Chapter 10

QUALITY OF SERVICE IN AD HOC WIRELESS NETWORKS

10.1 INTRODUCTION

Quality of service (QoS) is the performance level of a service offered by the network to the user. The goal of QoS provisioning is to achieve a more deterministic network behavior, so that information carried by the network can be better delivered and network resources can be better utilized. A network or a service provider can offer different kinds of services to the users. Here, a service can be characterized by a set of measurable prespecified service requirements such as minimum bandwidth, maximum delay, maximum delay variance (jitter), and maximum packet loss rate. After accepting a service request from the user, the network has to ensure that the service requirements of the user's flow are met, as per the agreement, throughout the duration of the flow (a packet stream from the source to the destination). In other words, the network has to provide a set of service guarantees while transporting a flow.

After receiving a service request from the user, the first task is to find a suitable loop-free path from the source to the destination that will have the necessary resources available to meet the QoS requirements of the desired service. This process is known as QoS routing. After finding a suitable path, a resource reservation protocol is employed to reserve necessary resources along that path. QoS guarantees can be provided only with appropriate resource reservation techniques. For example, consider the network shown in Figure 10.1. The attributes of each link are shown in a tuple $< BW, D >$, where BW and D represent available bandwidth in Mbps and delay[1] in milliseconds. Suppose a packet-flow from node B to node G requires a bandwidth guarantee of 4 Mbps. Throughout the chapter, the terms "node" and "station" are used interchangeably. QoS routing searches for a path that has sufficient bandwidth to meet the bandwidth requirement of the flow. Here, six paths are available between nodes B and G as shown in Table 10.1. QoS routing selects

[1] Delay includes transmission delay, propagation delay, and queuing delay.

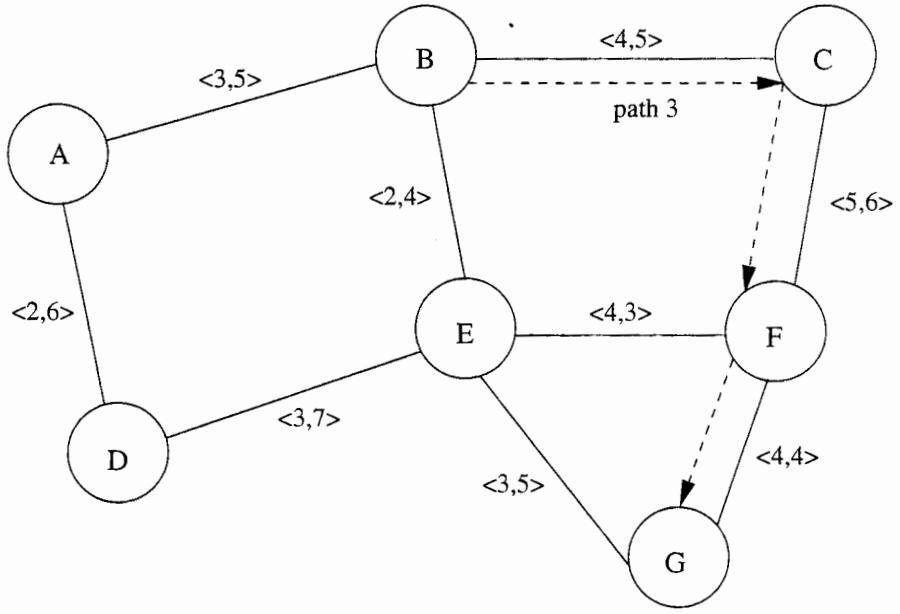


Figure 10.1. An example of QoS routing in ad hoc wireless network.

path 3 (*i.e.*, $B \rightarrow C \rightarrow F \rightarrow G$) because, out of the available paths, path 3 alone meets the bandwidth constraint of 4 Mbps for the flow. The end-to-end bandwidth of a path is equal to the bandwidth of the bottleneck link (*i.e.*, the link having minimum bandwidth among all the links of a path). The end-to-end delay of a path is equal to the sum of delays of all the links of a path. Clearly, path 3 is not optimal in terms of hop count and/or end-to-end delay parameters, while path 1 is optimal in terms of both hop count and end-to-end delay parameters. Hence, QoS routing has to select a suitable path that meets the QoS constraints specified in the service request made by the user. QoS routing has been described in detail in Section 10.5.1.

Table 10.1. Available paths from node B to node G

No.	Path	Hop Count	End-to-end Bandwidth (Mbps)	End-to-end Delay (milliseconds)
1	$B \rightarrow E \rightarrow G$	2	2	9
2	$B \rightarrow E \rightarrow F \rightarrow G$	3	2	11
3	$B \rightarrow C \rightarrow F \rightarrow G$	3	4	15
4	$B \rightarrow C \rightarrow F \rightarrow E \rightarrow G$	4	3	19
5	$B \rightarrow A \rightarrow D \rightarrow E \rightarrow G$	4	2	23
6	$B \rightarrow A \rightarrow D \rightarrow E \rightarrow F \rightarrow G$	5	2	25

QoS provisioning often requires negotiation between host and network, call admission control, resource reservation, and priority scheduling of packets. QoS can be rendered in ad hoc wireless networks through several ways, namely, per flow, per link, or per node. In ad hoc wireless networks, the boundary between the service provider (network) and the user (host) is not defined clearly, thus making it essential to have better coordination among the hosts to achieve QoS. Characteristics of ad hoc wireless networks such as lack of central coordination, mobility of hosts, and limited availability of resources make QoS provisioning very challenging.

10.1.1 Real-Time Traffic Support in Ad Hoc Wireless Networks

Real-time applications require mechanisms that guarantee bounded delay and delay jitter. The end-to-end delay in packet delivery includes the queuing delay at the source and intermediate nodes, the processing time at the intermediate nodes, and the propagation duration over multiple hops from the source node to the destination node. Real-time applications can be classified as hard real-time applications and soft real-time applications. A hard real-time application requires strict QoS guarantees. Some of the hard real-time applications include nuclear reactor control systems, air traffic control systems, and missile control systems. In these applications, failure to meet the required delay constraints may lead to disastrous results. On the other hand, soft real-time applications can tolerate degradation in the guaranteed QoS to a certain extent. Some of the soft real-time applications are voice telephony, video-on-demand, and video conferencing. In these applications, the loss of data and variation in delay and delay jitter may degrade the service but do not produce hazardous results. Providing hard real-time guarantees in ad hoc wireless networks is extremely difficult due to reasons such as the unrestricted mobility of nodes, dynamically varying network topology, time-varying channel capacity, and the presence of hidden terminals. The research community is currently focusing on providing QoS support for applications that require soft real-time guarantees.

10.1.2 QoS Parameters in Ad Hoc Wireless Networks

As different applications have different requirements, the services required by them and the associated QoS parameters differ from application to application. For example, in case of multimedia applications, bandwidth, delay jitter, and delay are the key QoS parameters, whereas military applications have stringent security requirements. For applications such as emergency search-and-rescue operations, availability of the network is the key QoS parameter. Applications such as group communication in a conference hall require that the transmissions among nodes consume as little energy as possible. Hence, battery life is the key QoS parameter here.

Unlike traditional wired networks, where the QoS parameters are mainly characterized by the requirements of multimedia traffic, in ad hoc wireless networks the QoS requirements are more influenced by the resource constraints of the nodes. Some of the resource constraints are battery charge, processing power, and buffer space.

10.2 ISSUES AND CHALLENGES IN PROVIDING QOS IN AD HOC WIRELESS NETWORKS

Providing QoS support in ad hoc wireless networks is an active research area. Ad hoc wireless networks have certain unique characteristics that pose several difficulties in provisioning QoS. Some of the characteristics are dynamically varying network topology, lack of precise state information, lack of a central controller, error-prone shared radio channel, limited resource availability, hidden terminal problem, and insecure medium. A detailed discussion on how each of the above-mentioned characteristics affects QoS provisioning in ad hoc wireless networks is given below.

- **Dynamically varying network topology:** Since the nodes in an ad hoc wireless network do not have any restriction on mobility, the network topology changes dynamically. Hence, the admitted QoS sessions may suffer due to frequent path breaks, thereby requiring such sessions to be reestablished over new paths. The delay incurred in reestablishing a QoS session may cause some of the packets belonging to that session to miss their delay targets/deadlines, which is not acceptable for applications that have stringent QoS requirements.

- **Imprecise state information:** In most cases, the nodes in an ad hoc wireless network maintain both the link-specific state information and flow-specific state information. The link-specific state information includes bandwidth, delay, delay jitter, loss rate, error rate, stability, cost, and distance values for each link. The flow-specific information includes session ID, source address, destination address, and QoS requirements of the flow (such as maximum bandwidth requirement, minimum bandwidth requirement, maximum delay, and maximum delay jitter). The state information is inherently imprecise due to dynamic changes in network topology and channel characteristics. Hence, routing decisions may not be accurate, resulting in some of the real-time packets missing their deadlines.

- **Lack of central coordination:** Unlike wireless LANs and cellular networks, ad hoc wireless networks do not have central controllers to coordinate the activity of nodes. This further complicates QoS provisioning in ad hoc wireless networks.

- **Error-prone shared radio channel:** The radio channel is a broadcast medium by nature. During propagation through the wireless medium, the radio waves suffer from several impairments such as attenuation, multipath propagation, and interference (from other wireless devices operating in the vicinity) as discussed in Chapter 1.

- **Hidden terminal problem:** The hidden terminal problem is inherent in ad hoc wireless networks. This problem occurs when packets originating from two or more sender nodes, which are not within the direct transmission range of each other, collide at a common receiver node. It necessitates the retransmission of the packets, which may not be acceptable for flows that

have stringent QoS requirements. The RTS/CTS control packet exchange mechanism, proposed in [1] and adopted later in the IEEE 802.11 standard [2], reduces the hidden terminal problem only to a certain extent. BTMA and DBTMA provide two important solutions for this problem, which are described in Chapter 6.

- **Limited resource availability:** Resources such as bandwidth, battery life, storage space, and processing capability are limited in ad hoc wireless networks. Out of these, bandwidth and battery life are critical resources, the availability of which significantly affects the performance of the QoS provisioning mechanism. Hence, efficient resource management mechanisms are required for optimal utilization of these scarce resources.

- **Insecure medium:** Due to the broadcast nature of the wireless medium, communication through a wireless channel is highly insecure. Therefore, security is an important issue in ad hoc wireless networks, especially for military and tactical applications. Ad hoc wireless networks are susceptible to attacks such as eavesdropping, spoofing, denial of service, message distortion, and impersonation. Without sophisticated security mechanisms, it is very difficult to provide secure communication guarantees.

Some of the design choices for providing QoS support are described below.

- **Hard state versus soft state resource reservation:** QoS resource reservation is one of the very important components of any QoS framework (a QoS framework is a complete system that provides required/promised services to each user or application). It is responsible for reserving resources at all intermediate nodes along the path from the source to the destination, as requested by the QoS session. QoS resource reservation mechanisms can be broadly classified into two categories: *hard state* and *soft state* reservation mechanisms. In hard state resource reservation schemes, resources are reserved at all intermediate nodes along the path from the source to the destination throughout the duration of the QoS session. If such a path is broken due to network dynamics, these reserved resources have to be explicitly released by a deallocation mechanism. Such a mechanism not only introduces additional control overhead, but may also fail to release resources completely in case a node previously belonging to the session becomes unreachable. Due to these problems, soft state resource reservation mechanisms, which maintain reservations only for small time intervals, are used. These reservations get refreshed if packets belonging to the same flow are received before the timeout period. The soft state reservation timeout period can be equal to packet inter-arrival time or a multiple of the packet inter-arrival time. If no data packets are received for the specified time interval, the resources are deallocated in a decentralized manner without incurring any additional control overhead. Thus no explicit teardown is required for a flow. The hard state schemes reserve resources explicitly and hence, at high network loads, the call blocking ratio will be

high, whereas soft state schemes provide high call acceptance at a gracefully degraded fashion.

- **Stateful versus stateless approach:** In the stateful approach, each node maintains either *global state* information or only *local state* information, while in the case of a stateless approach, no such information is maintained at the nodes. State information includes both the topology information and the flow-specific information. If global state information is available, the source node can use a centralized routing algorithm to route packets to the destination. The performance of the routing protocol depends on the accuracy of the global state information maintained at the nodes. Significant control overhead is incurred in gathering and maintaining global state information. On the other hand, if mobile nodes maintain only local state information (which is more accurate), distributed routing algorithms can be used. Even though control overhead incurred in maintaining local state information is low, care must be taken to obtain loop-free routes. In the case of the stateless approach, neither flow-specific nor link-specific state information is maintained at the nodes. Though the stateless approach solves the scalability problem permanently and reduces the burden (storage and computation) on nodes, providing QoS guarantees becomes extremely difficult.

- **Hard QoS versus soft QoS approach:** The QoS provisioning approaches can be broadly classified into two categories: *hard QoS* and *soft QoS* approaches. If QoS requirements of a connection are guaranteed to be met for the whole duration of the session, the QoS approach is termed a hard QoS approach. If the QoS requirements are not guaranteed for the entire session, the QoS approach is termed a soft QoS approach. Keeping network dynamics of ad hoc wireless networks in mind, it is very difficult to provide hard QoS guarantees to user applications. Thus, QoS guarantees can be given only within certain statistical bounds. Almost all QoS approaches available in the literature provide only soft QoS guarantees.

10.3 CLASSIFICATIONS OF QOS SOLUTIONS

The QoS solutions can be classified in two ways. One classification is based on the QoS approach employed, while the other one classifies QoS solutions based on the layer at which they operate in the network protocol stack.

10.3.1 Classifications of QoS Approaches

As shown in Figure 10.2, several criteria are used for classifying QoS approaches. The QoS approaches can be classified based on the interaction between the routing protocol and the QoS provisioning mechanism, based on the interaction between the network and the MAC layers, or based on the routing information update mechanism. Based on the interaction between the routing protocol and the QoS provisioning mechanism, QoS approaches can be classified into two categories: *coupled* and

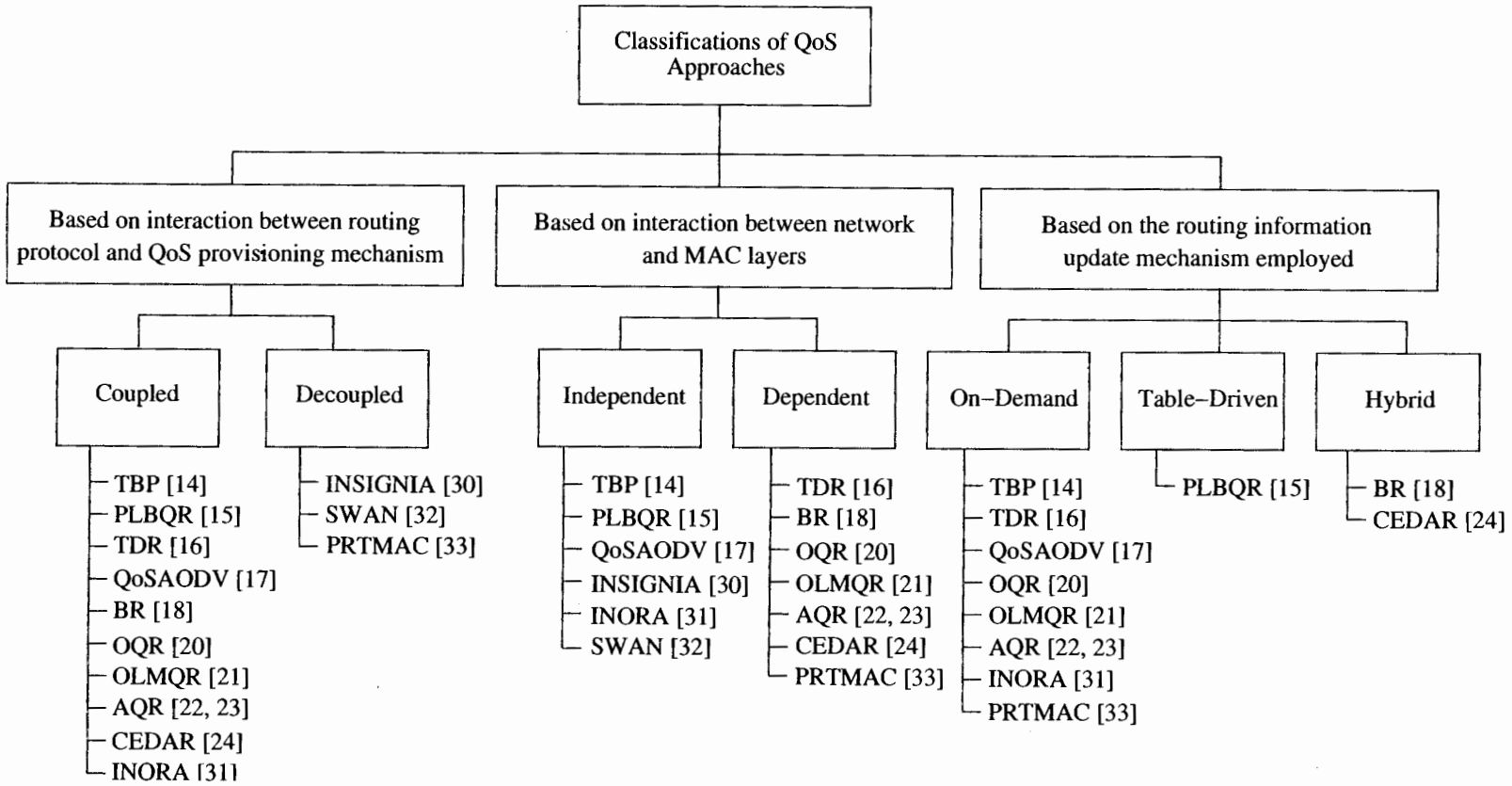


Figure 10.2. Classifications of QoS approaches.

decoupled QoS approaches. In the case of the coupled QoS approach, the routing protocol and the QoS provisioning mechanism closely interact with each other for delivering QoS guarantees. If the routing protocol changes, it may fail to ensure QoS guarantees. But in the case of the decoupled approach, the QoS provisioning mechanism does not depend on any specific routing protocol to ensure QoS guarantees.

Similarly, based on the interaction between the routing protocol and the MAC protocol, QoS approaches can be classified into two categories: *independent* and *dependent* QoS approaches. In the independent QoS approach, the network layer is not dependent on the MAC layer for QoS provisioning. The dependent QoS approach requires the MAC layer to assist the routing protocol for QoS provisioning. Finally, based on the routing information update mechanism employed, QoS approaches can be classified into three categories, namely, *table-driven*, *on-demand*, and *hybrid* QoS approaches. In the table-driven approach, each node in the network maintains a routing table which aids in forwarding packets. In the on-demand approach, no such tables are maintained at the nodes, and hence the source node has to discover the route on the fly. The hybrid approach incorporates features of both the table-driven and the on-demand approaches.

10.3.2 Layer-Wise Classification of Existing QoS Solutions

The existing QoS solutions can also be classified based on which layer in the network protocol stack they operate in. Figure 10.3 gives a layer-wise classification of QoS solutions. The figure also shows some of the cross-layer QoS solutions proposed for ad hoc wireless networks. The following sections describe the various QoS solutions listed in Figure 10.3.

10.4 MAC LAYER SOLUTIONS

The MAC protocol determines which node should transmit next on the broadcast channel when several nodes are competing for transmission on that channel. The existing MAC protocols for ad hoc wireless networks use channel sensing and random back-off schemes, making them suitable for best-effort data traffic. Real-time traffic (such as voice and video) requires bandwidth guarantees. Supporting real-time traffic in these networks is a very challenging task.

In most cases, ad hoc wireless networks share a common radio channel operating in the ISM band[2] or in military bands. The most widely deployed medium access technology is the IEEE 802.11 standard [2]. The 802.11 standard has two modes of operation: a distributed coordination function (DCF) mode and a point coordination function (PCF) mode. The DCF mode provides best-effort service, while the PCF mode has been designed to provide real-time traffic support in infrastructure-based wireless network configurations. Due to lack of fixed infrastructure support, the PCF mode of operation is ruled out in ad hoc wireless networks. Currently,

[2]ISM refers to the industrial, scientific, and medical band. The frequencies in this band (from 2.4 GHz to 2.4835 GHz) are unlicensed.

the IEEE 802.11 Task Group e (TGe) is enhancing the legacy 802.11 standard to support real-time traffic. The upcoming 802.11e standard has two other modes of operation, namely, enhanced DCF (EDCF) and hybrid coordination function (HCF) to support QoS in both infrastructure-based and infrastructure-less network configurations. These two modes of operation are discussed later in this section. In addition to these standardized MAC protocols, several other MAC protocols that provide QoS support for applications in ad hoc wireless networks have been proposed. Some of these protocols are described below.

10.4.1 Cluster TDMA

Gerla and Tsai proposed Cluster TDMA [3] for supporting real-time traffic in ad hoc wireless networks. In bandwidth-constrained ad hoc wireless networks, the limited resources available need to be managed efficiently. To achieve this goal, a dynamic clustering scheme is used in Cluster TDMA. In this clustering approach, nodes are split into different groups. Each group has a cluster-head (elected by members of that group), which acts as a regional broadcast node and as a local coordinator to enhance the channel throughput. Every node within a cluster is one hop away from the cluster-head. The formation of clusters and selection of cluster-heads are done in a distributed manner. Clustering algorithms split the nodes into clusters so that they are interconnected and cover all the nodes. Three such algorithms used are lowest-ID algorithm, highest-degree (degree refers to the number of neighbors which are within transmission range of a node) algorithm, and least cluster change (LCC) algorithm. In the lowest-ID algorithm, a node becomes a cluster-head if it has the lowest ID among all its neighbors. In the highest-degree algorithm, a node with a

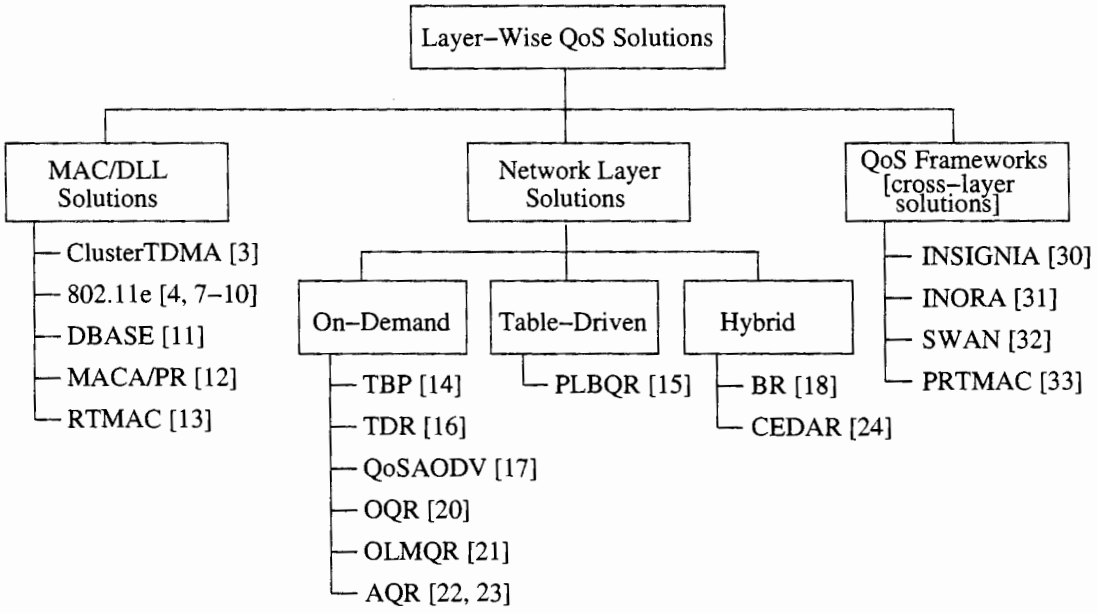


Figure 10.3. Layer-wise classification of QoS solutions.

degree greater than the degrees of all its neighbors becomes the cluster-head. In the LCC algorithm, cluster-head change occurs only if a change in network causes two cluster-heads to come into one cluster or one of the nodes moves out of the range of all the cluster-heads.

