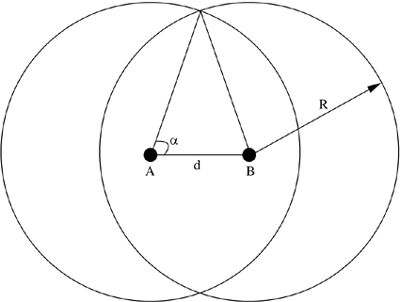
**4.10 SUMMARY**

In this chapter, the major issues involved in the design of a MAC protocol for ad hoc wireless networks were identified. The different classifications of the existing MAC protocols were provided. The chapter also gave detailed descriptions of the major MAC protocols that exist for ad hoc wireless networks.

The main issues to be considered while designing the MAC protocol are bandwidth efficiency, QoS support, time synchronization, error-prone nature of the shared broadcast transmission channel, and node mobility. The MAC protocols are classified into several types based on criteria such as availability of bandwidth reservation mechanisms, data transfer initiation approach, usage of single or multiple transmission channels, and requirement of time synchronization. Protocols falling under the above categories were explained in detail.

**4.11 PROBLEMS**

1. Consider [Figure 4.40](https://www.safaribooksonline.com/library/view/ad-hoc-wireless/013147023X/ch06.html#ch06fig40). Transmission is going on between nodes A and B that are separated by a distance d. R is the transmission range of each node. Mark in the figure the regions that could contain exposed terminals and hidden terminals. Also calculate the area of each region.



***Figure 4.40. Hidden and exposed regions.***

1. What are the advantages of reservation-based MAC protocols over contention-based MAC protocols?
2. Give application scenarios where contention-based, reservation-based, and packet scheduling-based MAC protocols can be used.
3. Compare the pros and cons of using scheduling-based MAC protocols over reservation-based MAC protocols.
4. What are the disadvantages of the binary exponential back-off mechanism used in MACA? How are they overcome in MACAW?
5. How does the packet queuing mechanism of MACA differ from that of MACAW? Which one of them is better? Why?
6. Calculate the probability of data packet collision in the MACA protocol. Assume that Tc is the control packet transmission and propagation delay, Tw is the optimal maximum back-off time, β is the percentage of ready nodes, and R is the transmission range of each node.
7. What are the disadvantages of the BTMA protocol? How are they overcome in the DBTMA protocol?
8. What is the main difference between the receiver-initiated MAC protocols MACA-BI and MARCH?
9. How is synchronization between nodes achieved in the HRMA protocol?
10. Which protocol is more bandwidth efficient, RTMAC or MACA/PR? Explain.
11. Explain the back-off calculation mechanism used in DWOP. Is it guaranteed to be accurate at all times? If not, explain why.
12. What is meant by the carrier-sensing zone of a transmission? Does it have any effect on the performance of a MAC protocol?
13. Channel quality estimation can be done both at the sender and the receiver. Which is more advantageous? Why?
14. What are the pros and cons of using multichannel MAC protocols over single-channel MAC protocols?
15. In FPRP, can a situation occur where a requesting node is not able to detect collisions that have occurred in the reservation request phase? If so, suggest simple modifications to solve the problem.
16. What are the advantages and disadvantages of MAC protocols using directional antennas?

**BIBLIOGRAPHY**

[1] P. Karn, "MACA – A New Channel Access Method for Packet Radio," *Proceedings of ARRL/CRRL Amateur Radio Computer Networking Conference 1990*, pp. 134-140, September 1990.

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proceedings of ACM SIGCOMM 1994*, pp. 212-225, August 1994.

[3] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," *Proceedings of ACM SIGCOMM 1995*, pp. 262-273, September 1995.

[4] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple Access and the Busy Tone Solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417-1433, December 1975.

[5] J. Deng and Z. J. Haas, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," *Proceedings of IEEE ICUPC 1998*, vol. 1, pp. 973-977, October 1998.

[6] C. S. Wu and V. O. K. Li, "Receiver-Initiated Busy Tone Multiple Access in Packet Radio Networks," *ACM Computer Communication Review*, vol. 17, no. 5, pp. 336- 42, August 1987.

[7] F. Talucci and M. Gerla, "MACA-BI (MACA by Invitation): A Wireless MAC Protocol for High Speed Ad Hoc Networking," *Proceedings of IEEE ICUPC 1997*, vol. 2, pp. 913-917, October 1997.

[8] C. K. Toh, V. Vassiliou, G. Guichal, and C. H. Shih, "MARCH: A Medium Access Control Protocol for Multi-Hop Wireless Ad Hoc Networks," *Proceedings of IEEE MILCOM 2000*, vol. 1, pp. 512-516, October 2000.