The time division multiple access (TDMA) scheme is used within a cluster for controlling access to the channel. Further, it is possible for multiple sessions to share a given TDMA slot via code division multiple access (CDMA). Across clusters, either spatial reuse of the time-slots or different spreading codes can be used to reduce the effect of inter-cluster interference. A synchronous time division frame is defined to support TDMA access within a cluster and to exchange control information. Each synchronous time division frame is divided into slots. Slots and frames are synchronized throughout the network. A frame is split into a control phase and a data phase. In the control phase, control functions such as frame and slot synchronization, routing, clustering, power management, code assignment, and virtual circuit (VC) setup are done.

The cluster-head does the reservation for the VC by assigning the slot(s) and code(s) to be used for that connection. The number of slots per frame to be assigned to a VC is determined by the bandwidth requirement of the VC. Each station broadcasts the routing information it has, the ID of its cluster-head, the power gain[3] list (the power gain list consists of the power gain values corresponding to each of the single-hop neighbors of the node concerned) it maintains, reservation status of the slots present in its data phase, and ACKs for frames that are received in the last data phase. Upon receiving this information, a node updates its routing table, calculates power gains for its neighbors, updates the power gain matrix, selects its cluster-head, records the slot reservation status of its neighbors, obtains ACKs for frames that are transmitted in the last data phase, and reserves slot(s). In each cluster, the corresponding cluster-head maintains a power gain matrix. The power gain matrix contains the power gain lists of all the nodes that belong to a particular cluster. It is useful for controlling the transmission power and the code division within a cluster.

The data phase supports both real-time and best-effort traffic. Based on the bandwidth requirement of the real-time session, a VC is set up by allocating sufficient number of slots in the data phase. The remaining data slots (*i.e.,* free slots) can be used by the best-effort traffic using the slotted-ALOHA scheme. For each node, a predefined slot is assigned in the control phase to broadcast its control information. The control information is transmitted over a common code throughout the network. At the end of the control phase, each node would have learned from the information broadcast by the cluster-head, the slot reservation status of the data phase and the power gain lists of all its neighbors. This information helps a node to schedule free slots, verify the failure of reserved slots, and drop expired real-time packets. A fast reservation scheme is used in which a reservation is made when the first packet is transmitted, and the same slots in the subsequent frames can be used for the same connection. If the reserved slots remain idle for a certain timeout period, then they are released.

[3]Power gain is the power propagation loss from the transmitter to the receiver.

10.4.2 IEEE 802.11e

In this section, the IEEE 802.11 MAC protocol is first described. Then, the recently proposed mechanisms for QoS support, namely, enhanced distributed coordination function (EDCF) and hybrid coordination function (HCF), defined in the IEEE 802.11e draft, are discussed.

IEEE 802.11 MAC Protocol

The 802.11 MAC protocol [2], which is discussed in Chapter 2, describes how a station present in a WLAN should access the broadcast channel for transmitting data to other stations. It supports two modes of operation, namely, distributed coordination function (DCF) and point coordination function (PCF). The DCF mode does not use any kind of centralized control, while the PCF mode requires an access point (AP, *i.e.*, central controller) to coordinate the activity of all nodes in its coverage area. All implementations of the 802.11 standard for WLANs must provide the DCF mode of operation, while the PCF mode of operation is optional.

The time interval between the transmission of two consecutive frames is called the inter-frame space (IFS). There are four IFSs defined in the IEEE 802.11 standard, namely, short IFS (SIFS), PCF IFS (PIFS), DCF IFS (DIFS), and extended IFS (EIFS). The relationship among them is as follows:

$$SIFS < PIFS < DIFS < EIFS$$

Distributed Coordination Function

In the DCF mode, all stations are allowed to contend for the shared medium simultaneously. CSMA/CA mechanism and random back-off scheme are used to reduce frame collisions. Each unicast frame is acknowledged immediately after being received. If the acknowledgment is not received within the timeout period, the data frame is retransmitted. Broadcast frames do not require acknowledgments from the receiving stations.

If a station A wants to transmit data to station B, station A listens to the channel. If the channel is busy, it waits until the channel becomes idle. After detecting the idle channel, station A further waits for a DIFS period and invokes a back-off procedure. The back-off time is given by

$$\text{Back} - \text{off Time} = rand(0, CW) \times slottime$$

where *slottime* includes the time needed for a station to detect a frame, the propagation delay, the time needed to switch from the receiving state to the transmitting state, and the time to signal to the MAC layer the state of the channel. The function $rand(0, CW)$ returns a pseudo-random integer from a uniform distribution over an interval [0, CW]. The current value of the contention window (CW) plays an important role in determining the back-off period of the station. The initial value of CW is CW_{min}. If a collision occurs, the value of CW is doubled. As the number of collisions increases, the value of CW is increased exponentially in order to reduce the chance of collision occurrence. The maximum value of CW is CW_{max}. The

values of CW_{min} and CW_{max} specified by the IEEE 802.11 standard are presented in Chapter 2.

After detecting the channel as being idle for a DIFS period, station A starts decrementing the back-off counter. If it senses the channel as busy during this count-down process, it suspends the back-off counter till it again detects the channel as being idle for a DIFS period. Station A then continues the count-down process, where it suspended the back-off counter. Once the back-off counter reaches zero, station A transmits a request-to-send (RTS) frame and waits for a clear-to-send (CTS) frame from the receiver B. If other stations do not cause any interference, station B acknowledges the RTS frame by sending a CTS frame. Upon receiving the CTS frame, station A transmits its data frame, the reception of which is acknowledged by receiver B by sending an ACK frame. In the above scenario, if another station C apart from station A also senses the channel as being idle (*i.e.*, stations A and C sense the channel as being idle and the back-off counters set by them expire at the same time) and transmits an RTS frame, a collision occurs and both the stations initiate back-off procedures.

If the size of the MAC frame, that is, MAC service data unit (MSDU),[4] is greater than the fragmentation threshold, it is fragmented into smaller frames, that is, MAC protocol data units (MPDUs),[5] before transmission, and each MPDU has to be acknowledged separately. Once an MSDU is transmitted successfully, CW is reset to CW_{min}. The RTS/CTS control frame exchange helps in reducing the hidden terminal problem inherent in CSMA-based ad hoc wireless networks.

Point Coordination Function

The IEEE 802.11 standard incorporates an optional access method known as PCF to let stations have priority access to the wireless medium. This access method uses a point coordinator (PC), which operates at an AP. Hence PCF is usable only in infrastructure-based network configurations. A station which requires the PCF mode of operation sends an association message to the PC to register in its polling list and gets an association identifier (AID). The PC polls the stations registered in its polling list in ascending order of AIDs to allow them contention-free access to the medium. The role of the PC is to determine which station should gain access to the channel. The stations requesting the PCF mode of operation get associated with the PC during the contention period (CP). With PCF, the channel access alternates between the contention-free period (CFP) and the contention period (CP) for the PCF and DCF modes of operation, respectively.

A CFP and the following CP form a superframe. The PC generates a beacon frame at regular beacon frame intervals called target beacon transmission time (TBTT). The value of TBTT is announced in the beacon frame. Each superframe starts with a beacon frame, which is used to maintain synchronization among local timers in the stations and to deliver protocol-related parameters. The PC uses contention-free poll (*CF-Poll*) packets to ask stations to transmit their frames. A

[4]MSDU is the information that is delivered as a unit between MAC service access points.

[5]MPDU is the unit of data exchanged between two peer MAC entities using the services of the physical layer.

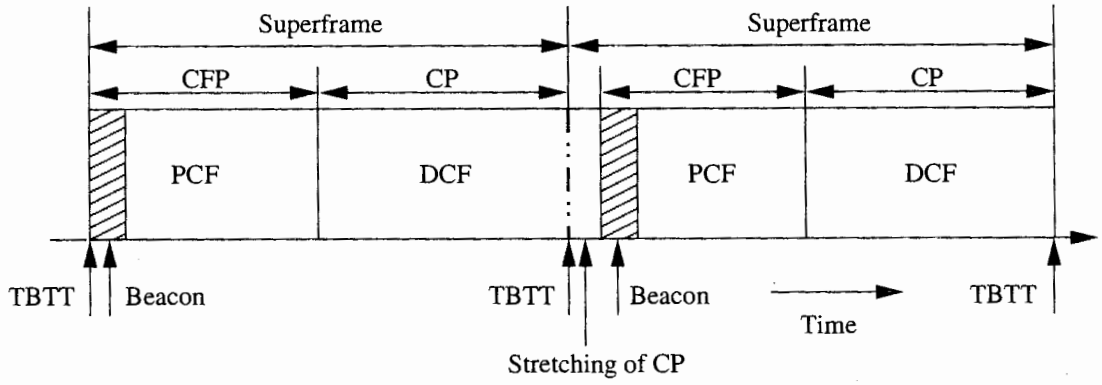


Figure 10.4. PCF and DCF frame sharing.

station that is able to respond to *CF-Poll* frames is said to be *CF-Pollable*. It is optional for a *CF-Pollable* station to respond to a *CF-Poll* frame received from the PC. If the PC receives no response from the polled station for a PIFS period, it polls the next station in the polling list (in case the remaining duration of CFP is long enough for at least one CFP transmission) or ends the CFP by transmitting *CF-End* control frame. The PC and the *CF-Pollable* stations do not use the RTS/CTS control frame exchange in the CFP. Figure 10.4 shows the operation of the network in the combined PCF and DCF modes. The channel access switches alternately between the PCF mode and the DCF mode, but the CFP may shrink due to stretching when DCF takes more time than expected. This happens when an MSDU is fragmented into several MPDUs, hence giving priority to these fragments over the PCF mode of operation.

PCF has certain shortcomings which make it unsuitable for supporting real-time traffic [4]. At TBTT, the PC has to sense the medium idle for at least PIFS before transmitting the beacon frame. If the medium is busy around TBTT, the beacon is delayed, thereby delaying the transmission of real-time traffic that has to be delivered in the following CFP. Further, polled stations' transmission durations are unknown to the PC. The MAC frame (*i.e.,* MSDU) of the polled station may have to be fragmented and may be of arbitrary length. Further, the transmission time of an MSDU is not under the control of the PC because of different modulation and coding schemes specified in the IEEE 802.11 standard. PCF is not scalable to support real-time traffic for a large number of users, as discussed in [5] and [6]. Due to these reasons, several mechanisms have been proposed to enhance the IEEE 802.11 standard to provide QoS support. The QoS mechanisms that are proposed as part of the IEEE 802.11e standard are described below.

QoS Support Mechanisms of IEEE 802.11e

The IEEE 802.11 WLAN standard supports only best-effort service. The IEEE 802.11 Task Group e (TGe) has been set up to enhance the current 802.11 MAC protocol so that it is able to support multimedia applications. The TGe has chosen the virtual DCF (VDCF) [7] proposal as the enhanced DCF (EDCF) access

mechanism. EDCF supports real-time traffic by providing differentiated DCF access to the wireless medium. The TGe has also specified a hybrid coordination function (HCF) [8] that combines EDCF with the features of PCF to simplify the QoS provisioning. HCF operates during both the CFP and the CP.

Enhanced Distributed Coordination Function

Enhanced distributed coordination function (EDCF) [7] provides differentiated and distributed access to the wireless medium. Each frame from the higher layer carries its user priority (UP). After receiving each frame, the MAC layer maps it into an access category (AC). Each AC has a different priority of access to the wireless medium. One or more UPs can be assigned to each AC. EDCF channel access has up to eight ACs [9], to support UPs. EDCF supports eight UPs. Similar to the DCF, each AC has a set of access parameters, such as CW_{min}, CW_{max}, $AIFS$, and transmission opportunity (TXOP) limit, which would be described later in this section. Hence, each AC is an enhanced variant of the DCF. Flows that fall under the same AC are effectively given identical priority to access the channel. A station accesses the channel based on the AC of the frame to be transmitted. An access point that provides QoS is called QoS access point (QAP). Each QAP will provide at least four ACs. Each station contends for transmission opportunities (TXOPs) using a set of EDCF channel access parameters that are unique to the AC of the packet to be transmitted. The TXOP is defined as an interval of time during which a station has the right to initiate transmissions. It is characterized by a starting time and a maximum duration called TXOPLimit. Depending on the duration of TXOP, a station may transmit one or more MSDUs. Priority of an AC refers to the lowest UP assigned to that AC.

During CP, each AC (of priority i) of the station contends for a TXOP and independently starts a back-off counter after detecting the channel being idle for an arbitration inter-frame space ($AIFS[i]$) as specified in [10]. $AIFS[i]$ is set as given below.

$$AIFS[i] = SIFS + AIFSN[i] \times slottime$$

where $AIFSN[i]$ is the *AIFS slot count* (*i.e.,* the number of time-slots a station has to sense the channel as idle before initiating the back-off process) for priority class i and takes values greater than zero. For high-priority classes, low $AIFSN$ values are assigned to give higher priorities for them. After waiting for $AIFS[i]$, each back-off counter is set to a random integer drawn from the range:

$$[1,\ CW[i] + 1]\ for\ each\ class\ i\ with\ AIFSN[i] = 1;$$

$$[0,\ CW[i]]\ for\ other\ classes\ i\ with\ AIFSN[i] > 1.$$

The reason for having a different range for classes with $AIFSN[i] = 1$ is to avoid transmissions initiated by stations that are operating in the EDCF mode from colliding with the hybrid coordinator's (HC, which is explained later in this section) poll packets. The HC operates at QAP and controls QoS basic service set (QBSS)

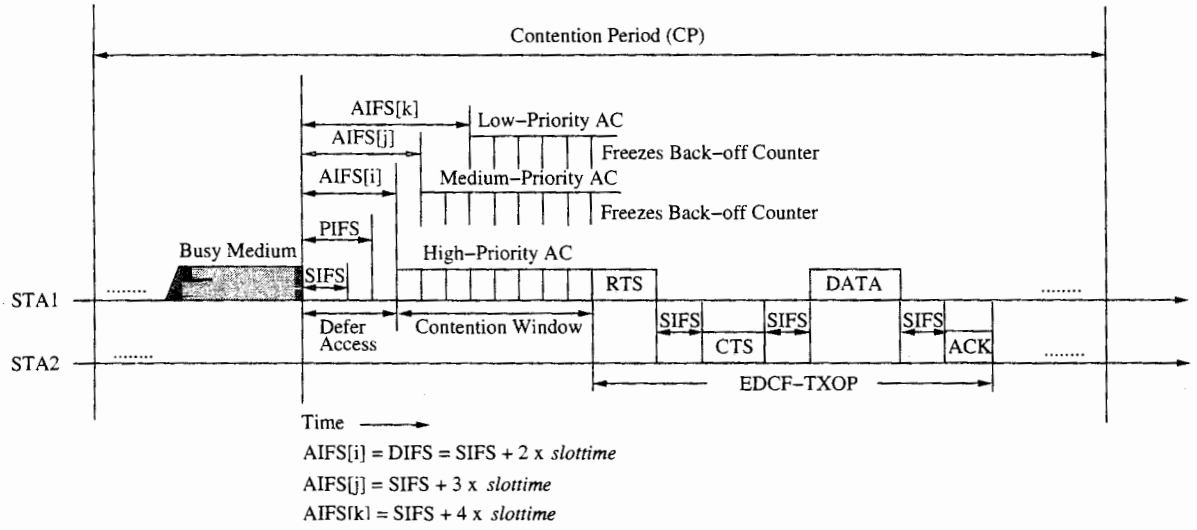


Figure 10.5. An example of EDCF access mechanism.

operation under the HCF. Figure 10.5 illustrates the relationship between SIFS, PIFS, DIFS, and various AIFS values. As in legacy DCF, if a station detects the channel to be busy before the back-off counter reaches zero, the back-off counter is suspended. The station has to wait for the channel to become idle again for an AIFS period, before continuing to decrement the counter. In this figure, it is assumed that station $STA1$ has traffic that belongs to three different ACs. The back-off counter of the highest priority AC expires first, which causes the corresponding AC to seize an EDCF-TXOP for initiating data transmission. The other ACs suspend their back-off counters and wait for the channel to become idle again. When the back-off counter of a particular AC reaches zero, the corresponding station initiates a TXOP and transmits frame(s) that have the highest priority. TXOPs are allocated via contention (EDCF-TXOP) or granted through HCF (polled-TXOP) [4]. The duration of EDCF-TXOP is limited by a QBSS-wide TXOPLimit transmitted in beacons by the HC, while during the CFP the starting time and maximum duration of each polled-TXOP is specified in the corresponding *CF-Poll* frame by the HC. If the back-off counters of two or more ACs in a single station reach zero at the same time, a scheduler inside the station avoids the *virtual collision* by granting the TXOP to the highest priority AC, while low-priority ACs behave as if there was an external collision on the wireless medium.

Hybrid Coordination Function

The hybrid coordination function (HCF) [8] combines features of EDCF and PCF to provide the capability of selectively handling MAC service data units (MSDUs), in a manner that has upward compatibility with both DCF and PCF. It uses a common set of frame exchange sequences during both the CP and the CFP. The HCF is usable only in infrastructure-based BSSs that provide QoS, that is, QBSSs. The HCF uses a QoS-aware point coordinator, called HC, which is typically col-

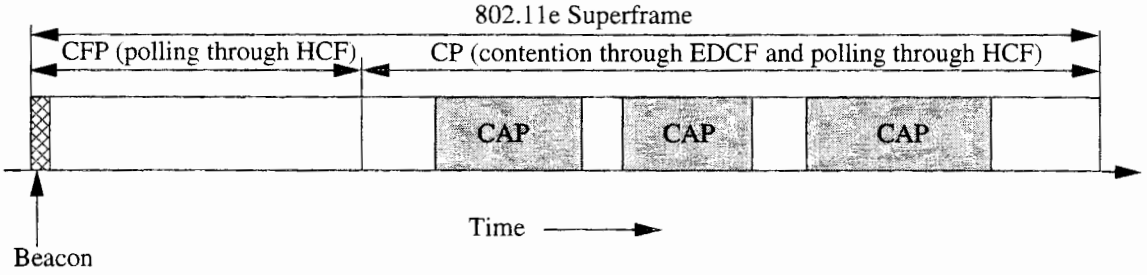


Figure 10.6. Division of time into CFP, CP, and CAP intervals.

located with a QAP. The HC implements the frame exchange sequences and the MSDU handling rules defined in HCF, operating during both the CP and the CFP. It allocates TXOPs to stations and initiates controlled contention periods for the stations to send reservation requests. When the HC needs access to the wireless medium, it senses the medium. If the medium remains idle for a PIFS period, it initiates MSDU deliveries. The HC can start contention-free controlled access periods (CAPs) at any time during a CP, after the medium is determined to be idle for at least one PIFS period.

A CAP may include one or more TXOPs. During the CAP, the HC may transmit frames and issue polls to stations which grant them TXOPs. At the end of the TXOP or when the station has no more frames to transmit, it explicitly hands over control of the medium back to the HC. Figure 10.6 shows an example of a superframe divided into CFP, CP, and three CAP intervals. During CP, each TXOP begins either when the medium is determined to be available under the EDCF rules (EDCF-TXOP) or when the station receives a QoS *CF-Poll* frame from the HC (polled-TXOP).

Figure 10.7 illustrates CFP in the HCF mode of operation. During CFP, the HC grants TXOPs to stations by sending QoS *CF-Poll* frames. The polled station can transmit one or more MSDUs in the allocated TXOP. If the size of an MSDU is too large, it can be divided into two or more fragments and transmitted sequentially with SIFS waiting periods in between them. These fragments have to be acknowledged individually. The CFP ends after the time announced in the beacon frame or by a *CF-End* frame from the HC.

10.4.3 DBASE

The distributed bandwidth allocation/sharing/extension (DBASE) protocol [11] supports multimedia traffic [both variable bit rate (VBR) and constant bit rate (CBR)] over ad hoc WLANs. In an ad hoc WLAN, there is no fixed infrastructure (*i.e.,* AP) to coordinate the activity of individual stations. The stations are part of a single-hop wireless network and contend for the broadcast channel in a distributed manner. For real-time traffic (*rt-traffic*), a contention-based process is used in order to gain access to the channel. Once a station gains channel access, a reservation-based process is used to transmit the subsequent frames. The non-real-

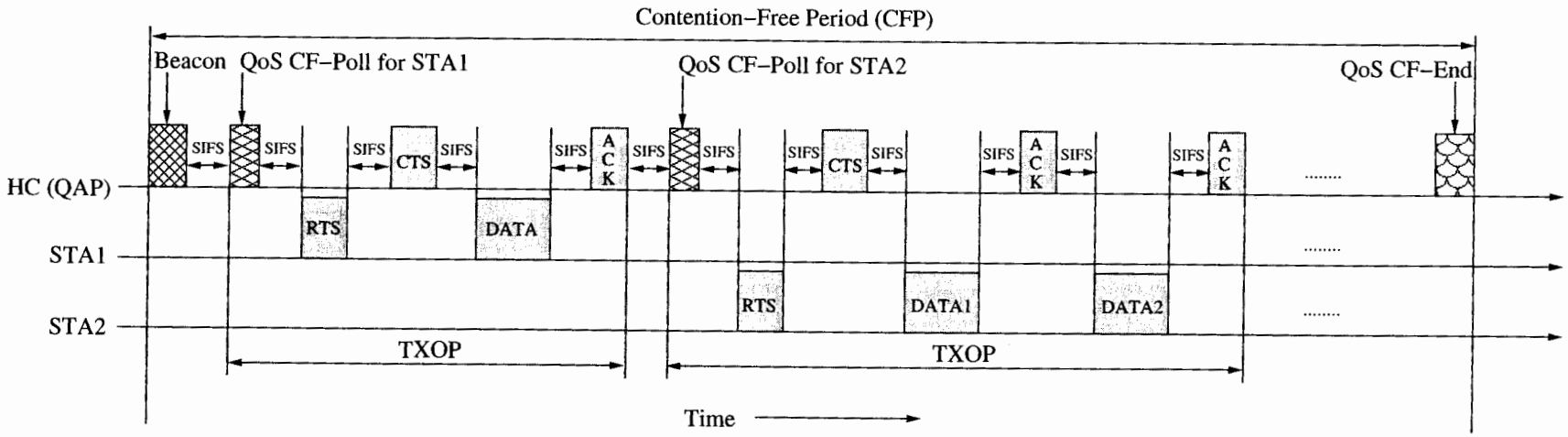


Figure 10.7. An example of HCF access mechanism.

time stations (*nrt*-stations) regulate their accesses to the channel according to the standard CSMA/CA protocol used in 802.11 DCF. The DBASE protocol permits real-time stations (*rt*-stations) to acquire excess bandwidth on demand. It is still compliant with the IEEE 802.11 standard [2].

Like the IEEE 802.11 standard, the DBASE protocol divides the frames into three priority classes. Frames belonging to different priority classes have to wait for different IFSs before they are transmitted. Stations have to wait for a minimum of PIFS before transmitting *rt*-frames such as reservation frame (RF) and request-to-send (RTS). The *nrt*-frames have the lowest priority, and hence stations have to wait for DIFS before transmitting such frames.

The Access Procedure for Non Real-Time Stations

The channel access method for *nrt*-stations is based on conventional DCF. An *nrt*-station with data traffic has to keep sensing the channel for an additional random time called data back-off time (DBT) after detecting the channel as being idle for a DIFS period. The DBT is given by

$$DBT = rand(a,\ b) \times slottime$$

The function $rand(a,\ b)$ returns a pseudo-random integer from a uniform distribution over an interval $[a,\ b]$, where b grows exponentially for each retransmission attempt, and the range of b is between b_{min} and b_{max}. DBASE adopts the contention window parameters from the IEEE 802.11 DSSS specification. If the channel is idle, the DBT counter is decremented till it reaches zero, but it is frozen while the channel becomes busy. Once the DBT counter reaches zero, the *nrt*-station transmits its *nrt*-frame. The destination sends an ACK to the source after SIFS period after receiving the *nrt*-frame correctly from the source.

The Access Procedure for Real-Time Stations

Each *rt*-station maintains a virtual reservation table (RSVT). In this virtual table, the information regarding all *rt*-stations that have successfully reserved the required bandwidth is recorded. Before initiating an *rt*-session, the *rt*-station sends an RTS in order to reserve the required bandwidth. Before transmitting the RTS, a corresponding entry is made in the RSVT of the node. Every station that hears this RTS packet also makes a corresponding entry in its RSVT. After recording into the RSVT successfully, an *rt*-station need not contend for the channel any more during its whole session.

Bandwidth Reservation

One of the *rt*-stations takes the responsibility of initiating the contention-free period (CFP) periodically. Such an *rt*-station is designated as CFP generator (CFPG). The CFP is utilized by the active *rt*-stations present in the network to transmit their *rt*-frames. The CFPG issues a reservation frame (RF) periodically and has the right to send its *rt*-frame first in the CFP. The maximum delay between any two consecutive RFs is D_{max}, where D_{max} is the minimum of maximum delay bounds among all

active *rt*-connections. The RF is a broadcast frame that announces the beginning of the CFP. The RF contains the information about the number of active *rt*-stations and the information about all *rt*-stations recorded in the RSVT of the CFPG.

Assume that at time t an *rt*-station wants to transmit data. Then it monitors the channel for detecting the RF during the interval $(t,\ t + D_{max})$. If the *rt*-station detects the RF, it waits until the CFP finishes. After the CFP finishes, the *rt*-station keeps sensing the channel for a period of real-time back-off time (RBT) after detecting the channel as being idle for a PIFS period. The RBT of an *rt*-station is given by

$$RBT = rand(c,\ d) \times slottime$$

where $rand(c,\ d)$ returns a pseudo-random integer from a uniform distribution over an interval $[c,\ d]$. The values of c and d are set to 0 and 3, respectively. If the channel is idle, the RBT counter is decremented till it reaches zero, but it is frozen while the medium is sensed busy. Once the RBT counter reaches zero, the *rt*-station contends for its reservation by sending an RTS packet. If no collision occurs, it updates its tables and transmits its first *rt*-frame. If a collision occurs, the P-persistent scheme is used to resolve the contention. The *rt*-station involved in collision retransmits the RTS in the next time-slot (*i.e., slottime*) with a probability P. With probability $(1 - P)$, it defers for at least one time-slot and recalculates the RBT using the following equation:

$$RBTP = rand(c + 1,\ d) \times slottime$$

where RBTP is the recalculated RBT for the P-persistent scheme.

If an RF is not received during the interval $(t,\ t + D_{max})$, it means that there are no active *rt*-stations. If the channel is still idle in the interval $(t + D_{max} + \delta,\ t + D_{max} + \delta + PIFS)$ and no RF is detected, the *rt*-station that wants to transmit data at time instant t will execute the back-off scheme. Here δ represents the remaining transmitting time of the current frame at the time instant $t + D_{max}$. During the back-off process, the *rt*-station should keep monitoring the channel to check whether any *rt*-station has started acting as the CFP generator. If RBT reaches zero, the *rt*-station sends an RTS frame to the receiver. If no collision occurs, it gets CTS from the receiver and plays the role of CFPG in the network. If a collision occurs, the P-persistent scheme as mentioned above is used to decide when the stations are to transmit again.

The bandwidth reservation scheme is illustrated in Figure 10.8. Figure 10.8 (a) depicts a case in which no collision occurs, while Figure 10.8 (b) shows a scenario in which a collision occurs. In Figure 10.8 (a), stations A and C have *rt*-frames for transmission to stations B and D, respectively. Besides these, station E has *nrt*-frames to be transmitted to station D. After listening to the channel for D_{max} time period in order to detect the presence of an RF, stations A and C conclude that no CFPG exists in the network. Then, if they find the channel as being idle for a PIFS period, they initiate their back-off timers. In this case, assume that RBT_A is one slot and RBT_C is three slots. During the back-off process, once the channel

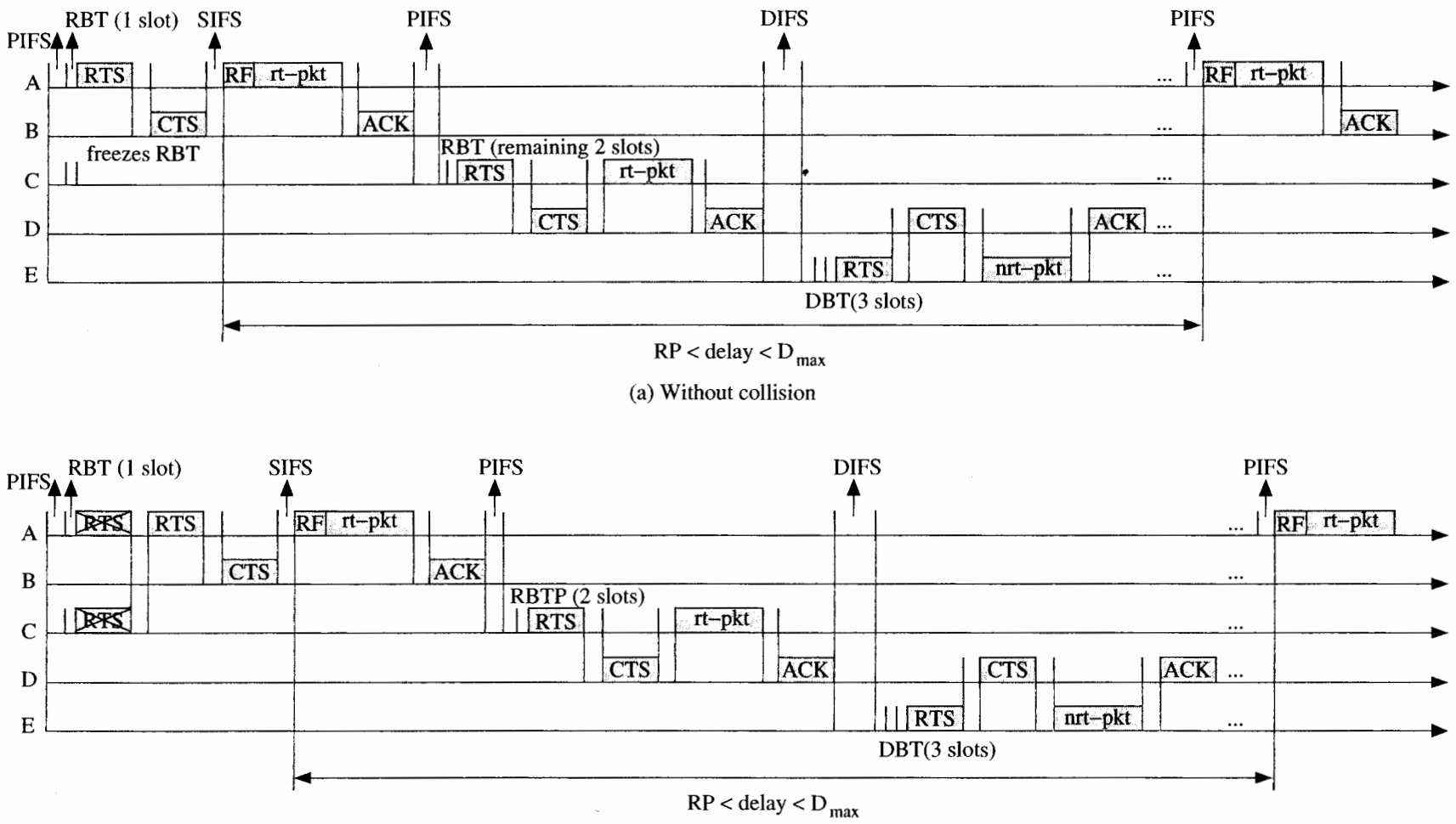


Figure 10.8. An example of new *rt*-stations joining the network.

becomes busy, the back-off timer of station C is paused as shown in Figure 10.8 (a). It is restarted from the same value once the channel becomes idle again. After RBT_A counts down to zero, station A seizes the channel and sends an RTS. When station A starts transmitting, station C pauses its back-off counter. If no collision occurs, station A receives a CTS within SIFS time duration. Then station A records its reservation information into the RSVT and becomes the CFPG. Since station A is currently playing the role of CFPG, it transmits an RF before transmitting its first *rt*-frame. Once station A completes its transmission, station C continues its back-off process. When RBT_C counts down to zero, station C reserves bandwidth by adding a corresponding entry into the RSVT and transmits its first *rt*-frame. When station E detects the channel as being idle for DIFS, it implies that no other *rt*-station wants to transmit currently, and hence station E sends its RTS as soon as DBT_E counts down to zero. By the end of a contention period whose length is limited by a parameter RP_{max} (maximum repetition period), bandwidth would be reserved for the *rt*-stations, and thereafter they need not exchange RTS/CTS control frames before transmitting their *rt*-frames. The delay between two RFs varies from real-time period (RP) to D_{max}, where RP is the sum of the CFP (*rt*-stations reserved period) and the CP for new *rt*-stations.

In Figure 10.8 (b), assume that both station A and station C generate RBT as one slot. After waiting for one time-slot, both transmit their RTS frames, which results in a collision. Then the P-persistent scheme is applied. Assume that station A gets access to the channel during the next slot itself, but station C does not. Then, station A will retransmit its RTS in the following slot, while station C initiates a new back-off time $RBTP_C$. If no collision occurs, station A gets a CTS within SIFS, and sends out an RF and its *rt*-frame. When $RBTP_C$ counts down to zero, station C seizes the channel to send an RTS. If any collision occurs, the *rt*-station uses the P-persistent scheme to resolve the collision. The collision resolution process is restricted from crossing the RP_{max} boundary.

The MAC layer solutions such as MACA/PR [12] and RTMAC [13] provide real-time traffic support in asynchronous ad hoc wireless networks. These solutions are discussed in Chapter 6.

10.5 NETWORK LAYER SOLUTIONS

The bandwidth reservation and real-time traffic support capability of MAC protocols can ensure reservation at the link level only, hence the network layer support for ensuring end-to-end resource negotiation, reservation, and reconfiguration is very essential. This section describes the existing network layer solutions that support QoS provisioning.

10.5.1 QoS Routing Protocols

QoS routing protocols search for routes with sufficient resources in order to satisfy the QoS requirements of a flow. The information regarding the availability of resources is managed by a resource management module which assists the QoS rout-

ing protocol in its search for QoS feasible paths. The QoS routing protocol should find paths that consume minimum resources. The QoS metrics can be classified as additive metrics, concave metrics, and multiplicative metrics.

An additive metric A_m is defined as $\sum_{i=1}^{h} L_i(m)$, where $L_i(m)$ is the value of metric m over link L_i and $L_i \epsilon$ P. The hop length of path P is h. A concave metric represents the minimum value over a path P and is formally defined as $C_m = min(L_i(m))$, $L_i(m) \epsilon$ P. A multiplicative metric represents the product of QoS metric values and is defined as $M_m = \prod_{i=1}^{h}(L_i(m))$, $L_i(m) \epsilon$ P. To find a QoS feasible path for a concave metric, the available resource on each link should be at least equal to the required value of the metric. Bandwidth is a concave metric, while cost, delay, and delay jitter are additive metrics. The reliability or availability of a link, based on some criteria such as link-break-probability, is a multiplicative metric. Finding an optimal path with multiple constraints may be an NP-complete problem if it involves two or more additive metrics. For example, finding a delay-constrained least-cost path is an NP-complete problem.

To assist QoS routing, the topology information can be maintained at the nodes of ad hoc wireless networks. The topology information needs to be refreshed frequently by sending link state update messages, which consume precious network resources such as bandwidth and battery power. Otherwise, the dynamically varying network topology may cause the topology information to become imprecise. This trade-off affects the performance of the QoS routing protocol. As path breaks occur frequently in ad hoc wireless networks, compared to wired networks where a link goes down very rarely, the path satisfying the QoS requirements needs to be recomputed every time the current path gets broken. The QoS routing protocol should respond quickly in case of path breaks and recompute the broken path or bypass the broken link without degrading the level of QoS. In the literature, numerous routing protocols have been proposed for finding QoS paths. In the following sections, some of these QoS routing protocols are described.

10.5.2 Ticket-Based QoS Routing Protocol

Ticket-based QoS routing [14] is a distributed QoS routing protocol for ad hoc wireless networks. This protocol has the following features:

- It can tolerate imprecise state information during QoS route computation and exhibits good performance even when the degree of imprecision is high.
- It probes multiple paths in parallel for finding a QoS feasible path. This increases the chance of finding such a path. The number of multiple paths searched is limited by the number of tickets issued in the probe packet by the source node. State information maintained at intermediate nodes is used for more accurate route probing. An intelligent hop-by-hop selection mechanism is used for finding feasible paths efficiently.
- The optimality of a path among several feasible paths is explored. A low-cost path that uses minimum resources is preferred when multiple feasible paths are available.

- A primary-backup-based fault-tolerant technique is used to reduce service disruption during path breaks that occur quite frequently in ad hoc wireless networks.

Protocol Overview

The basic idea of the ticket-based probing protocol is that the source node issues a certain number of tickets and sends these tickets in probe packets for finding a QoS feasible path. Each probe packet carries one or more tickets. Each ticket corresponds to one instance of the probe. For example, when the source node issues three tickets, it means that a maximum of three paths can be probed in parallel. The number of tickets generated is based on the precision of state information available at the source node and the QoS requirements of the connection request. If the available state information is not precise or if the QoS requirements are very stringent, more tickets are issued in order to improve the chances of finding a feasible path. If the QoS requirements are not stringent and can be met easily, fewer tickets are issued in order to reduce the level of search, which in turn reduces the control overhead. There exists a trade-off here between the performance of the QoS routing protocol and the control overhead.

The state information, at the source node, about intermediate nodes is useful in finding a much better QoS path, even if such information is not precise. The state information maintained at each node is comprised of estimations of end-to-end delay and available path bandwidth for every other node present in the network. When an intermediate node receives a probe packet, it is either split to explore more than one path or is forwarded to just one neighbor node based on the state information available at that intermediate node.

Based on the idea of ticket-based probing, two heuristic algorithms are proposed, one for delay-constrained QoS routing, and the other for bandwidth-constrained QoS routing. In delay-constrained QoS routing, each probe accumulates the delay of the path it has traversed so far. In other words, if an intermediate node A receives a probe packet (PKT) from a neighbor node B, node A updates the delay field in PKT by adding delay value of the link between nodes B and A. Then node A determines the list of candidate neighbors to which it has to send probe packets. It distributes tickets present in PKT among these new probe packets and then forwards these probe packets to the respective candidate neighbors. If multiple probe packets arrive at the destination node (with each carrying the list of intermediate nodes along its path), node A selects the path with least cost as the primary path and the other paths as the backup paths which will be used when the primary path is broken due to the mobility of intermediate nodes.

Optimizing Cost of a Feasible Path

This protocol searches for the lowest cost path among the feasible paths. This is done during the QoS path probing. The source node issues two types of tickets, yellow tickets and green tickets, and sends them along with probe packets. Yellow tickets prefer paths that satisfy the requirement of a probe in terms of QoS metrics.

For example, in delay-constrained QoS routing, yellow tickets are used to search for paths that have least delay, such that the end-to-end delay requirement is met. If the delay requirement is very large and can be met easily, only one yellow ticket is issued. If the delay requirement is too small to be met, then the source node does not issue any yellow ticket and rejects the connection request. Otherwise, more than one yellow ticket is issued to search multiple paths for finding a feasible QoS path. Green tickets are used to search for QoS paths with low costs. Similar to the manner in which the source node determines the number of yellow tickets, it also determines the number of green tickets to be issued on the basis of the delay requirement of the connection request. The distribution of yellow and green tickets (by an intermediate node to its candidate neighbors) is based on the delay and cost requirements of the connection request, respectively. The concept behind two types of tickets is to use the more aggressive green tickets to find a least cost feasible path, and use yellow tickets as a backup to maximize the probability of finding a feasible path.

Path Rerouting and Path Maintenance

This protocol suggests a primary-backup-based, fault-tolerant technique to cope up with the network dynamics. To tolerate faults, a multi-level redundancy scheme is proposed. For the highest level of redundancy, multiple paths (preferably disjoint) are probed and data is routed independently on all paths. The destination node selects the first data copy and discards other copies which arrive later. Another level of redundancy which requires less resources has also been proposed. Here, one path is selected as the primary path and other paths (having resources reserved) act as backup paths. The third type of redundancy incurs even less control overhead and consumes very few resources. Here, backup paths are available along with the primary path, but resources are not reserved in these backup paths. During path maintenance, in order to eliminate the broken link, the call is rerouted over a backup path which has enough resources to satisfy the QoS requirements of the call. In the case of the third type of redundancy, since no resource reservation has been done along backup paths, during path breaks it is extremely difficult to find a backup that has enough resources to satisfy the QoS requirements of the call.

Advantages and Disadvantages

The objective of ticket-based probing is to improve the average call acceptance ratio (ACAR) of ad hoc wireless networks. ACAR is the ratio of the number of calls accepted to the number of calls received by the network. The protocol adapts dynamically to the requirements of the application and the degree of imprecision of state information maintained. It offers a trade-off between control overhead incurred in finding a feasible path and the cost of a feasible path. As the maximum number of probes in the network is equal to the number of tickets issued, the control overhead is bound by the number of tickets. The performance of the protocol depends on the ticket-issuing mechanism at the source node and the ticket-splitting procedure at the intermediate nodes.

The protocol assumes that each node has global state information, but maintaining such information incurs huge control overhead in the already bandwidth-constrained ad hoc wireless networks. The proposed heuristic algorithms, which are based on an imprecise state information model, may fail in finding a feasible path in the extreme cases where the topology changes very rapidly. In delay-constrained QoS routing, the queuing delay and the processing delay at the intermediate nodes are not taken into consideration while measuring the delay experienced so far by the probe packet. This may cause some data packets to miss their deadlines. The routing algorithm works well only when the average lifetime of an established path is much longer than the average rerouting time. During the rerouting process, if QoS requirements are not met, data packets are transmitted as best-effort packets. This may not be acceptable for applications that have stringent QoS requirements.

10.5.3 Predictive Location-Based QoS Routing Protocol

The predictive location-based QoS routing protocol (PLBQR) [15] is based on the prediction of the location of nodes in ad hoc wireless networks. The prediction scheme overcomes to some extent the problem arising due to the presence of stale routing information. No resources are reserved along the path from the source to the destination, but QoS-aware admission control is performed. The network does its best to support the QoS requirements of the connection as specified by the application. The QoS routing protocol takes the help of an update protocol and location and delay prediction schemes. The update protocol aids each node in broadcasting its geographic location and resource information to its neighbors. Using the update messages received from the neighbors, each node updates its own view of the network topology. The update protocol has two types of update messages, namely, *Type 1 update* and *Type 2 update*. Each node generates a Type 1 update message periodically. A Type 2 update message is generated when there is a considerable change in the node's velocity or direction of motion. From its recent update messages, each node can calculate an expected geographical location where it should be located at a particular instant and then periodically checks if it has deviated by a distance greater than δ from this expected location. If it has deviated, a Type 2 update message is generated.

Location and Delay Predictions

In establishing a connection to the destination D, the source S first has to predict the geographic location of node D and the intermediate nodes, at the instant when the first packet reaches the respective nodes. Hence, this step involves location prediction as well as propagation delay prediction. The location prediction is used to predict the geographic location of the node at a particular instant t_f in the future when the packet reaches that node. The propagation delay prediction is used to estimate the value of t_f used in the above location prediction. These predictions are performed based on the previous update messages received from the respective nodes.