[9] S. Jiang, J. Rao, D. He, and C. C. Ko, "A Simple Distributed PRMA for MANETs," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 293-305, March 2002.

[10] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications," *IEEE Transactions on Communications*, vol. 37, no. 8, pp. 885-890, August 1989.

[11] Z. Tang and J. J. Garcia-Luna-Aceves, "A Protocol for Topology-Dependent Transmission Scheduling in Wireless Networks," *Proceedings of IEEE WCNC 1999*, vol. 3, no. 1, pp. 1333-1337, September 1999.

[12] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad Hoc Networks," *Proceedings of IEEE INFOCOM 1999*, vol. 1, pp. 194-201, March 1999.

[13] C. W. Ahn, C. G. Kang, and Y. Z. Cho, "Soft Reservation Multiple Access with Priority Assignment (SRMA/PA): A Novel MAC Protocol for QoS-Guaranteed Integrated Services in Mobile Ad Hoc Networks," *Proceedings of IEEE Fall VTC 2000*, vol. 2, pp. 942-947, September 2000.

[14] C. Zhu and M. S. Corson, "A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks," *ACM/Baltzer Journal of Wireless Networks*, vol. 7, no. 4, pp. 371-384, July 2001.

[15] C. R. Lin and M. Gerla, "Real-Time Support in Multi-Hop Wireless Networks," *ACM/Baltzer Journal of Wireless Networks*, vol. 5, no. 2, pp. 125-135, March 1999.

[16] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proceedings of ACM SIGCOMM 1994*, pp. 234-244, August 1994.

[17] B. S. Manoj and C. Siva Ram Murthy, "Real-Time Traffic Support for Ad Hoc Wireless Networks," *Proceedings of IEEE ICON 2002*, pp. 335-340, August 2002.

[18] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed Priority Scheduling and Medium Access in Ad Hoc Networks," *ACM/Baltzer Journal of Wireless Networks*, vol. 8, no. 5, pp. 455-466, September 2002.

[19] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis," *Proceedings of ACM MOBIHOC 2002*, pp. 58-70, June 2002.

[20] I. Karthigeyan, B. S. Manoj, and C. Siva Ram Murthy, "A Distributed Laxity-Based Priority Scheduling Scheme for Time-Sensitive Traffic in Mobile Ad Hoc Networks," to appear in *Ad Hoc Networks Journal*.

[21] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto, "A MAC Protocol for Mobile Ad Hoc Networks Using Directional Antennas," *Proceedings of IEEE WCNC 2000*, vol. 1, pp. 1214-1219, September 2000.

[22] Z. Huang, C. C. Shen, C. Srisathapornphat, and C. Jaikaeo, "A Busy Tone-Based Directional MAC Protocol for Ad Hoc Networks," *Proceedings of IEEE MILCOM 2002*, October 2002.

[23] Y. B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks," *Proceedings of IEEE INFOCOM 2000*, vol. 1, pp. 13-21, March 2000.

[24] J. So and N. H. Vaidya, "A Multi-Channel MAC Protocol for Ad Hoc Wireless Networks," *Technical Report:* [*http://www.crhc.uiuc.edu/~nhv/papers/jungmintech.ps*](http://www.crhc.uiuc.edu/~nhv/papers/jungmintech.ps), January 2003.

[25] A. Nasipuri, J. Zhuang, and S. R. Das, "A Multi-Channel CSMA MAC Protocol for Multi-Hop Wireless Networks," *Proceedings of IEEE WCNC 1999*, vol. 1, pp. 1402-1406, September 1999.

[26] E. S. Jung and N. H. Vaidya, "A Power Control MAC Protocol for Ad Hoc Networks," *Proceedings of ACM MOBICOM 2002*, pp. 36-47, September 2002.

[27] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving Transmission Power in Wireless Ad Hoc Networks," *Proceedings of ICNP 2001*, pp. 11-14, November 2001.

[28] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," *Proceedings of ACM MOBICOM 2001*, pp. 236-251, July 2001.

[29] S. Jagadeesan, B. S. Manoj, and C. Siva Ram Murthy, "Interleaved Carrier Sense Multiple Access: An Efficient MAC Protocol for Ad Hoc Wireless Networks," *Proceedings of IEEE ICC 2003*, vol. 2, pp. 11-15, May 2003.

[30] I. Karthigeyan and C. Siva Ram Murthy, "Medium Access Control in Ad Hoc Wireless Networks: A Survey of Issues and Solutions," *Technical Report*, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, March 2003.