Location Prediction

Let (x_1, y_1) at t_1 and (x_2, y_2) at t_2 $(t_2 > t_1)$ be the latest two updates from the destination D to the source node S. Assume that the second update message also indicates v, which is the velocity of D at (x_2, y_2). Assume that node S wants to predict the location (x_f, y_f) of node D at some instant t_f in the future. This situation is depicted in Figure 10.9. The value of t_f has to be estimated first using the delay prediction scheme, which will be explained later in this section. From Figure 10.9, using similarity of triangles, the following equation is obtained:

$$\frac{y_2 - y_1}{y_f - y_1} = \frac{x_2 - x_1}{x_f - x_1} \tag{10.5.1}$$

By solving the above equation for y_f,

$$y_f = y_1 + \frac{(x_f - x_1)(y_2 - y_1)}{x_2 - x_1} \tag{10.5.2}$$

Using the above Equation 10.5.2, source S can calculate y_f if it knows x_f, which in turn can be calculated as follows. Using similarity of triangles again, the following equation is obtained:

$$y_f - y_2 = \frac{(y_2 - y_1)(x_f - x_2)}{x_2 - x_1} \tag{10.5.3}$$

By using the Pythagorean theorem,

$$(x_f - x_2)^2 + (y_f - y_2)^2 = v^2(t_f - t_2)^2 \tag{10.5.4}$$

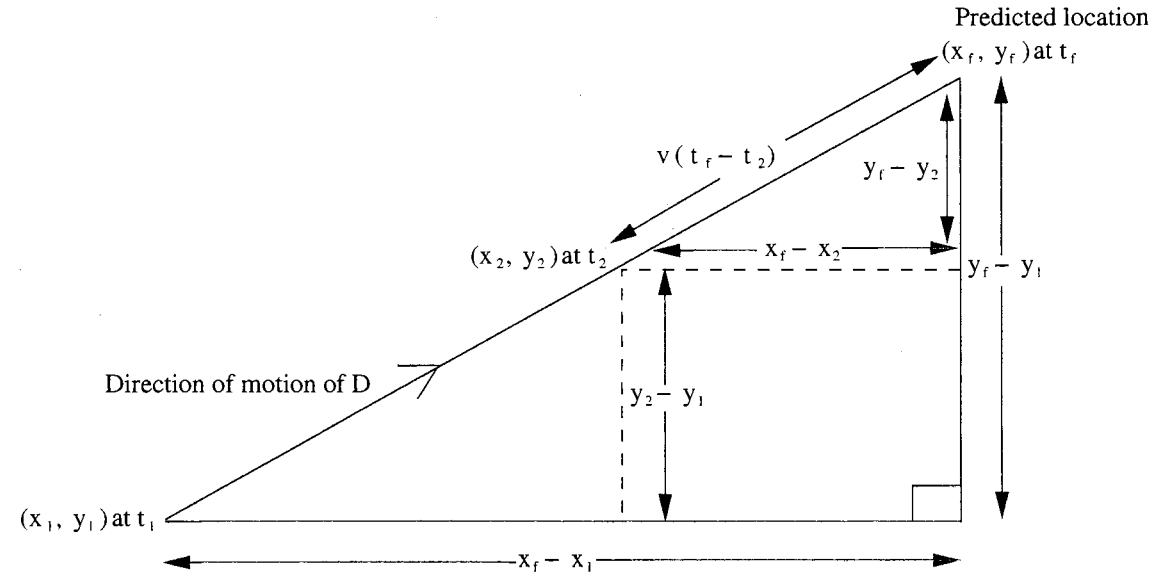


Figure 10.9. Prediction of location at a future time by node S using the last two updates.

Substituting for $y_f - y_2$ from Equation 10.5.3 in the above Equation 10.5.4 and solving for x_f, the following equation is obtained:

$$x_f = x_2 + \frac{v(t_f - t_1)(x_2 - x_1)}{\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{10.5.5}$$

If updates include the direction information of nodes, only one previous update is required to predict future location (x_f, y_f). The calculation of (x_f, y_f) is then exactly the same as that of the periodic calculation of expected location (x_e, y_e) by the update protocol [15].

Delay Prediction

The source node S has to predict the time instant t_f at which a packet reaches the given destination node or intermediate node D. This can be known only if the end-to-end delay between nodes S and D is known. It is assumed that the end-to-end delay for a data packet from node S to node D is equal to the delay experienced by the latest update message received by node S from node D.

QoS Routing

Each node in the network has information about the complete network topology, which is refreshed by means of update messages. Using this information, the source node performs source-routing. The network state information is maintained in two tables, namely, the *update table* and the *routing table*. When node A receives an update message from node B, node A updates the corresponding entry for node B in the update table. In that entry, node A stores the ID of node B, the time instant at which the update packet was sent, the time at which the update packet was received, the geographic coordinates, speed, resource parameters of node B, and optionally the direction of motion of node B. For each node N in the network, node A stores the last two update packets received from that node in its update table. For some nodes, node A also maintains proximity lists. The proximity list of node K is a list of all nodes lying within a distance 1.5×transmission range of node K. The proximity lists are used during route computation. By maintaining a proximity list rather than a neighbor list for node K (*i.e.*, list of nodes lying within node K's transmission range), node A also considers the nodes that were outside node K's transmission range at the time their respective last updates were sent, but that have since moved into node K's transmission range, while computing the neighbors of node K. The routing table at node A contains information about all active connections with node A as source. When an update message from any node in the network reaches node A, it checks if any of the routes in its routing table is broken or is about to be broken. In either case, route recomputation is initiated. Using the location prediction based on the updates, it is possible to predict whether any link on the path is about to break. Thus, route recomputation can be initiated even before the route actually breaks.

The routing algorithm given in [15] works as follows. The source node S first runs location and delay predictions on each node in its proximity list in order to

obtain a list of its neighbors at present. It determines which of these neighbors have the resources to satisfy the QoS requirements of the connection (the neighbors that satisfy the QoS requirements are called candidates). Then it performs a depth-first search for the destination, starting with each of these candidate neighbors to find all candidate routes satisfying the QoS requirements of the connection request. From the resulting candidate routes, the geographically shortest route is chosen and the connection is established. Data packets are forwarded along this chosen route until the end of the connection or until the route is recomputed in anticipation of breakage. Note that only node S uses its view of the network for the entire computation.

Advantages and Disadvantages

PLBQR protocol uses location and delay prediction schemes which reduce to some extent the problem arising due to the presence of stale routing information. Using the prediction schemes, it estimates when a QoS session will experience path breaks and proactively finds an alternate path to reroute the QoS session quickly. But, as no resources are reserved along the route from the source to the destination, it is not possible to provide hard QoS guarantees using this protocol. Even soft QoS guarantees may be broken in cases when the network load is high. Since the location prediction mechanism inherently depends on the delay prediction mechanism, the inaccuracy in delay prediction adds to the inaccuracy of the location prediction. The end-to-end delay for a packet depends on several factors such as the size of the packet, current traffic load in the network, scheduling policy and processing capability of intermediate nodes, and capacity of links. As the delay prediction mechanism does not take into consideration some of the above factors, the predictions made by the location prediction mechanism may not be accurate, resulting in QoS violations for the real-time traffic.

10.5.4 Trigger-Based Distributed QoS Routing Protocol

The trigger-based (on-demand) distributed QoS routing (TDR) protocol [16] was proposed by De *et al.* for supporting real-time applications in ad hoc wireless networks. It operates in a distributed fashion. Every node maintains only the local neighborhood information in order to reduce computation overhead and storage overhead. To reduce control overhead, nodes maintain only the active routes. When a link failure is imminent, TDR utilizes the global positioning system-based (GPS) location information of the destination to localize the reroute queries only to certain neighbors of the nodes along the source-to-destination active route. For a quick rerouting with reduced control overhead, rerouting is attempted from the location of an imminent link failure, called intermediate node-initiated rerouting (INIR). If INIR fails, then in order to keep the flow state disruption to a minimum, rerouting is attempted from the source, which is termed source-initiated rerouting (SIRR).

Database Management

All nodes in the network maintain the local neighborhood information. For each neighbor, every node maintains *received power level*, current geographic coordinates, velocity, and direction of motion in the database.

Activity-Based Database

In addition to the local neighborhood information, node N maintains a source table ST_N, a destination table DT_N, or an intermediate table IT_N based on whether it actively participates in a session as the source (S), the destination (D), or as an intermediate node (I), respectively. These tables are referred to as the activity-based database. For a session, the source table contains the following fields: session ID, source ID, destination ID, maximum bandwidth demand (MaxBW), maximum acceptable delay (MaxDelay measured in terms of hop count), destination location (DLoc), next-node ID (NID) toward the destination, and activity flag (NodActv). An intermediate table contains the following fields: session ID, source ID, destination ID, source location (SLoc), MaxBW, MaxDelay, DLoc, NID, previous-node ID toward the source (PID), distance from the source (measured in terms of hop count), and NodActv. The destination table contains the following fields: session ID, source ID, destination ID, SLoc, MaxBW, MaxDelay, PID, distance from the source (hop count), and NodActv. At any time instant, a node may have to maintain one or more tables simultaneously for different on-going sessions. Each node N also maintains an updated residual bandwidth ($ResiBW_N$) which indicates its ability to participate in a session. A soft state approach is used to maintain the activity-based database. Hence, the database needs to be refreshed periodically. It is refreshed when data packets belonging to the on-going sessions are received by a node.

Routing Protocol

The messages that are exchanged for initiating, maintaining, and terminating a real-time session are described below.

Initial Route Discovery

If the source S has enough $ResiBW_S$ to satisfy the MaxBW for the session, the required bandwidth is temporarily reserved for a certain duration within which it expects an acknowledgment from the destination D. If the source knows the location of the destination, it performs route discovery through selective forwarding. In this approach, the source node takes advantage of location information of its neighbors and forwards route requests to only selective neighbors that are lying closely toward the destination node and satisfying QoS requirements of the connection request. Otherwise, the source initiates a flooding-based initial route discovery process. Before transmitting the route discovery packet, an entry is made in the source table ST_S for this session with NodActv flag set to zero (*i.e.*, idle). To ensure the stability of routes and in order to reduce the control overhead, only selected neighbors, from which packets were received with power level more than a threshold level (P_{th1}), are considered during route establishment. After receiving

a route discovery packet, an intermediate node (IN) checks in its IT_{IN} whether any such packet was already received for the same session. If so, the current route discovery packet is rejected to ensure loop-free routing. Otherwise, it is the first discovery packet for a session. Then the intermediate node (IN) increments the hop-count field of the received packet by one and checks for $ResiBW_{IN}$. If it can meet the MaxBW requirement for the session and if the updated hop-count field is less than MaxDelay, the required bandwidth is temporarily reserved, and an entry is made into the activity table IT_{IN} for the session with NodActv flag set to zero. Then the packet is forwarded to its downstream neighbors with the updated NID field. If either or both of *ResiBW* and MaxDelay criteria cannot be satisfied, the discovery packet is simply dropped. Upon receiving the first discovery packet, if the destination D is also able to satisfy both the *ResiBW* and the MaxDelay criteria, the discovery packet and the corresponding route are accepted.

Route/Reroute Acknowledgment

After accepting the route, the destination node D builds DT_D table with the NodActv flag set to 1 (*i.e.,* active) and sends an ACK to the source S along the selected route. On receiving the ACK packet, all intermediate nodes and the source S set the NodActv flags in their respective tables to 1 and refresh their *ResiBW* status. The packet transmission for the session follows immediately.

Alternate Route Discovery

In SIRR, when the received power level at an intermediate node falls below a threshold P_{th2}, the intermediate node sends a rerouting indication to the source S. Then the source S initiates the rerouting process through selective forwarding. But in INIR, when the power level of a packet received from the next node toward the destination falls below a threshold P_{th1} ($P_{th1} > P_{th2}$), it initiates a status query packet toward the source with appropriate identification fields and with a flag field called route repair status (*RR_Stat*) set to zero. If any upstream node is in the rerouting process, upon reception of the status query packet it sets the *RR_Stat* flag to 1 and sends the status reply packet to the querying node. On arriving at the source, the status query packet is discarded. If the querying node receives no status reply packet before its received power level from the downstream node goes below P_{th2}, it triggers the alternate route discovery process (*i.e.,* SIRR). Otherwise, it relinquishes control of rerouting. This query/reply process eliminates the chances of duplicate reroute discovery for a session. In both SIRR and INIR, the alternate route discovery process is similar to the initial route discovery except that the rerouting process takes advantage of the location information of the local neighbors and the approximate location of the destination, and forwards the rerouting requests to only selected neighbors that are close to the destination and that satisfy the delay and bandwidth constraints. The threshold parameters P_{th1} and P_{th2} have to be selected judiciously in order to avoid unnecessary rerouting.

Route Deactivation

In case of session completion or termination, the source node purges its corresponding *ST* table and sends a route deactivation packet toward the destination.

Upon receiving a deactivation request, each node which was part of that session updates its *ResiBW* and purges the activity table for that session. No explicit deactivation packet is sent in case of rerouting, as the new route could still consist of some nodes that were part of the old route.

Advantages and Disadvantages

In TDR protocol, if the source node knows the location of the destination node, it performs route discovery through selective forwarding to reduce the control overhead. For a quick rerouting with reduced control overhead and to reduce the packet loss during path breaks, it uses INRR and SIRR schemes. However, in this protocol a QoS session is rerouted if the received power level from a downstream node falls below a certain value (*i.e.,* threshold). Due to small-scale fading, the received power level may vary rapidly over short periods of time or distance traveled. Some of the factors that influence fading are multipath propagation, velocity of the nodes, and bandwidth of the channel. Even though the downstream node may be within the transmission range of the upstream node, due to fading the received power level at the upstream node may fall below the threshold value. This increases the control overhead because of initiation of the alternate route discovery process and false rerouting of some of the sessions.

10.5.5 QoS-Enabled Ad Hoc On-Demand Distance Vector Routing Protocol

Perkins *et al.* have extended the basic ad hoc on-demand distance vector (AODV) routing protocol to provide QoS support in ad hoc wireless networks [17]. To provide QoS, packet formats have been modified in order to specify the service requirements which must be met by the nodes forwarding a *RouteRequest* or a *RouteReply.*

QoS Extensions to AODV Protocol

Several modifications have been carried out for the routing table structure and *RouteRequest* and *RouteReply* messages in order to support QoS routing. Each routing table entry corresponds to a different destination node. The following fields are appended to each routing table entry:

- Maximum delay
- Minimum available bandwidth
- List of sources requesting delay guarantees
- List of sources requesting bandwidth guarantees

Maximum Delay Extension Field

The maximum delay extension field is interpreted differently for *RouteRequest* and *RouteReply* messages. In a *RouteRequest* message, it indicates the maximum time (in seconds) allowed for a transmission from the current node to the destination

node. In a *RouteReply* message, it indicates the current estimate of cumulative delay from the current intermediate node forwarding the *RouteReply*, to the destination. Using this field the source node finds a path (if it exists) to the destination node satisfying the maximum delay constraint. Before forwarding the *RouteRequest*, an intermediate node compares its *node traversal time* (*i.e.*, the time it takes for a node to process a packet) with the (remaining) delay indicated in the maximum delay extension field. If the delay is less than the node traversal time, the node discards the *RouteRequest* packet. Otherwise, the node subtracts node traversal time from the delay value in the extension and processes the *RouteRequest* as specified in the AODV protocol.

The destination node returns a *RouteReply* with the maximum delay extension field set to zero. Each intermediate node forwarding the *RouteReply* adds its own node traversal time to the delay field and forwards the *RouteReply* toward the source. Before forwarding the *RouteReply* packet, the intermediate node records this delay value in the routing table entry for the corresponding destination node.

Minimum Bandwidth Extension Field

In a *RouteRequest* message, this field indicates the minimum bandwidth (in Kbps) that must be available along an acceptable path from the source to the destination. In a *RouteReply* message, it indicates the minimum bandwidth available on the route between the node forwarding the *RouteReply* and the destination node. Using this field, the source node finds a path (if it exists) to the destination node satisfying the minimum bandwidth constraint. Before forwarding the *RouteRequest*, an intermediate node compares its available bandwidth with the bandwidth field in the extension. If the requested amount of bandwidth is not available, the node discards the *RouteRequest* message. Otherwise, the node processes the *RouteRequest* as specified in the AODV protocol.

The destination node returns a *RouteReply* in response to a *RouteRequest* with the bandwidth field set to infinity (a very large number). Each node forwarding the *RouteReply* compares the bandwidth field in the *RouteReply* with its own link capacity and updates the bandwidth field of the *RouteReply* with the minimum of the two, before forwarding the *RouteReply*. This value is also stored in the routing table entry for the corresponding destination and indicates the minimum available bandwidth to the destination.

List of Sources Requesting QoS Guarantees

A *QoSLost* message is generated when an intermediate node experiences an increase in node traversal time or a decrease in the link capacity. The *QoSLost* message is forwarded to all sources potentially affected by the change in the QoS parameter. These are the sources to which *RouteReply*s with QoS extension have been forwarded by the node earlier.

Advantages and Disadvantages

The advantage of QoS AODV protocol is the simplicity of extension of the AODV protocol that can potentially enable QoS provisioning. However, as no resources are reserved along the path from the source to the destination, this protocol is not suitable for applications that require hard QoS guarantees. Further, node traversal time is only the processing time for the packet, so the major part of the delay at a node is contributed by packet queuing and contention at the MAC layer. Hence, a packet may experience much more delay than this when the traffic load is high in the network.

10.5.6 Bandwidth Routing Protocol

The bandwidth routing (BR) protocol [18] consists of an end-to-end path bandwidth calculation algorithm to inform the source node of the available bandwidth to any destination in the ad hoc network, a bandwidth reservation algorithm to reserve a sufficient number of free slots for the QoS flow, and a standby routing algorithm to reestablish the QoS flow in case of path breaks.

Here, only bandwidth is considered to be the QoS parameter. In TDMA-based networks, bandwidth is measured in terms of the number of free slots available at a node. The goal of the bandwidth routing algorithm is to find a shortest path satisfying the bandwidth requirement. The transmission time scale is organized into frames, each containing a fixed number of time-slots. The entire network is synchronized on a frame and slot basis. Each frame is divided into two phases, namely, the control phase and the data phase. The control phase is used to perform the control functions such as slot and frame synchronization, virtual circuit (VC) setup, and routing. The data phase is used for transmission/reception of data packets. For each node, a slot is assigned in the control phase for it to broadcast its routing information and slot requirements. At the end of the control phase, each node knows about the channel reservations made by its neighbors. This information helps nodes to schedule free slots, verify the failure of reserved slots, and drop expired real-time packets. The BR protocol assumes assumes a half-duplex CDMA-over-TDMA system in which only one packet can be transmitted in a given slot.

Bandwidth Calculation

Since the network is multi-hop in nature, the free slots recorded at each node may be different. The set of common free slots between two adjacent nodes denotes the link bandwidth between them. The path bandwidth between two nodes is the maximum bandwidth available in the path between them. If the two nodes are adjacent, the path bandwidth between them equals their link bandwidth. For example, consider two adjacent nodes, node A and node B, having free slots {2,5,6,8} and {1,2,4,5}, respectively. The link bandwidth $linkBW(A, B) = freeslot(A) \cap freeslot(B) = \{2, 5\}$. It means that only slots 2 and 5 can be used by nodes A and B for transmitting data packets to each other. The $freeslot(X)$ is defined as the set of slots which are not used by any adjacent node of node X (to receive or to send) from the point of view of node X.

To compute the end-to-end bandwidth for a path in a TDMA-based network, one has to know not only the available bandwidth on the individual links on the path, but also determine the scheduling of the free slots. The BR protocol also provides a heuristic-based hop-by-hop path bandwidth calculation algorithm to assign free slots at every hop along the path. The call admission control mechanism of the BR protocol uses the information regarding the availability of end-to-end bandwidth while making a decision on whether to admit or reject a new QoS session. The path bandwidth calculation algorithm is explained with the help of the example shown in Figure 10.10, where a path from source node S to destination node D is illustrated. The process of computing $pathBW(S, D)$ is explained below.

- $pathBW(S, A)$: Since node S and node A are adjacent, the $pathBW(S, A) = linkBW(A, S)$, which is four slots. The four slots are $\{2, 5, 6, 7\}$.
- $pathBW(S, B)$: Since $pathBW(S, A) = linkBW(A, B) = \{2, 5, 6, 7\}$, if S uses slots 6 and 7 to send packets to A, then A can use only slots 2 and 5 for transmission of packets to B. This is because a node cannot be in transmission and reception modes simultaneously. Hence $pathBW(S, B)$ is two slots, by assigning slots $\{6, 7\}$ on link(S, A) and slots $\{2, 5\}$ on link(A, B).

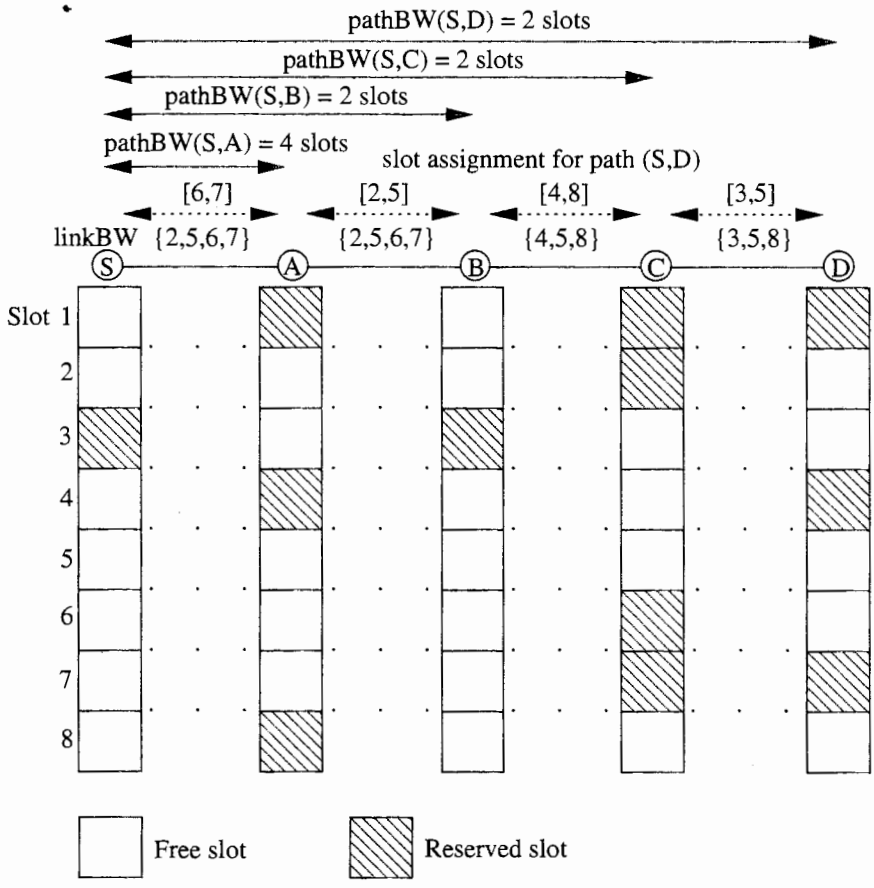


Figure 10.10. An example of path bandwidth calculation in BR protocol.

- $pathBW(S, C)$: Here slots 4 and 8 are exclusively available for $linkBW(B, C)$, slot 2 is exclusively available for $pathBW(S, B)$, and slot 5 is common for both of them. So assign one of slots 4, 8 to link(B, C), for example, assign slot 4 to link(B, C), and slot 2 to path(S, B). For achieving maximum bandwidth, assign slot 8 to link (B, C) and slot 5 to path(S, B). Hence, $pathBW(S, C)$ is 2 slots, by assigning slots $\{6, 7\}$ on link(S, A), slots $\{2, 5\}$ on link(A, B), and slots $\{4, 8\}$ on link(B, C).

- $pathBW(S, D)$: This case is similar to the previous one. So slots 4 and 8 are assigned to path(S, C) and slots 3 and 5 are assigned to link(C, D) to get two slots for $pathBW(S, D)$.

Slot Assignment

The path bandwidth calculation algorithm requires periodic exchange of bandwidth information. The slot assignment algorithm in each node assigns free slots during the call setup. When a node receives a call setup packet, it checks whether the slots that the immediate sender will use for transmission are free, and it also finds out if there are free slots that can be used for forwarding the incoming packets. If such free slots are available, the slot assignment algorithm reserves the required number of slots, updates the routing table, and then forwards the call setup packet to the next hop. If the required number of slots are not available at the node, all the reservations that have been made so far along the path from the source node to the current node have to be canceled in order to release the slots assigned for this connection. This is done by sending a *Reset* packet back to the source along the path that has been established so far. If reservations are made successfully along the path from the source to the destination, the destination sends a *Reply* packet back to the source to acknowledge having set up the connection. The reservations are soft state in nature in order to avoid resource lock-up at intermediate nodes due to path breaks.

Standby Routing Mechanism

The connections may get broken due to dynamic changes in the network topology. The standby routing mechanism has to reestablish such broken connections. Secondary paths are maintained in the routing table, which can be used when the primary path fails. The standby route is easily computed using the DSDV algorithm [19] without any extra overhead. Each node periodically exchanges routing information with its neighboring nodes. The neighbor with the shortest distance to the destination node becomes the next node on the primary path to the destination node. The neighbor node with the second shortest distance to the destination becomes the next node on the standby route to the destination. It is to be noted that this standby route is not guaranteed to be a link- or node-disjoint one. When a primary path fails, the upstream node that detects the link break will try to rebuild a new path immediately, using the standby route. If the standby route satisfies the

QoS requirements, the new path from the point of the path break is established by sending a call setup packet hop-by-hop to the destination through the standby path.

Since this scheme follows DSDV protocol, a table-driven routing protocol, and uses on-demand call admission control, similar to the on-demand routing protocols, it is classified into the category of hybrid solutions in the classifications in Figure 10.2.

Advantages and Disadvantages

The BR protocol provides an efficient bandwidth allocation scheme for CDMA-over-TDMA-based ad hoc wireless networks. The standby routing mechanism can reduce the packet loss during path breaks. But the CDMA-over-TDMA channel model that is used in this protocol requires assigning a unique control slot in the control phase of superframe for each node present in the network. This assignment has to be done statically before commissioning the network. Due to this, it is not possible for a new node to enter into the network at a later point of time. If a particular node leaves the network, the corresponding control slot remains unused and there is no way to reuse such a slot(s). Further, the network needs to be fully synchronized.

10.5.7 On-Demand QoS Routing Protocol

Lin proposed an admission control scheme over an on-demand QoS routing (OQR) protocol [20] to guarantee bandwidth for real-time applications. Since routing is on-demand in nature, there is no need to exchange control information periodically and maintain routing tables at each node. Similar to the bandwidth routing (BR) protocol, the network is time-slotted and bandwidth is the key QoS parameter. The path bandwidth calculation algorithm proposed in BR is used to measure the available end-to-end bandwidth. The on-demand QoS routing protocol is explained below.

Route Discovery

During the route discovery process, the source node that wants to find a QoS route to the destination floods a QoS route request (QRREQ) packet. A QRREQ packet contains the following fields: packet type, source ID, destination ID, sequence number, route list, slot array list, data, and TTL. The pair {source ID, sequence number} is used to uniquely identify a packet. For each QRREQ packet, the source node uses a new sequence number (which is monotonically increasing) in order to avoid multiple forwarding of the same packet by intermediate nodes. The route list records the nodes that have been visited by the QRREQ packet, whereas the slot array list records free slots available at each of these nodes. The TTL field limits the maximum length of the path to be found. A node N receiving a QRREQ packet performs the following operations:

1. If a QRREQ with the same {source ID, sequence number} had been received already, this QRREQ packet gets discarded.

2. Otherwise, the route list field is checked for the address of this node N. If it is present, node N discards this QRREQ packet.

3. Otherwise,

   - Node N decrements TTL by one. If TTL counts down to zero, it discards this QRREQ packet.
   - It calculates the path bandwidth from the source to this node. If it satisfies the QoS requirement, node N records the available free slots in the slot array list of the QRREQ packet. Otherwise, node N discards this QRREQ packet.
   - Node N appends the address of this node to the route list and rebroadcasts this QRREQ packet if it is not the destination.

For the example shown in Figure 10.10, assume that the source S floods a QRREQ packet with bandwidth requirement of two time-slots. Here, the destination D receives a QRREQ packet with the following information in its fields. The route list field contains (S, A, B, C) and the slot array list contains ([A, {2, 5, 6, 7}], [B, {2, 5}], [C, {4, 5}], [D, {3, 8}]). The destination may receive more than one QRREQ packet, each giving a unique feasible QoS path from the source to the destination.

Bandwidth Reservation

The destination node may receive one or more QRREQ packets, each giving a feasible QoS path for the connection request. The destination node selects the least-cost path among them. Then it copies the fields {route list, slot array list} from the corresponding QRREQ packet to the QoS Route Reply (QRREP) packet and sends the QRREP packet to the source along the path recorded in the route list. As the QRREP traverses back to the source, each node recorded in the route list reserves the free slots that have been recorded in the slot array list field. Finally, when the source receives the QRREP, the end-to-end bandwidth reservation process is completed successfully. The reservations made are soft state in nature in order to avoid resource lock-up. The source can start sending data packets in the data phase. At the end of the session, all reserved slots are released.

Reservation Failure

The reservation of bandwidth may fail, either due to route breaks or because the free slots that are recorded in the slot array list get occupied by some other connection(s) before the QRREP packet sent by the destination reaches the corresponding intermediate nodes. In the second case, the node at which reservation fails, sends a *ReserveFail* packet to the destination node. The destination then restarts the reservation process along the next feasible path. All nodes on the path from the interrupted node to the destination free the reserved slots for this connection on

receiving the *ReserveFail* packet. If no connection could be set up due to non-availability of feasible paths, the destination broadcasts a *NoRoute* packet to notify the source. Then the source either restarts the route discovery process, if it still needs a connection to the destination, or rejects the call.

Route Maintenance

When a route gets broken, the nodes detecting the link break send a *RouteBroken* packet to the source and the destination nodes. In other words, once the next hop becomes unreachable, the upstream node which is toward the source node sends a *RouteBroken* packet to the source, and the downstream node which is toward the destination sends another *RouteBroken* packet to the destination. The intermediate nodes relaying the *RouteBroken* packet release all reserved slots for this connection and drop all data packets of this connection which are still pending in their respective queues. After receiving the *RouteBroken* packet, the source restarts the route discovery process in order to reestablish the connection over a new path, while the destination releases resources reserved for that connection.

Advantages and Disadvantages

OQR protocol uses an on-demand resource reservation scheme and hence produces lower control overhead. Since it uses the CDMA-over-TDMA channel model, the network needs to be fully synchronized. Further, the on-demand nature of route discovery process leads to higher connection setup time.

10.5.8 On-Demand Link-State Multipath QoS Routing Protocol

Unlike the QoS routing protocols described above in this chapter which try to find a single path from the source to the destination satisfying the QoS requirements, the on-demand link-state multipath QoS routing (OLMQR) protocol [21] searches for multiple paths which collectively satisfy the required QoS. The original bandwidth requirement is split into sub-bandwidth requirements. Notably, the paths found by the multipath routing protocol are allowed to share the same sub-paths. OLMQR has better call acceptance rate in ad hoc wireless networks where finding a single path satisfying all the QoS requirements is very difficult.

In this protocol, the MAC layer is assumed to be using the CDMA-over-TDMA channel model similar to BR and OQR protocols. A mobile node in the network knows the bandwidth available to each of its neighbors. When the source node requires a QoS session with bandwidth BW to the destination, it floods a QoS route request (QRREQ) packet. Each packet carries the path history and link-state information from the source to the destination. The destination node collects all possible link-state information from different QRREQ packets received and constructs its own view of the current network topology. A multipath routing algorithm is applied at the destination to determine multiple paths which collectively fulfill the original bandwidth requirement BW of the QoS flow. Then the destination node sends reply packets along these paths, which reserve the corresponding resources (sub-bandwidth requirements) on the corresponding paths on their way back to

the source. The operation of this protocol consists of three phases: Phase 1 is on-demand link-state discovery, phase 2 is unipath discovery, and phase 3 is multipath discovery and reply.

On-Demand Link-State Discovery

For each call request, the source node floods a QRREQ packet toward the destination. Each packet records the path history and all link-state information along its route. A QRREQ packet contains the following fields: source ID, destination ID, node history, free time-slot list, bandwidth requirement, and time to live (TTL). The node history field records the path from source to the current traversed node, the free time-slot list field contains a list of free time-slots of links, where each entry in the list records free time-slots between the current traversed node and the last node recorded in the node history, and TTL field limits the hop length of the search path.

The source S floods a QRREQ(S, D, node history = $\{S\}$, free time-slot list = ϕ, BW, TTL) packet into the network toward the destination D, if the given requirement is BW. An intermediate node N receiving a QRREQ packet performs the following operations:

1. Node N checks the node history field of the QRREQ packet for its address. If it is present, the node discards this QRREQ packet.

2. Otherwise,

   - Node N decrements TTL by one. If TTL counts down to zero, it discards this QRREQ packet.
   - Node N adds itself into the node history field, appends the free time-slots of the link between itself and the last node recorded in the node history field into the free time-slot list field, and rebroadcasts this QRREQ packet.

The destination may receive many different QRREQ packets from the source. It constructs its own view of the current network topology. It also calculates the available bandwidths of the links present in that network topology. For example, consider the network shown in Figure 10.11. The source S floods the network with a QRREQ packet by setting BW and TTL fields to 3 and 4, respectively. The destination D receives six QRREQ packets, which have traversed along the paths: $S \rightarrow A \rightarrow B \rightarrow D$, $S \rightarrow E \rightarrow F \rightarrow D$, $S \rightarrow A \rightarrow C \rightarrow B \rightarrow D$, $S \rightarrow A \rightarrow C \rightarrow F \rightarrow D$, $S \rightarrow E \rightarrow C \rightarrow F \rightarrow D$, and $S \rightarrow E \rightarrow C \rightarrow B \rightarrow D$. Using this information, a partial view of the network is constructed at the destination D.

Unipath Discovery

Unlike the BR [18] and the OQR [20] protocols discussed earlier in this section, here the unipath discovery operation (*i.e.*, path bandwidth calculation algorithm) does not follow the traditional hop-by-hop approach to determine the end-to-end path

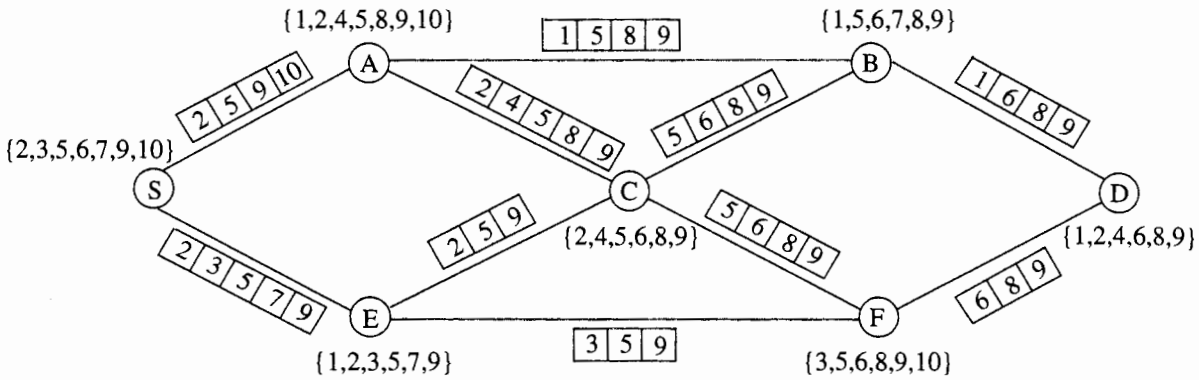


Figure 10.11. An example network.

bandwidth. The unipath discovery approach acquires higher end-to-end path bandwidth than that acquired through the hop-by-hop approach. For a given path (*i.e.*, unipath), the unipath discovery operation determines its maximum path bandwidth by constructing a least-cost-first time-slot reservation tree T_{LCF}. Before constructing T_{LCF}, a time-slot reservation tree T is constructed. The T_{LCF} and T trees are used to reserve time-slots efficiently for a given unipath.

A time-slot reservation tree T is constructed by the breadth-first-search approach as follows. Given a path $S \rightarrow A \rightarrow B \cdots K \rightarrow D$, let the root of T be represented as $abcd \cdots xy$, where a represents the bandwidth (*i.e.*, the set of free time-slots) of link(S, A) and b represents the bandwidth of link(A, B). Let $\underline{ab}cd \cdots xy$ denote the time-slots that are reserved on links a and b. Child nodes of the root are $\underline{ab}cd \cdots xy$, $a\underline{bc}d \cdots xy$, $ab\underline{cd} \cdots xy, \cdots$, and $abcd \cdots \underline{xy}$, which form the first level of tree T. The tree T recursively expands all child nodes of each node on each level of tree T, and follows the same rules as that of the first level of tree T until the leaf nodes are reached. Each path from the root to the leaf nodes gives a time-slot reservation pattern. This pattern is used to reserve time-slots from the source to the destination. To reduce the time needed to search a path satisfying a given bandwidth requirement BW, a least-cost-first time-slot reservation tree T_{LCF} is constructed from the time-slot reservation tree T as follows. To obtain the T_{LCF}, the child nodes on each level of tree T are sorted in ascending order from left to right by using the number of reserved time-slots in them. The unipath time-slot reservation algorithm performs depth-first-search on the T_{LCF} tree to determine a time-slot reservation pattern having maximum path bandwidth. The search is completed if either the tree traversal is completed or a reservation pattern is identified with a bandwidth $\bar{B}$, where $\bar{B} \geq BW$.

For example, consider the path $S \rightarrow A \rightarrow B \rightarrow D$ from the source S to the destination D in the network shown in Figure 10.11. Let a, b, c denote free time-slots of links (S, A), (A, B), and (B, D), respectively, as shown in Figure 10.12 (a). For this path, a time-slot reservation tree T can be constructed as shown in Figure 10.12 (b). It shows two reservation patterns: The first pattern is $\underline{ab}$, $\underline{c}$ and the second pattern is $\underline{bc}$, $\underline{a}$. In the first pattern, $\underline{ab}$ has three time-slots bandwidth

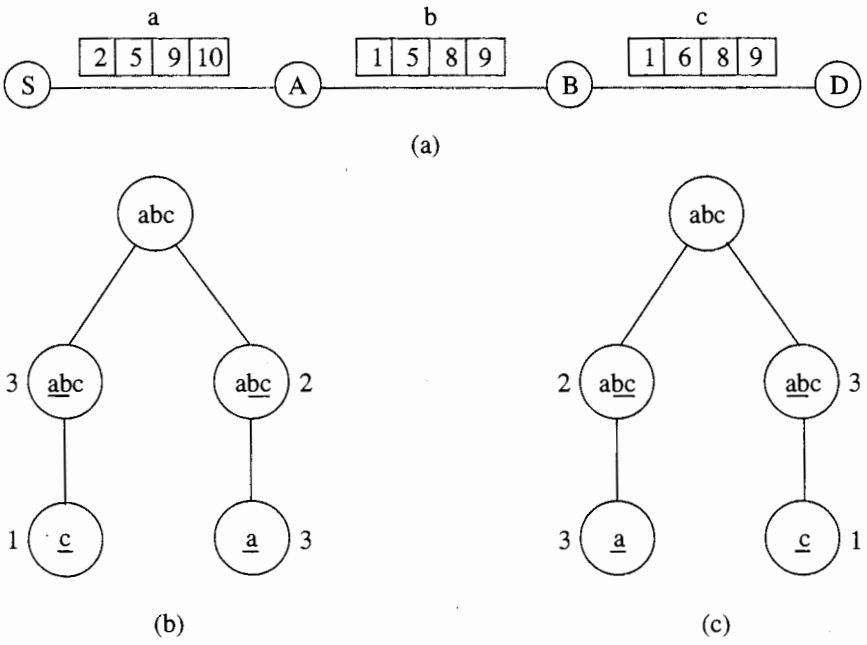


Figure 10.12. Example of T and T_{LCF} trees for a path.

(by assigning slots 2, 5, and 10 for the link a and slots 1, 8, and 9 for the link b) and $\underline{c}$ has one time-slot bandwidth (by assigning the remaining slot 6 for the link c). Hence, the first pattern $\underline{ab}$, $\underline{c}$ has one time-slot path bandwidth (which is the minimum of bandwidths of $\underline{ab}$ and $\underline{c}$). Similarly, in the second pattern, $\underline{bc}$ has two time-slots bandwidth (by assigning slots 1 and 5 for the link b and slots 6 and 8 for the link c), and $\underline{a}$ has three time-slots bandwidth (by assigning the remaining slots 2, 9, and 10 for the link a). Hence, the second pattern $\underline{bc}$, $\underline{a}$ has two time-slots path bandwidth. From T, a least-cost-first time-slot reservation tree T_{LCF} can be constructed as shown in Figure 10.12 (c). Comparing the T-tree traversal with the T_{LCF}-tree traversal scheme, the T_{LCF}-tree traversal scheme is more efficient than the T-tree traversal scheme as it reduces the time required to find a feasible QoS path.

Multipath Discovery and Reply

The destination initiates the multipath discovery operation by sequentially exploiting multiple unipaths such that the sum of path bandwidths fulfills the original bandwidth requirement BW. The destination applies the unipath discovery operation to each path in order to determine the maximum achievable path bandwidth of each path. After accepting a path, the destination updates the network state information it maintains in order to reflect the current bandwidth availability on the links. Finally, the destination sends reply packets along these paths, which reserve the corresponding resources (sub-bandwidth requirements) on the corresponding paths on their way back to the source. In the above example, the destination D

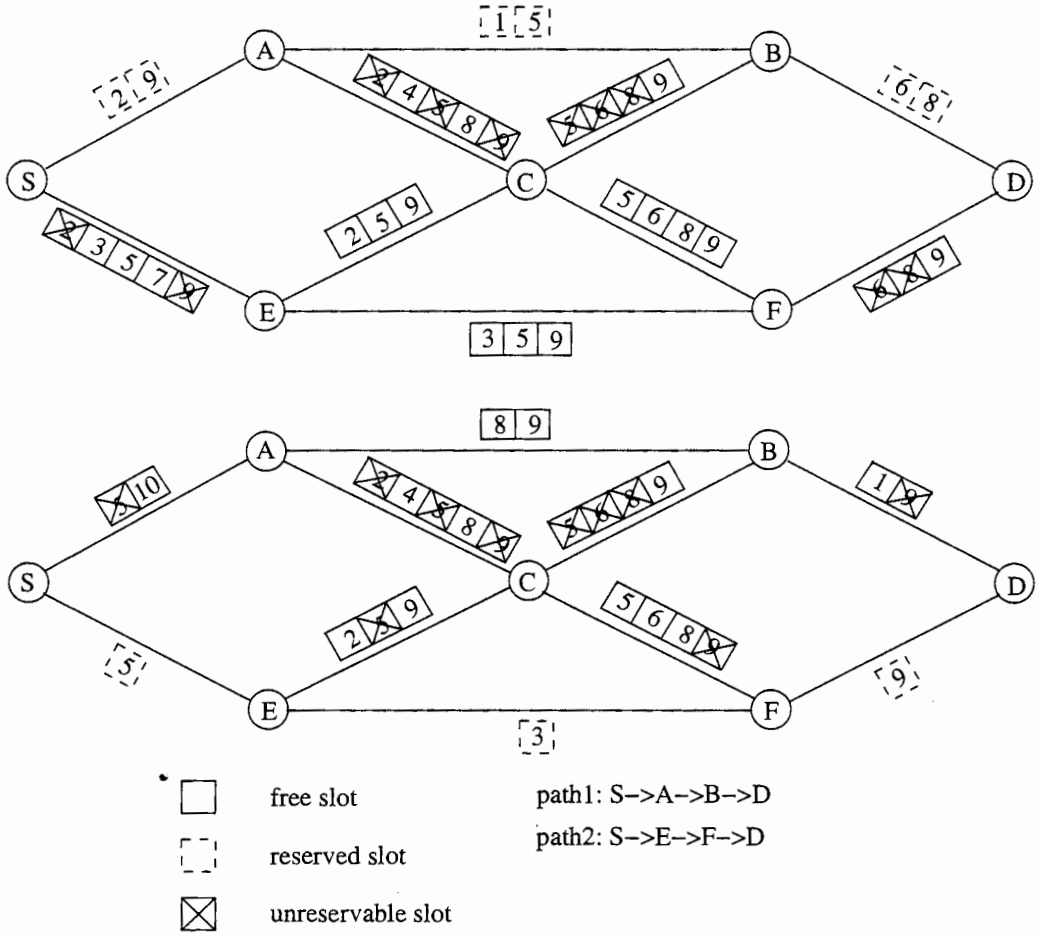


Figure 10.13. The unipaths found by multipath discovery algorithm.

finds two unipaths: $S \rightarrow A \rightarrow B \rightarrow D$ with two time-slots path bandwidth and $S \rightarrow E \rightarrow F \rightarrow D$ with one time-slot path bandwidth, as shown in Figure 10.13.

Advantages and Disadvantages

If the QoS requirements of a flow cannot be met by a single path from the source to the destination, multiple paths are checked which collectively satisfy the required QoS. Hence, OLMQR protocol has better ACAR. But the overhead of maintaining and repairing paths is very high compared to traditional unipath routing protocols because multiple paths are used to satisfy each flow's QoS requirements.

10.5.9 Asynchronous Slot Allocation Strategies

The QoS solutions discussed so far such as BR, OQR, and OLMQR assume a TDMA-based network or a CDMA-over-TDMA model for the network. This requires time synchronization across all nodes in the network. Time synchronization demands periodic exchange of control packets, which results in high bandwidth consumption. Ad hoc wireless networks experience rapid changes in topology leading

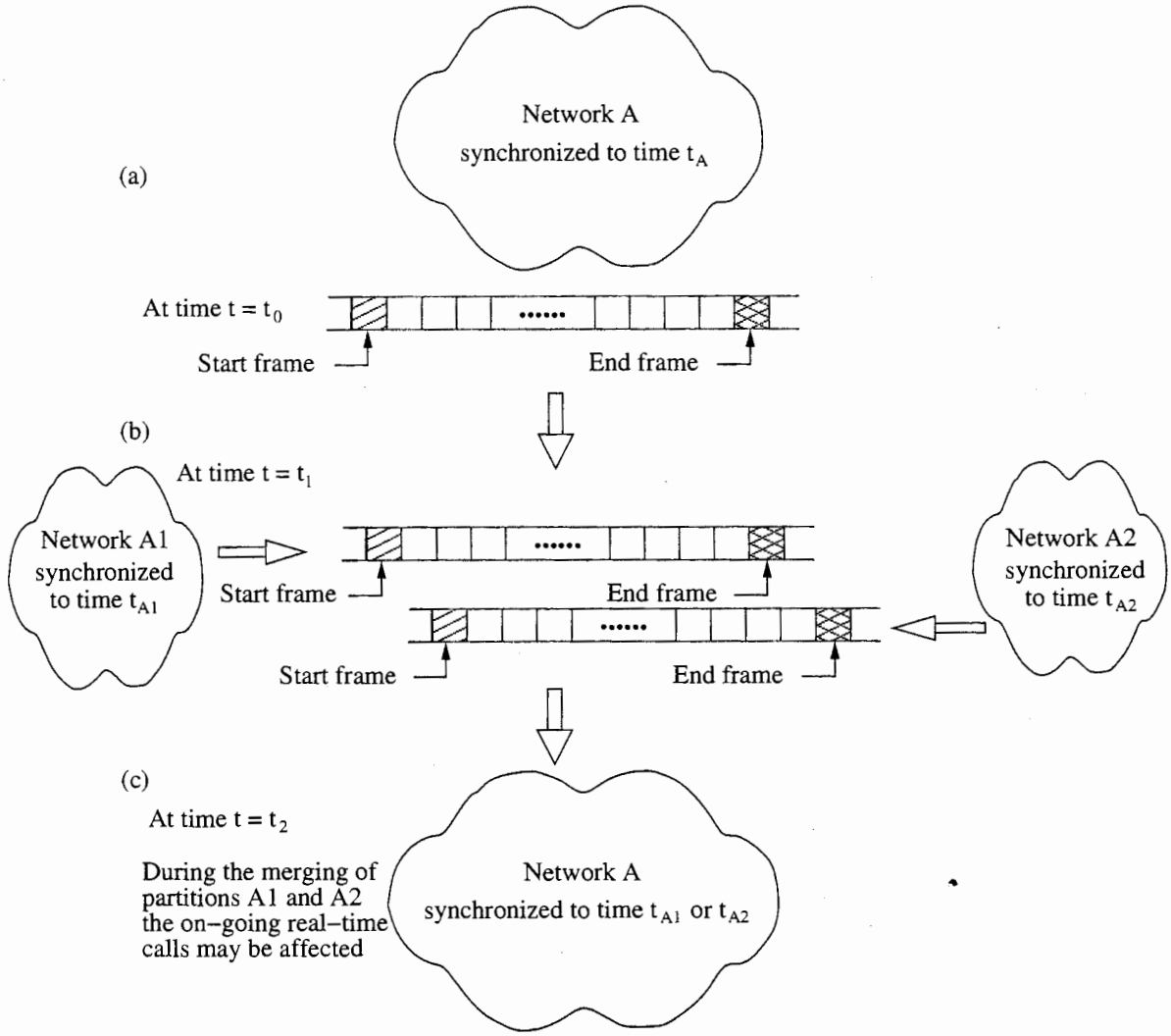


Figure 10.14. Illustration of synchronization problems in a dynamic network topology.

to a situation where network partitions and merging of partitions can take place. Figure 10.14 shows the synchronization problems arising out of dynamic topological changes in an ad hoc wireless network.

A completely connected and synchronized network A at time $t = t_0$ (shown in Figure 10.14 (a)) may be partitioned into two disjoint networks A1 and A2 at time $t = t_1$ (shown in Figure 10.14 (b)). These two networks may be synchronized to two different clock times as illustrated. Due to the dynamic topology experienced in an ad hoc wireless network, it is possible to have two separately synchronized networks A1 (synchronized to t_{A1}) and A2 (synchronized to t_{A2}) merge to form a combined network A (Figure 10.14 (c)). During the merging process, the real-time calls existing in the network may be affected while accommodating the changes in synchronization.

The asynchronous QoS routing (AQR) scheme and slot allocation strategies proposed in [22], [23] provide a unique mechanism to reserve asynchronous end-to-end bandwidth for real-time calls in ad hoc wireless networks. These strategies utilize the real-time MAC (RTMAC) [13] protocol that can effect bandwidth reservation in asynchronous ad hoc wireless networks. RTMAC is explained in detail in Section 6.6.7. RTMAC can reserve conn-slots [number of reservation slots (minimum time duration that can be reserved) sufficient for a real-time session] on a superframe (time duration in which the existing reservations repeat). AQR is an extension of dynamic source routing (DSR) protocol discussed in Section 7.5.1. The three major phases in the operation of AQR are bandwidth feasibility test phase, bandwidth allocation phase, and bandwidth reservation phase. An in-depth discussion of each of these phases follows.

Bandwidth Feasibility Test Phase

The objective of this phase is the selection of paths with required bandwidth, which is achieved by the propagation of *RouteRequest* packets. The source node that needs to set up a QoS path to a destination originates *RouteRequest* packets addressed to the destination. An intermediate node that receives this *RouteRequest* checks for bandwidth availability in the link through which it received the *RouteRequest* packet. AQR interacts with the MAC layer for obtaining reservation information. If sufficient bandwidth is available, then it forwards the *RouteRequest* packet, else the packet is dropped. The intermediate node adds its own reservation table along with the reservation tables of the nodes the packet has already traversed before forwarding it further. Routing loops are avoided by keeping track of the sequence number, source address, and traversed path informations contained in the *RouteRequest* packet. Apart from this reservation table, an intermediate node also incorporates necessary information in an *offset time* field to enable the destination node to make use of the reservation table. In other words, the offset time field carries synchronization information required for interpreting the reservation table with respect to the receiving node's current time. When the source node constructs a *RouteRequest* packet, it stores its reservation table in the packet with respect to its current time with the quantity offset set to zero. When the packet is about to be sent, the difference between the current time and time of construction of packet is stored in the offset. When the *RouteRequest* packet is received at a node, the offset is increased by the estimated propagation delay of transmission. Hence by using this offset time, the relative difference between the local clock and the time information contained in the reservation table carried in the *RouteRequest* can be incorporated and then used for synchronizing the reservation information. When the *RouteRequest* packet reaches its destination, it runs the slot allocation algorithm on a selected path, after constructing a data structure called *QoS Frame* for every link in that path. The *QoS Frame* is used to calculate, for every link, the free bandwidth slots in the superframe and unreservable slots due to reservations carried out by the neighborhood nodes (also referred to as unreservable slots due to hidden terminals). The destination node waits for a specific time interval, gathers a set of *RouteRequest* packets, and chooses a shortest path with necessary bandwidth.

Bandwidth Allocation Phase

In this phase, the destination node performs a bandwidth allocation strategy that assigns free slots to every intermediate link in the chosen path. The information about asynchronous slots assigned at every intermediate link is included in the *RouteReply* packet and propagated through the selected path back to the source. Slot allocation strategies such as early fit reservation (EFR), minimum bandwidth-based reservation (MBR), position-based hybrid reservation (PHR), and k-hopcount hybrid reservation (k-HHR) discussed later in this section can be used for allocation of bandwidth and positioning of slots in this phase.

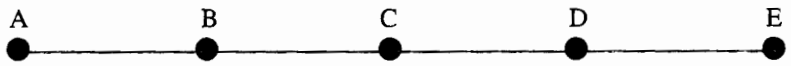
Slot Allocation Strategies

The slot allocation strategies are used in the bandwidth allocation phase in order to decide upon the order of links in a chosen path and particular slot positions to be assigned. The order of links chosen for allocation and the position of assigned bandwidth-slots on each link influence the end-to-end delay of the path and the call acceptance rate.

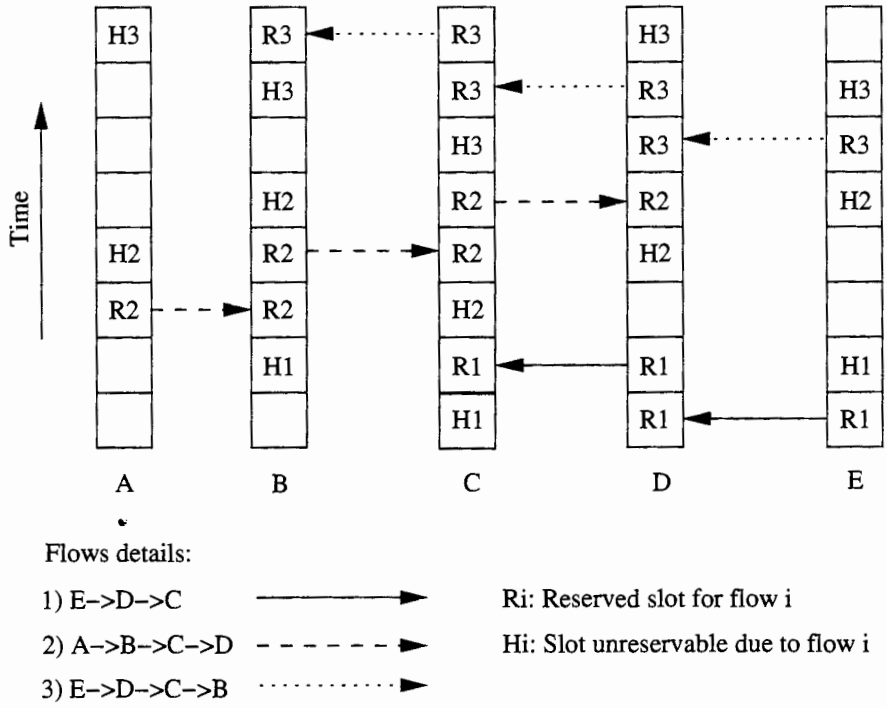
- Early fit reservation (EFR): During the bandwidth allocation phase, the destination node runs the following steps for the EFR scheme:
  - Step 1: Order the links in the path from source to destination.
  - Step 2: Allocate the first available free slot for the first link in the path.
  - Step 3: For every subsequent link, allocate the first immediate free slot after the assigned slot in the previous link.
  - Step 4: Continue Step 3 until the last link in the chosen path is reached.

  EFR attempts to provide the least end-to-end delay. The average end-to-end delay can be obtained as $(n-1) \times \frac{t_{sf}}{2}$ where n is the number of hops in the path and t_{sf} is the duration of the superframe. Figure 10.15 (a) illustrates a simple string topology and Figure 10.15 (b) shows the slot allocation carried out for three real-time flows. In the example, the average delay experienced can be calculated as $\frac{8}{3}$ slots. The flow $E \rightarrow C$ experiences a delay of two slots, and flows $A \rightarrow D$ and $E \rightarrow B$ experience a delay of three slots each, making the average delay of $\frac{8}{3}$ slots.

- Minimum bandwidth-based reservation (MBR): The following steps are executed by the destination node for the MBR scheme:
  - Step 1: Order the links in the non-decreasing order of free bandwidth.
  - Step 2: Allocate the first free slot in the link with lowest free bandwidth.
  - Step 3: Reorder the links in the non-decreasing order of free bandwidth and assign the first free slot on the link with lowest bandwidth.
  - Step 4: Continue Step 3 until bandwidth is allotted for all the links.



(a) Example network topology



(b) EFR scheme

Figure 10.15. Illustration of EFR scheme. *Reproduced with permission from [23], © John Wiley & Sons Limited, 2004.*

MBR allots bandwidth for the links in the increasing order of free bandwidth. In case a tie occurs, where two links exist with the same amount of free bandwidth, it is broken by choosing the link with lowest bandwidth in the neighboring links. Further ties are broken by choosing the link with lowest ID of the link-level sender. Figure 10.16 (b) shows the slot allocation carried out in the MBR scheme over a simple string topology network. The worst case end-to-end delay provided by MBR can be $(n-1) \times t_{sf}$. In the example in Figure 10.16 (b), the average delay experienced can be calculated as $\frac{33}{3}$ slots.

- Position-based hybrid reservation (PHR): Similar to EFR and MBR schemes, PHR also is executed at the destination node. The following are the steps in the PHR algorithm:

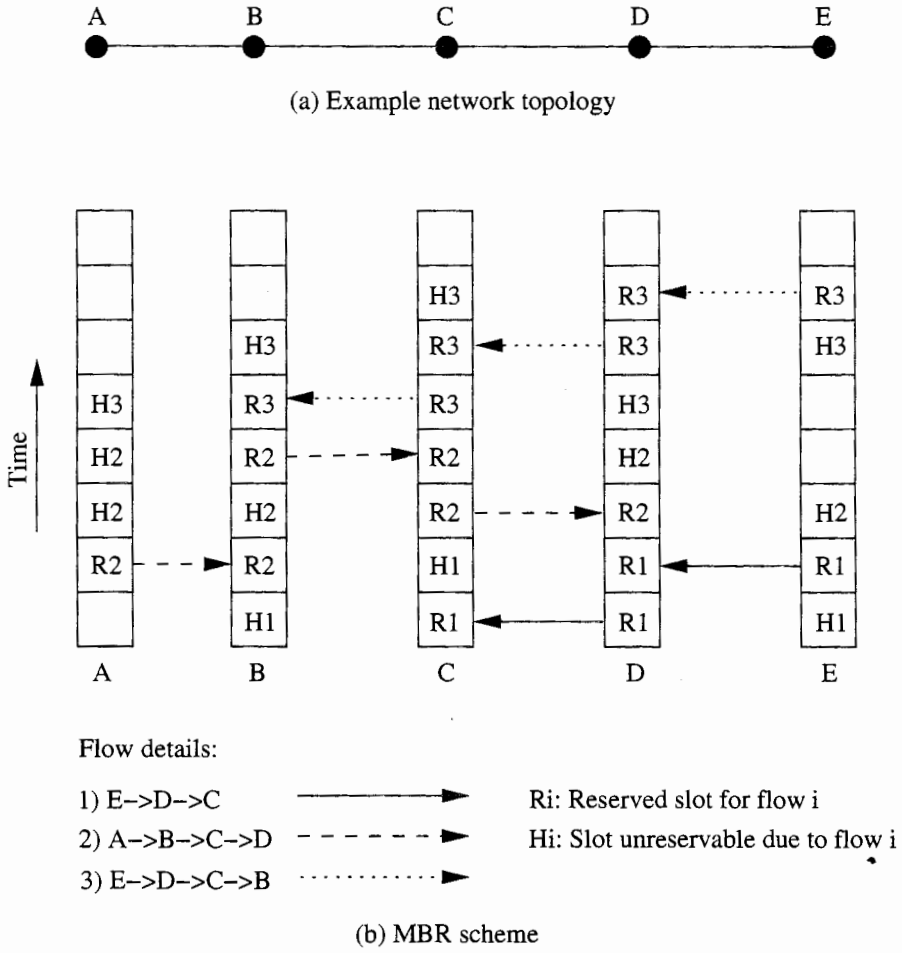


Figure 10.16. Illustration of MBR scheme. *Reproduced with permission from [23], © John Wiley & Sons Limited, 2004.*

- Step 1: List the links in the order of increasing bandwidth.
- Step 2: Assign a free slot for the link with least amount of bandwidth, such that the position of assignment of bandwidth is proportional to $\frac{i}{L_{path}}$ where i is the position of the link and L_{path} is the path length.
- Step 3: Repeat Step 2 until all the links are assigned with free slots.

Figure 10.17 shows the slot allocation done on a string topology for three flows. In the given example, the average delay experienced can be calculated as $\frac{18}{3}$ slots.

- k-hopcount hybrid routing (k-HHR): This is a hybrid slot allocation scheme in which either EFR or PHR is chosen dynamically by the destination node based on the hop length of the path. The k-HHR scheme is described below.

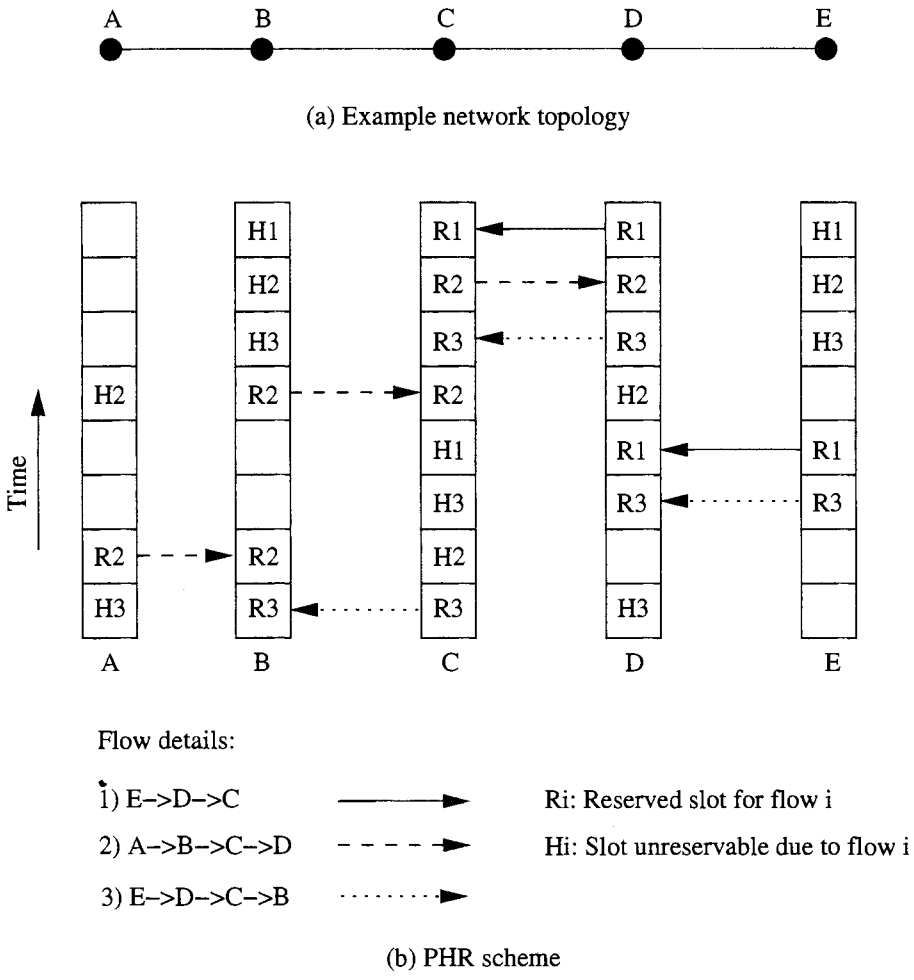


Figure 10.17. Illustration of PHR scheme. *Reproduced with permission from [23], © John Wiley & Sons Limited, 2004.*

if $(pathlength > k)$
    Use EFR for slot allocation
else
    Use PHR for slot allocation

This takes the end-to-end delay advantage of the EFR scheme for long flows and the high call acceptance with medium end-to-end delay of the PHR scheme for flows with shorter length.

Bandwidth Reservation Phase

In this phase, a reservation of bandwidth at every link of a path is carried out. The reservation is effected by the intermediate nodes with the information carried in the *RouteReply* packet, in an asynchronous fashion using RTMAC protocol. Once the reservation at an intermediate link is successful in the designated time duration (the

time duration for a free conn-slot, at which the reservation is to be carried out), the *RouteReply* packet is further forwarded. If the designated slot is not free at the time the intermediate node attempts the reservation (this can happen either due to the mobility of nodes or due to the staleness of the information), the intermediate node can try reserving any of the free slots available. If the intermediate node finds it impossible to reserve bandwidth, it drops the *RouteReply* and sends a control packet to the destination, which makes all the nodes in its way, those that have successfully reserved bandwidth, release the bandwidth and the destination node to find another path with the necessary bandwidth.

Advantages and Disadvantages

AQR has a unique advantage in that it can provide end-to-end bandwidth reservation in asynchronous networks. Also, the slot allocation strategies can be used to plan for the delay requirements and dynamically choose appropriate algorithms. AQR is an on-demand QoS routing scheme and hence the setup time and reconfiguration time of real-time calls are high. Also, the bandwidth efficiency of such an asynchronous system may not be as high as a fully synchronized TDMA system due to the formation of bandwidth holes (short free slots which cannot be used).

10.6 QOS FRAMEWORKS FOR AD HOC WIRELESS NETWORKS

A framework for QoS is a complete system that attempts to provide required/promised services to each user or application. All components within this system cooperate in providing the required services.

The key component of any QoS framework is the QoS service model which defines the way user requirements are met. The key design issue here is whether to serve users on a per session basis or on a per class basis. Each class represents an aggregation of users based on certain criteria. The other key components of the framework are QoS routing which is used to find all or some of the feasible paths in the network that can satisfy user requirements, QoS signaling for resource reservation, QoS medium access control, call admission control, and packet scheduling schemes. The QoS modules, namely, routing protocol, signaling protocol, and the resource management mechanism, should react promptly to changes in the network state (topology changes) and flow state (change in the end-to-end view of the service delivered). In what follows, each component's functionality and its role in providing QoS in ad hoc wireless networks will be described.

- *Routing protocol:* Similar to the QoS routing protocols, discussed earlier in this chapter, the routing protocol module in any QoS framework is used to find a path from the source to the destination and to forward the data packet to the next intermediate relay node. QoS routing describes the process of finding suitable path(s) that satisfy the QoS service requirements of an application. If multiple paths are available, the information regarding such paths helps to restore the service quickly when the service becomes disturbed due

to a path break. The performance of the routing protocol, in terms of control overhead, affects the performance of the QoS framework. The routing protocol should be able to track changes in the network topology with minimum control overhead. The routing protocol needs to work efficiently with other components of the QoS framework such as signaling, admission control, and resource management mechanisms in order to provide end-to-end QoS guarantees. These mechanisms should consume minimal resources in operation and react rapidly to changes in the network state and the flow state.

- *QoS resource reservation signaling:* Once a path with the required QoS is found, the next step is to reserve the required resources along that path. This is done by the resource reservation signaling protocol. For example, for applications that require certain minimum bandwidth guarantees, signaling protocol communicates with the medium access control subsystem to find and reserve the required bandwidth. On completion/termination of a session, the previously reserved resources are released.

- *Admission control:* Even though a QoS feasible path may be available, the system needs to decide whether to actually serve the connection or not. If the call is to be served, the signaling protocol reserves the resources; otherwise, the application is notified of the rejection. When a new call is accepted, it should not jeopardize the QoS guarantees given to the already admitted calls. A QoS framework is evaluated based on the number of QoS sessions it serves and it is represented by the average call acceptance ratio (ACAR) metric. Admission control ensures that there is no perceivable degradation in the QoS being offered to the QoS sessions admitted already.

- *Packet scheduling:* When multiple QoS connections are active at the same time through a link, the decision on which QoS flow is to be served next is made by the scheduling scheme. For example, when multiple delay-constrained sessions are passing through a node, the scheduling mechanism decides on when to schedule the transmission of packets when packets belonging to more than one session are pending in the transmission queue of the node. The performance of a scheduling scheme is reflected by the percentage of packets that meet their deadlines.

10.6.1 QoS Models

A QoS model defines the nature of service differentiation. In wired network QoS frameworks, several service models have been proposed. Two of these models are the integrated services (IntServ) model [25] and the differentiated services (DiffServ) model [26]. The IntServ model provides QoS on a per flow basis, where a flow is an application session between a pair of end users. Each IntServ-enabled router maintains all the flow specific state information such as bandwidth requirements, delay bound, and cost. The different types of services offered in this model are guaranteed service, controlled load service, and best-effort service. The resource reservation protocol (RSVP) [27] is used for reserving the resources along the route.

The volume of information maintained at an IntServ-enabled router is proportional to the number of flows. Hence, the IntServ model is not scalable for the Internet, but it can be applied to small-sized ad hoc wireless networks. However, per flow information is difficult to maintain precisely at a node in an ad hoc wireless network due to reasons such as limited processing capability, limited battery energy, frequent changes in network topology, and continuously varying link capacity due to the time-varying characteristics of radio links. The DiffServ model was proposed in order to overcome the difficulty in implementing and deploying IntServ model and RSVP in the Internet. In this model, flows are aggregated into a limited number of service classes. Each flow belongs to one of the DiffServ classes of service. This solved the scalability problem faced by the IntServ model.

The above two service models cannot be directly applied to ad hoc wireless networks because of the inherent characteristics of ad hoc wireless networks such as continuously varying network topology, limited resource availability, and error-prone shared radio channel. Any service model proposed should first decide upon what types of services are feasible in such networks. A hybrid service model for ad hoc wireless networks called flexible QoS model for mobile ad hoc networks (FQMM) is described below. This model is based on the above two QoS service models.

Flexible QoS Model for Mobile Ad Hoc Networks

The flexible QoS model for mobile ad hoc networks (FQMM) [28] takes advantage of the per flow granularity of IntServ and aggregation of services into classes in DiffServ. In this model the nodes are classified into three different categories, namely, *ingress* node (source), *interior* node (intermediate relay node), and *egress* node (destination) on a per flow basis. A source node, which is the originator of the traffic, is responsible for traffic-shaping. Traffic-shaping is the process of delaying packets belonging to a flow so that packets conform to a certain defined traffic profile. The traffic profile contains a description of the temporal properties of a flow such as its mean rate (*i.e.*, the rate at which data can be sent per unit time on average) and burst size (which specifies in bits per burst how much traffic can be sent within a given unit of time without creating scheduling concerns). The FQMM model provides per flow QoS services for the high-priority flows while lower priority flows are aggregated into a set of service classes as illustrated in Figure 10.18. This hybrid service model is based on the assumption that the percentage of flows requiring per flow QoS service is much less than that of low-priority flows which can be aggregated into QoS classes. Based on the current traffic load in the network, the service level of a flow may change dynamically from per flow to per class and vice versa.

Advantages and Disadvantages

FQMM provides the ideal per flow QoS service and overcomes the scalability problem by classifying the low-priority traffic into service classes. This protocol addresses the basic problem faced by QoS frameworks and proposes a generic solution

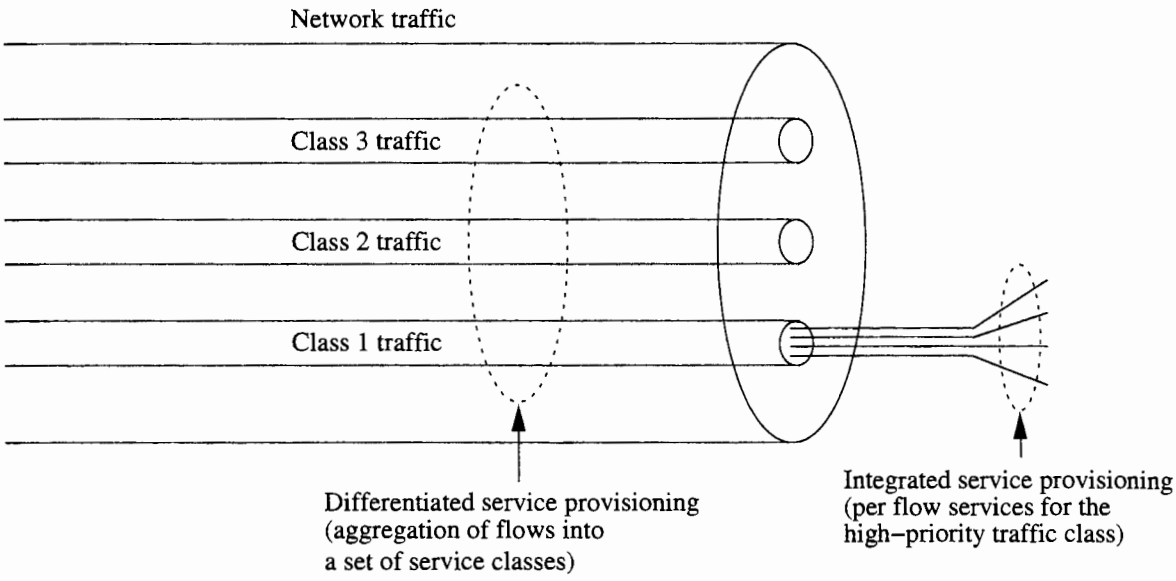


Figure 10.18. FQMM model.

for ad hoc wireless networks that can be a base for a better QoS model. However, several issues still remain unresolved, such as decision upon traffic classification, allotment of per flow or aggregated service for the given flow, amount of traffic belonging to per flow service, the mechanisms used by the intermediate nodes to get information regarding the flow, and scheduling or forwarding of the traffic by the intermediate nodes.

10.6.2 QoS Resource Reservation Signaling

The QoS resource reservation signaling scheme is responsible for reserving the required resources and informing the corresponding applications, which then initiate data transmission. Signaling protocol consists of three phases, namely, connection establishment, connection maintenance, and connection termination. On establishing a connection, it monitors the path and repairs/reconfigures it if the connection suffers from any violation in its QoS guarantees. On completion/termination of a session, it releases the resources that had been reserved for that session. In the wired networks, the RSVP protocol [27] is used for resource reservation, but it cannot be applied directly to ad hoc wireless networks due to the following reasons:

- The amount of control overhead generated during the connection maintenance phase of RSVP signaling is too heavy for bandwidth-constrained ad hoc wireless networks.

- It is not adaptive to network dynamics. In wired networks, once the resources are reserved, they are assumed to be available to applications throughout the

session. But these assumptions are not true in ad hoc wireless networks due to the unrestricted mobility of nodes, which results in dynamic changes in the network topology.

MRSVP: A Resource Reservation Protocol for Cellular Networks

The MRSVP [29], as discussed in Chapter 4, is an extension of RSVP protocol for mobile hosts. It allows a mobile host to connect to the network through different points (base stations) in the course of time. It is basically proposed as an extension of RSVP for cellular networks to integrate them with the IP network. It is assumed that a mobile host predicts precisely the set of locations that the host is expected to visit during the lifetime of the flow. This information is provided in the form of mobility specifications to the network, so that reservations are made before that host uses the paths. The protocol proposes two types of reservations: *active* and *passive*. The reservation made over a path for a QoS flow is said to be active if data packets currently flow along that path. A reservation is said to be passive if the path on which resources have been reserved is to be used only in the future. Resources that are reserved passively for a flow can be utilized by other flows that require best-effort service.

MRSVP employs *proxy agents* (just as home agents and foreign agents in mobile IP protocol) to reserve resources along the path from the sender to the locations in the mobility specification of the mobile host. The *local proxy agent* (*i.e.*, the *proxy agent* present at the current location of the mobile host) makes an *active* reservation from the sender to the mobile host. The *remote proxy agents* (*i.e.*, *proxy agents* present at other locations in the mobility specification of the mobile host) make *passive* reservations on behalf of the mobile host.

Limitations of Adapting MRSVP for Ad Hoc Wireless Networks

MRSVP requires the future locations of mobile hosts in advance. Obtaining such location information is extremely difficult in ad hoc wireless networks because of the unrestricted mobility of the mobile hosts. Due to this reason, passive reservations fail in the case of ad hoc wireless networks. Secondly, even if future locations are known, finding a path and reserving resources on that path in advance may not be a viable and efficient solution. This inefficiency is because of the random and unpredictable movement of the intermediate nodes. It is also unknown which nodes should act as proxy agents due to the lack of infrastructure support in ad hoc wireless networks.

10.6.3 INSIGNIA

The INSIGNIA QoS framework [30] was developed to provide adaptive services in ad hoc wireless networks. Adaptive services support applications that require only a minimum quantitative QoS guarantee (such as minimum bandwidth) called *base QoS*. The service level can be extended later to *enhanced QoS* when sufficient resources become available. Here user sessions adapt to the available level of service

without explicit signaling between the source-destination pairs. The key design issues in providing adaptive services are as follows:

- How fast can the application service level be switched from *base QoS* to *enhanced QoS* and vice versa in response to changes in the network topology and channel conditions?
- How and when is it possible to operate on the *base QoS* or *enhanced QoS* level for an adaptive application (*i.e.*, an application that can sustain variation in QoS levels)?

This framework can scale down, drop, or scale up user sessions adaptively based on network dynamics and user-supplied adaptation policies. A key component of this framework is the INSIGNIA in-band signaling system, which supports fast reservation, restoration, and adaptation schemes to deliver the adaptive services. The signaling system is light-weight and responds rapidly to changes in the network topology and end-to-end QoS conditions. As depicted in Figure 10.19, the INSIGNIA framework has the following key components for supporting adaptive real-time services:

- *Routing module:* The routing protocol finds a route from the source to the destination. It is also used to forward a data packet to the next intermediate

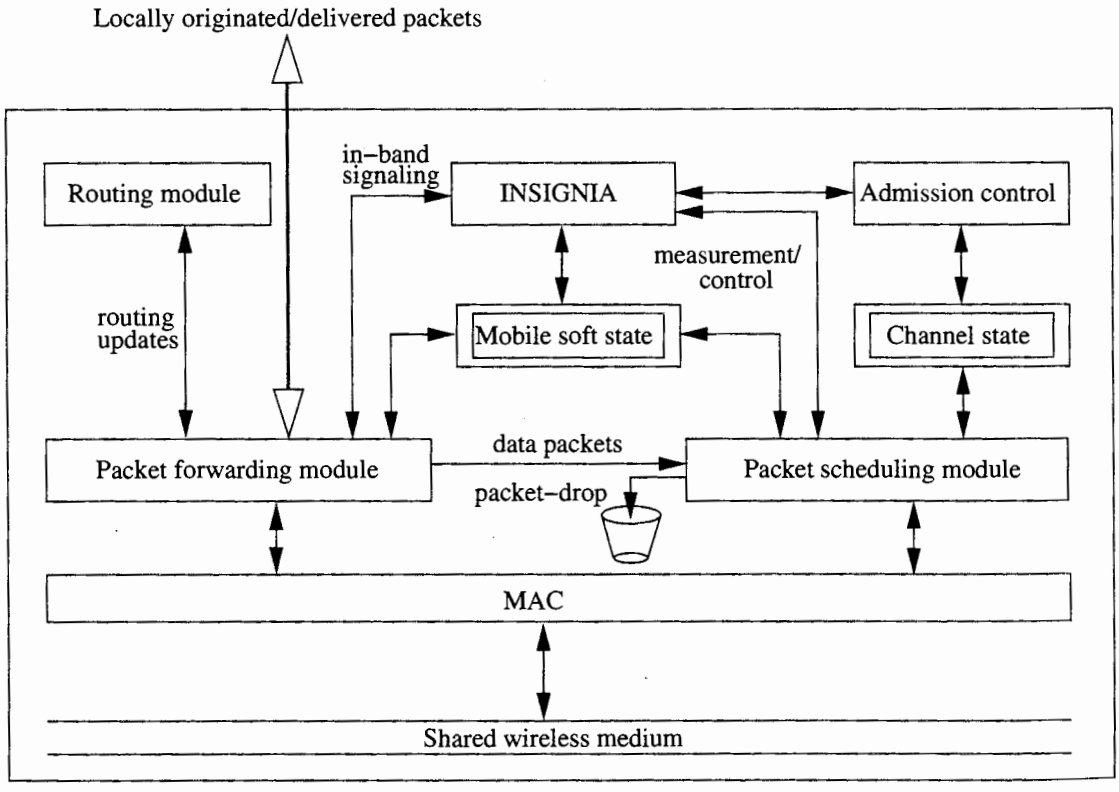


Figure 10.19. INSIGNIA QoS framework.

relay node. The routing module is independent of other components and hence any existing routing protocol can be used. INSIGNIA assumes that the routing protocol provides new routes in case of topology changes.

- *In-band signaling:* This module is used to establish, adapt, restore, and tear down adaptive services between source-destination pairs. It is not dependent on any specific link layer protocol. In in-band signaling systems, the control information is carried along with data packets and hence no explicit control channel is required. In the INSIGNIA framework, each data packet contains an optional QoS field (INSIGNIA option) to carry the control information. The signaling information is encoded into this optional QoS field. The in-band signaling system can operate at speeds close to that of packet transmissions and is therefore better suited for highly dynamic mobile network environments.

- *Admission control:* This module allocates bandwidth to flows based on the maximum/minimum bandwidth requested. Once the bandwidth is reserved, the reservation must be refreshed periodically by a soft state mechanism. Typically, the reception of data packets refreshes the reservations done.

- *Packet forwarding:* This module classifies the incoming packets and delivers them to the appropriate module. If the current node is the destination of the packet, then the packet is delivered to the local application. If the packet has an INSIGNIA option, it is delivered to the INSIGNIA signaling module. If the destination is some other node, it is relayed to the next intermediate node with the help of the routing and packet-scheduling modules.

- *Packet scheduling:* Packets that are to be routed to other nodes are handled by the packet-scheduling module. The packets to be transmitted by a node are scheduled by the scheduler based on the forwarding policy. INSIGNIA uses a weighted round-robin service discipline.

- *Medium access control (MAC):* The MAC protocol provides QoS-driven access to the shared wireless medium for adaptive real-time services. The INSIGNIA framework is transparent to any underlying MAC protocol.

The INSIGNIA framework uses a soft state resource management mechanism for efficient utilization of resources. When an intermediate node receives a data packet with a RES (reservation) flag set for a QoS flow and no reservation has been done until now, the admission control module allocates the resources based on availability. If the reservation has been done already, it is reconfirmed. If no data packets are received for a specified timeout period, the resources are deallocated in a distributed manner without incurring any control overhead. In setting the value for the timeout period, care should be taken to avoid *false restoration* (which occurs when the time interval is smaller than the inter-arrival time of packets) and resource lock-up (which occurs when the time interval is much greater than the inter-arrival time of packets).

Operation of INSIGNIA Framework

The INSIGNIA framework supports adaptive applications which can be applications requiring best-effort service or applications with *base QoS* requirements or those with *enhanced QoS* requirements. Due to the adaptation of the protocol to the dynamic behavior of ad hoc wireless networks, the service level of an application can be degraded in a distributed manner if enough resources are not available. For example, data packets belonging to the *enhanced QoS* service mode may have to be routed in the *base QoS* or best-effort service modes adaptively due to lack of enough resources along the path. If enough resources become available later during the lifetime of the connection, the application can be upgraded.

The INSIGNIA option field contains the following information: service mode, payload type, bandwidth indicator, and bandwidth request. These indicate the dynamic behavior of the path and the requirements of the application. The intermediate nodes take decisions regarding the flow state in a distributed manner based on the INSIGNIA option field. The service mode can be either best-effort (BE) or service requiring reservation (RES) of resources. The payload type indicates the QoS requirements of the application. It can be either *base QoS* for an application that requires minimum bandwidth, or *enhanced QoS* for an application which requires a certain maximum bandwidth but can operate with a certain minimum bandwidth below which they are useless. Examples of applications that require enhanced service mode are video applications that can tolerate packet loss and delay jitter to a certain extent. The bandwidth indicator flag has a value of MAX or MIN, which represents the bandwidth available for the flow. Table 10.2 shows how service mode, payload type, and bandwidth indicator flags reflect the current status of flows. It can be seen from the table that the best-effort (BE) packets are routed as normal data packets. If QoS is required by an application, it can opt for *base QoS* in which a certain minimum bandwidth is guaranteed. For that application, the bandwidth indicator flag is set to MIN. For *enhanced QoS*, the source sets the bandwidth indicator flag to MAX, but it can be downgraded at the intermediate nodes to MIN; the service mode flag is changed to BE from RES if sufficient bandwidth is not available. The downgraded service can be restored to RES if sufficient bandwidth becomes available. For *enhanced QoS*, the service can

Table 10.2. INSIGNIA flags reflecting the behavior of flows

Service Mode	Payload Type	Bandwidth Indicator	Downgrading	Upgrading
BE	-	-	-	-
RES	Base QoS	MIN	Base QoS → BE	BE → Base QoS
RES	Enhanced QoS (EQoS)	MAX	EQoS → BE EQoS → BQoS	BE → EQoS BQoS → EQoS

be downgraded either to BE service or RES service with *base QoS*. The downgraded *enhanced QoS* can be upgraded later, if all the intermediate nodes have the required (MAX) bandwidth.

The destination nodes actively monitor the on-going flows, inspecting the bandwidth indicator field of incoming data packets and measuring the delivered QoS (*e.g.*, packet loss, delay, and throughput). The destination nodes send QoS reports to the source nodes. The QoS reports contain information regarding the status of the on-going flows.

Releasing Resources in INSIGNIA

In order to release resources, the destination node sends a QoS report to the source so that the intermediate nodes release the extra resources. Assume that a source node transmits an *enhanced QoS* data packet with MAX requirements. If sufficient bandwidth is available, the intermediate nodes reserve the MAX bandwidth. Now assume that sufficient bandwidth is not available at an intermediate node (bottleneck node). Then the bottleneck node changes the bandwidth indicator flag of the incoming data packet from MAX to MIN. In this case, the intermediate nodes (from the source to the bottleneck node) would have allocated extra resources that remain unutilized, while downstream nodes from the bottleneck node allocate resources only for the downgraded service. Upon receiving the incoming data packet with bandwidth indicator flag set to MIN, the destination node sends a QoS report to inform the corresponding source node that the service level of the flow has been degraded. Further, the intermediate nodes receiving this QoS report release the extra unutilized resources.

Route Maintenance

Due to host mobility, an on-going session may have to be rerouted in case of a path break. The flow restoration process must reestablish the reservation as quickly and efficiently as possible. During restoration, INSIGNIA does not preempt resources from the existing flows for admitting the rerouted flows. INSIGNIA supports three types of flow restoration, namely, *immediate restoration*, which occurs when a rerouted flow immediately recovers to its original reservation; *degraded restoration*, which occurs when a rerouted flow is degraded for a period (T) before it recovers to its original reservation; and *permanent restoration*, which occurs when the rerouted flow never recovers to its original reservation.

Advantages and Disadvantages

INSIGNIA framework provides an integrated approach to QoS provisioning by combining in-band signaling, call admission control, and packet scheduling. The soft state reservation scheme used in this framework ensures that resources are quickly released at the time of path reconfiguration. However, this framework supports only adaptive applications, for example, multimedia applications. Since this framework is transparent to any MAC protocol, fairness and the reservation scheme of the MAC protocol have a significant influence in providing QoS guarantees. Also, because this framework assumes that the routing protocol provides new routes in the

case of topology changes, the route maintenance mechanism of the routing protocol employed significantly affects the delivery of real-time traffic. If enough resources are not available because of the changing network topology, the *enhanced* QoS application may be downgraded to *base* QoS or even to best-effort service. As this framework uses in-band signaling, resources are not reserved before the actual data transmission begins. Hence, INSIGNIA is not suitable for real-time applications that have stringent QoS requirements.

10.6.4 INORA

INORA [31] is a QoS framework for ad hoc wireless networks that makes use of the INSIGNIA in-band signaling mechanism and the TORA routing protocol discussed in Section 7.5.3. The QoS resource reservation signaling mechanism interacts with the routing protocol to deliver QoS guarantees. The TORA routing protocol provides multiple routes between a given source-destination pair. The INSIGNIA signaling mechanism provides feedback to the TORA routing protocol regarding the route chosen and asks for alternate routes if the route provided does not satisfy the QoS requirements. For resource reservation, a soft state reservation mechanism is employed. INORA can be classified into two schemes: *coarse feedback scheme* and *class-based fine feedback scheme.*

Coarse Feedback Scheme

In this scheme, if a node fails to admit a QoS flow either due to lack of minimum required bandwidth (BW_{min}) or because of congestion at the node, it sends an out-of-band *admission control failure* (ACF) message to its upstream node. After receiving the ACF message, the upstream node reroutes the flow through another downstream node provided by the TORA routing protocol. If none of its neighbors is able to admit the flow, it in turn sends an ACF message to its upstream node. While INORA is trying to find a feasible path by searching the *directed acyclic graph* (DAG) following admission control failure at an intermediate node, the packets are transmitted as best-effort packets from the source to destination. In this scheme, different flows between the same source-destination pair can take different routes.

The operations of the coarse feedback scheme are explained through the following example. Here a QoS flow is being initiated by the source node S to the destination node D.

1. Let the DAG created by the TORA protocol be as shown in Figure 10.20. Let $S \rightarrow A \rightarrow B \rightarrow D$ be the path chosen by the TORA routing protocol.

2. INSIGNIA tries to establish soft state reservations for the QoS flow along the path. Assume that node A has admitted the flow successfully and node B fails to admit the flow due to lack of sufficient resources. Node B sends an ACF message to node A.

3. Node A tries to reroute the flow through neighbor node Y provided by TORA.

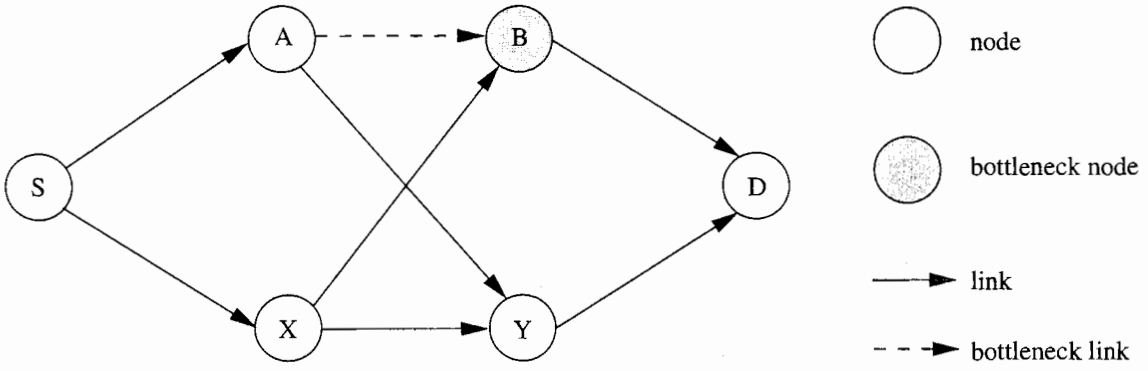


Figure 10.20. INORA coarse feedback scheme: admission control failure at node B.

4. If node Y admits the flow, the flow gets the required reservation all along the path. The new path is $S \rightarrow A \rightarrow Y \rightarrow D$.
5. If node Y fails to admit the flow, it sends an ACF message to node A, which in turn sends an ACF message to node S.
6. Node S tries with its other downstream neighbors to find a QoS path for the flow.
7. If no such neighbor is available, node S rejects the flow.

Class-Based Fine Feedback Scheme

In this scheme, the interval between BW_{min} and BW_{max} of a QoS flow is divided into N classes, where BW_{min} *and* BW_{max} are the minimum and maximum bandwidths required by the QoS flow. Consider a QoS flow being initiated by the source node S to destination node D. Let the flow be admitted with class m ($m < N$).

1. Let the DAG created by the TORA protocol be as shown in Figure 10.21. Let $S \rightarrow A \rightarrow B \rightarrow D$ be the path chosen by the TORA routing protocol.
2. INSIGNIA tries to establish soft state reservations for the QoS flow along the path. Assume that node A has admitted the flow with class m successfully and node B has admitted the flow with bandwidth of class l ($l < m$) only.
3. Node B sends an *Admission Report* message ($AR(l)$) to upstream node A, indicating its ability to give only class l bandwidth to the flow.
4. Node A splits the flow in the ratio of l to $m - l$ and forwards the flow to node B and node Y in that ratio.
5. If node Y is able to give class $(m - l)$ as requested by node A, then the flow of class m is split into two flows, one flow with bandwidth of class l along the path $S \rightarrow A \rightarrow B \rightarrow D$ and the other one with bandwidth of class $(m - l)$ along path $S \rightarrow A \rightarrow Y \rightarrow D$.

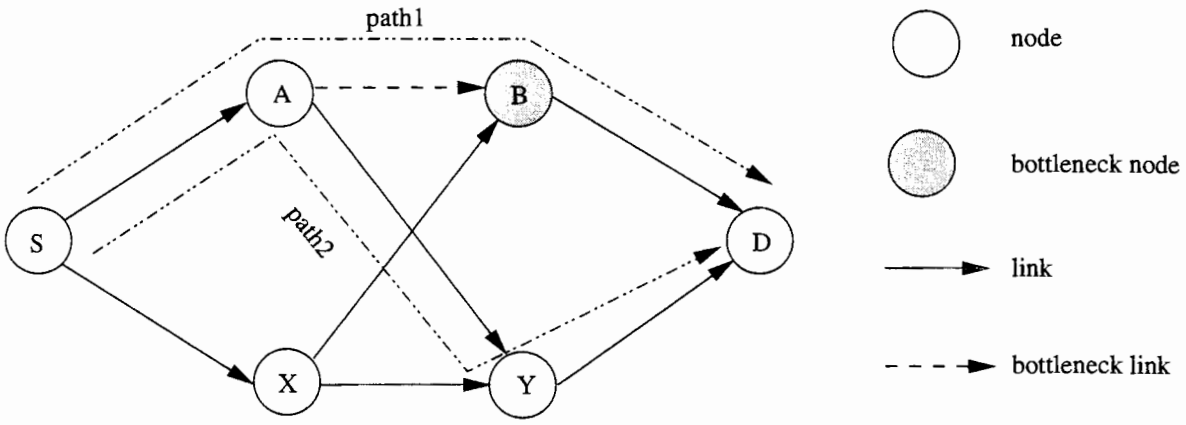


Figure 10.21. INORA fine feedback scheme: node A has admitted the flow with class m, but node B is able to give it class l ($l < m$).

6. If node Y gives only class n ($n < m - l$), it sends an $AR(n)$ message to the upstream node A.

7. Node A, realizing that its downstream neighbors are unable to give class m service, informs of its ability to provide service class of $(l + n)$ by sending an $AR(l + n)$ to node S.

8. Node S tries to find another downstream neighbor which might be able to accommodate the flow with class $(m - (l + n))$.

9. If no such neighbor is available, node S rejects the flow.

Advantages and Disadvantages

INORA is better than INSIGNIA in that it can search multiple paths with lesser QoS guarantees. It uses the INSIGNIA in-band signaling mechanism. Since no resources are reserved before the actual data transmission begins and since data packets have to be transmitted as best-effort packets in case of admission control failure at the intermediate nodes, this model may not be suitable for applications that require hard service guarantees.

10.6.5 SWAN

Ahn *et al.* proposed a distributed network model called stateless wireless ad hoc networks (SWAN) [32] that assumes a best-effort MAC protocol and uses feedback-based control mechanisms to support real-time services and service differentiation in ad hoc wireless networks. SWAN uses a local rate control mechanism for regulating injection of best-effort traffic into the network, a source-based admission control while accepting new real-time sessions, and an explicit congestion notification (ECN) mechanism for dynamically regulating admitted real-time sessions. In this model, intermediate nodes are relieved of the responsibility of maintaining

per-flow or aggregate state information, unlike stateful QoS models such as INSIGNIA and INORA. Changes in topology and network conditions, even node and link failures, do not affect the operation of the SWAN control system. This makes the system simple, robust, and scalable.

SWAN Model

The SWAN model has several control modules which are depicted in Figure 10.22. Upon receiving a packet from the IP layer, the *packet classifier* module checks whether it is marked (*i.e.*, real-time packet) or not (*i.e.*, best-effort packet). If it is a best-effort packet, it is forwarded to the *traffic-shaper* for regulation. If it is a real-time packet, the module forwards it directly to the MAC layer, bypassing the *traffic shaper*. The *traffic shaper* represents a simple leaky bucket traffic policy. The traffic shaper delays best-effort packets in conformance with the rate calculated by the *traffic rate controller*. The *call admission controller* module is responsible for admitting or rejecting new real-time sessions. The decision on whether to admit or reject a real-time session is taken solely by the source node based on the result of an end-to-end request/response probe. The SWAN distributed control algorithms are described in the following sections.

Local Rate Control of Best-Effort Traffic

The SWAN model assumes that most of the traffic existing in the network is best-effort, which can serve as a "buffer zone" or absorber for real-time traffic bursts

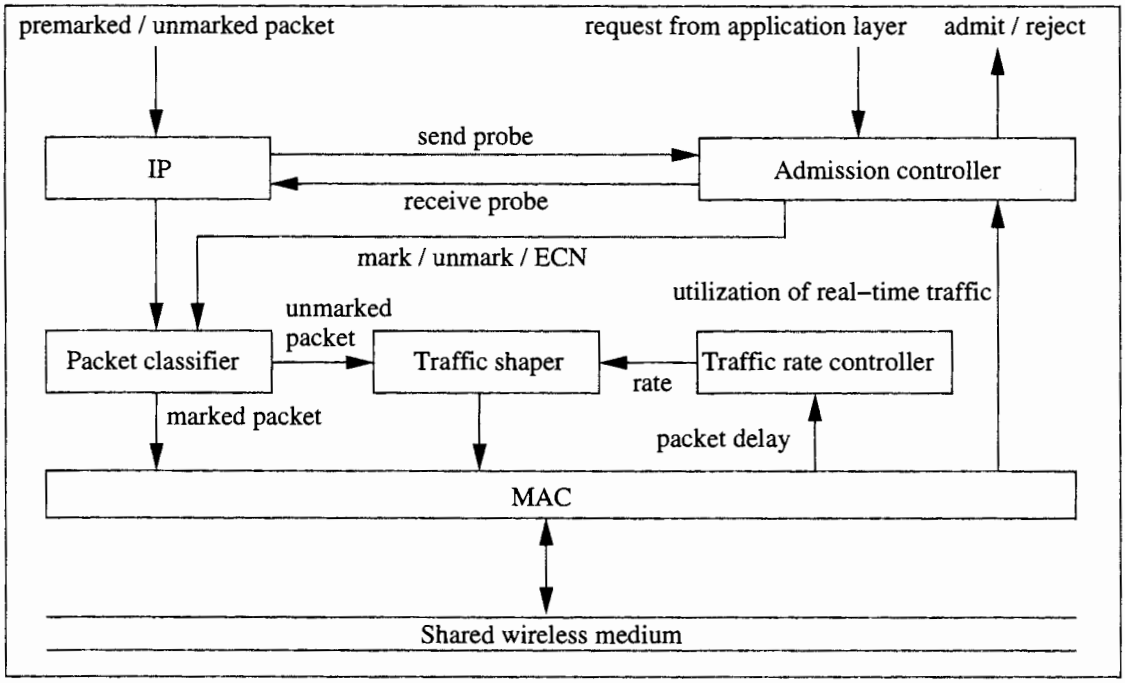


Figure 10.22. The SWAN model.

introduced by mobility (because of rerouting of the already admitted real-time sessions) or traffic variations (*e.g.,* bursty data). The best-effort traffic can be locally and rapidly rate-controlled in an independent manner at each node in order to yield the necessary low delays and stable throughput for real-time traffic. The best-effort traffic utilizes remaining bandwidth (if any) left out by real-time traffic. Hence, this model does not work in scenarios where most of the traffic is real-time in nature.

The *traffic rate controller* determines the departure rate of the traffic shaper using an additive increase multiplicative decrease (AIMD) rate-control algorithm which is based on packet delay feedback from the MAC layer. The SWAN AIMD rate-control algorithm works as follows. Every T seconds, each node increases its transmission rate gradually (additive increase with increment rate of c Kbps). If the packet delays exceed the threshold delay of d seconds, then the node decrements its transmission rate (multiplicative decrease by r percent). The shaping rate is adjusted every T seconds. The traffic rate controller monitors the actual transmission rate. When the difference between the shaping rate and the actual transmission rate is greater than g percent of the actual rate, then the traffic rate controller adjusts the shaping rate to be g percent above the actual rate. This gap allows the best-effort traffic to increase its actual rate gradually. The threshold delay d is based on the delay requirements of real-time applications in the network.

Source-Based Admission Control of Real-Time Traffic

At each node, the admission controller measures the rate of real-time traffic in bps. If the threshold rate that would trigger excessive delays is known, then the bandwidth availability in a shared media channel is simply the difference between the threshold rate and the current rate of transmission of the real-time traffic. But it is difficult to estimate the threshold rate accurately because it may change dynamically depending on traffic patterns. If real-time traffic is admitted up to the threshold rate, then best-effort traffic would be starved of resources. Also, there would be no flexibility to support any increase in the rate of the already admitted real-time sessions, which could occur due to channel dynamics. Hence real-time traffic should be admitted up to an admission control rate which is more conservative than the threshold rate; the best-effort traffic should be allowed to use any remaining bandwidth. The value for the admission control rate can be estimated statistically.

The process of admitting a new real-time session is as follows. The admission controller module at the source node sends a probing request packet toward the destination node to assess the end-to-end bandwidth availability. This is a best-effort control packet that contains a bottleneck bandwidth field. Each intermediate node on the path between the source-destination pair that receives the probing request packet updates the bottleneck bandwidth field in the packet if the bandwidth availability at the node is less than the current value of the field. On receiving the probing request packet, the destination node sends a probing response packet back to the source node with the bottleneck field copied from the received probing request packet. After receiving the response message, the source node admits the

new real-time session only if sufficient end-to-end bandwidth is available. In this model, no bandwidth request is carried in the probing message, no admission control is done at intermediate nodes, and no resource allocation or reservation is done on behalf of the source node during the lifetime of an admitted session.

Impact of Mobility and False Admission

Host mobility and false admission pose a serious threat for fulfilling the service guarantees promised to the flows. Mobility necessitates dynamic rerouting of the admitted real-time flows. Since, due to mobility, nodes may be unaware of flow rerouting, resource conflicts can arise. The newly selected intermediate nodes may not have sufficient resources for supporting previously admitted real-time traffic. Take the case of multiple source nodes initiating admission control at the same instant and sharing common intermediate nodes on their paths to destination nodes. Since intermediate nodes do not maintain state information and since admission control is fully source-based, each source node may receive a response to its probe packet indicating that resources are available, even though the available resources may not be sufficient to satisfy all the requests. The source node, being unaware of this fact, falsely admits a new flow and starts transmitting real-time packets under the assumption that resources are available for meeting the flow's needs. If left unresolved, the rerouting of admitted real-time flows can cause excessive delays in delivery of real-time traffic since the admission control rate is violated by the falsely admitted calls. To resolve this problem, the SWAN AIMD rate control and source-based admission control algorithms were augmented with dynamic regulation of real-time traffic. The algorithms used for this dynamic regulation are described below.

Regulation Algorithms

The ECN-based regulation of real-time sessions operates as follows. Each node continuously estimates the locally available bandwidth. When a node detects congestion/overload conditions, it starts marking the ECN bits in the IP header of the real-time packets. If the destination receives a packet with ECN bits marked, it notifies the source using a regulate message. After receiving a regulate message, the source node initiates reestablishment of its real-time session based on its original bandwidth requirements by sending a probing request packet to the destination. A source node terminates the session if the available end-to-end bandwidth cannot meet its bandwidth requirements. If the node detecting violations marks (*i.e.*, sets) the ECN bits of all packets, then all sessions passing through this node are forced to reestablish their connections at the same instance. Since such an approach is inefficient, the SWAN model considered two approaches in which only a small number of sources are penalized.

Source-Based Regulation

In this scheme, the source node waits for a random amount of time after receiving a regulate message from a congested or overloaded intermediate node on the path to the destination node and then initiates the reestablishment process. This can avoid

flash-crowd conditions. In this scheme, the rate of the real-time traffic will gradually decrease until it reaches below the admission control rate. Then the congested or overloaded nodes will stop marking packets. Even though this scheme is simple and source-based, it has the disadvantage that sources that regulate earlier than other sources are more likely to find the path overbooked and be forced to terminate their sessions.

Network-Based Regulation

Unlike the previous scheme, in this scheme, congested or overbooked nodes randomly select a *congestion set* of real-time sessions and mark only packets associated with that set. A congested node marks the congested set for a time period of T seconds and then calculates a new congested set. Hence, some intelligence is required at the intermediate nodes. As in the previous approach, nodes stop marking packets as *congested* when the measured rate of real-time traffic reaches below the admission control rate.

Advantages and Disadvantages

SWAN gives a framework for supporting real-time applications by assuming a best-effort MAC protocol and not making any resource reservation. It uses feedback-based control mechanisms to regulate real-time traffic at the time of congestion in the network. As best-effort traffic serves as a buffer zone for real-time traffic, this model does not work well in scenarios where most of the traffic is real-time in nature. Even though this model is scalable (because the intermediate nodes do not maintain any per-flow or aggregate state information), it cannot provide hard QoS guarantees due to lack of resource reservation at the intermediate nodes. An admitted real-time flow may encounter periodic violations in its bandwidth requirements. In the worst case, it may have to be dropped or be made to live with downgraded best-effort service. Hence, the local rate control of best-effort traffic mechanism alone may not be sufficient to fully support real-time traffic.

10.6.6 Proactive RTMAC

Proactive RTMAC (PRTMAC) [33] is a cross-layer framework, with an on-demand QoS extension of DSR routing protocol at the network layer and real-time MAC (RTMAC) [13] protocol at the MAC layer. PRTMAC is a tightly coupled solution which requires the bandwidth reservation and bandwidth availability estimation services from the underlying MAC protocol. It is designed to provide enhanced real-time traffic support and service differentiation to highly mobile ad hoc wireless networks such as that formed by military combat vehicles. The performance of real-time calls in ad hoc wireless networks are affected by the mobility of nodes in many different ways. The two major ways in which mobility affects real-time calls are *breakaways* and reservation *clashs*, which will be explained later in this section.

Reservation-based QoS solutions provide end-to-end bandwidth reservation for a real-time connection. This explicit reservation can be affected by the *breakaways* and *clashs*. Proactive RTMAC (PRTMAC) is a solution that operates in both network

and MAC layers and uses an out-of-band signaling channel to gather additional information about the on-going real-time calls, so that proactive measures can be taken to protect these calls. A narrow-band control channel that operates over a transmission range with twice that of the data transmission range, is used as the out-of-band signaling channel.

Figures 10.23 (a) and 10.23 (b) show the *breakaway* problem where a path between node A to node D is established through nodes B and C. The intermediate link B-C can be broken due to the mobility of node C. Now, in a reservation-based real-time support scheme, the intermediate nodes that detect the broken link have to either repair the broken link or inform the sender and receiver about the path break. Since the *breakaways* are very frequent in ad hoc wireless networks, the control overhead generated as part of route reconfiguration can consume a significant amount of bandwidth, in addition to the annoying effect that each path reconfiguration can give to the end users. Figures 10.24 (a) and 10.24 (b) show the reservation *clash* problem in ad hoc wireless networks. Consider the bandwidth reservations done in a given slot (say, slot #1) between nodes A and B and between nodes C and D as illustrated in Figure 10.24 (a). This is a valid reservation because the node pairs are not overlapping in the reserved slot.

Now assume that the node D is mobile and moving toward node B, then at some point of time when the nodes B and D tend to get closer, *i.e.*, within each other's transmission range, the reservation overlapping occurs in slot #1. This is illustrated in Figure 10.24 (b). As a result of this, the packets scheduled to be transmitted by both nodes at this slot can get scrambled and lost. In such a case, traditional path reconfiguration processes identify the broken reservations and reconfigure both of them. This problem is referred to as *clash*.

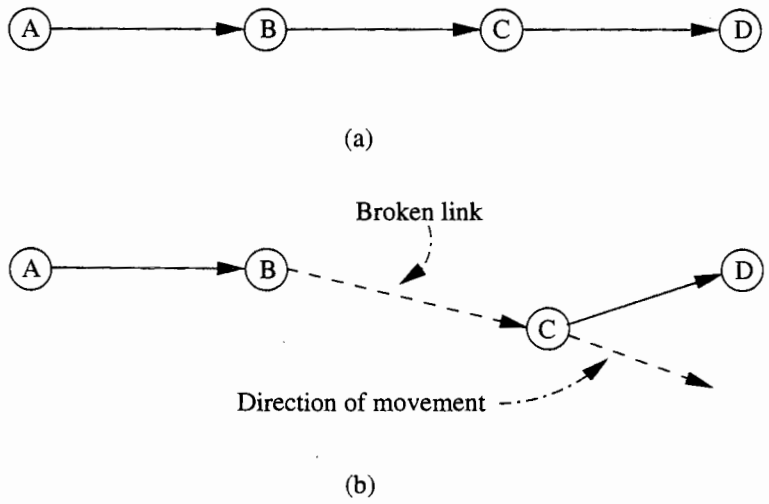
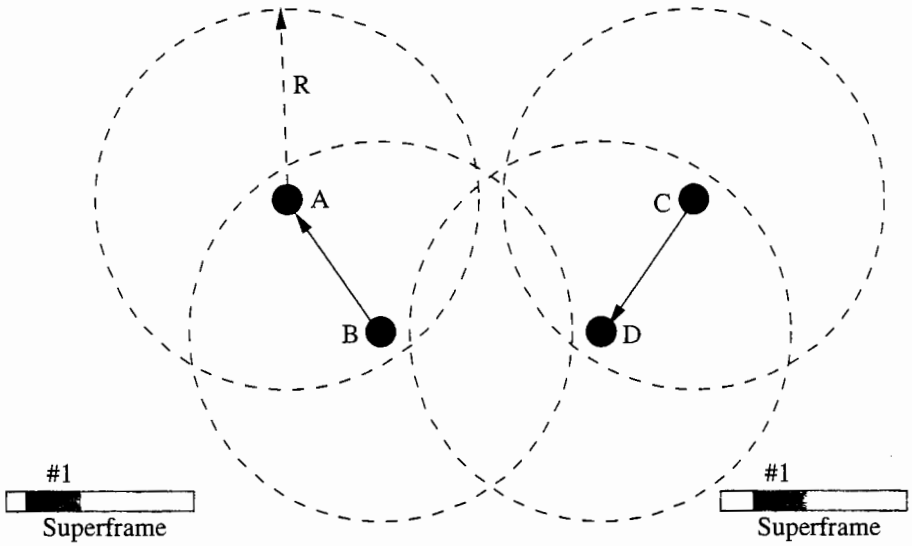
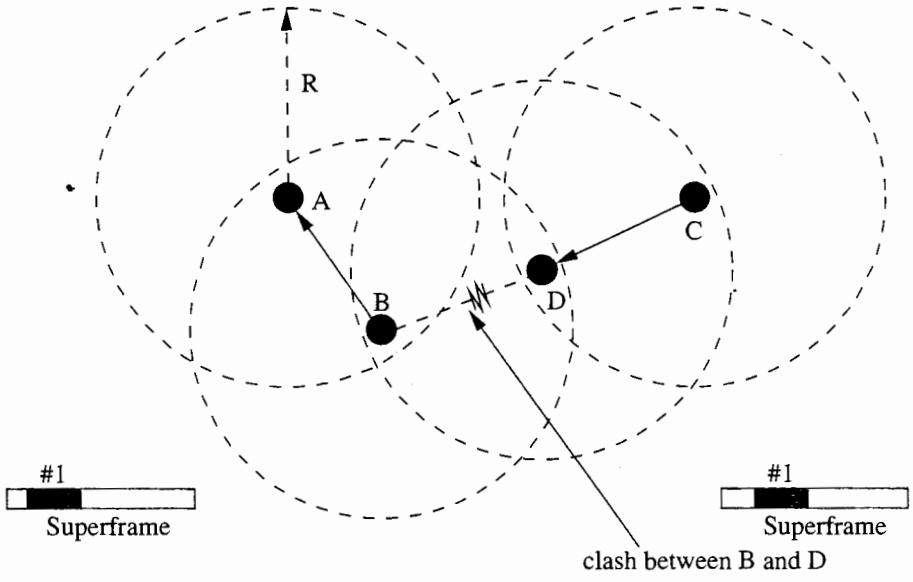


Figure 10.23. Illustration of *breakaway*.



(a) Two overlapping reservations, but out of range: valid.



R: Transmission range

(b) Overlapping reservations within range: clash.

Figure 10.24. Illustration of reservation *clash* due to mobility of nodes.

Operation of PRTMAC

The PRTMAC framework is shown in Figure 10.25. This framework includes an out-of-band signaling module, a proactive call maintenance module, and a routing and call admission control module. The MAC protocol used is RTMAC, which is discussed in detail in Section 6.6.7. The operation of PRTMAC lies in collecting additional information about the real-time calls in the network to counter the effects of *breakaways* and *clashs*. This information is gathered over a narrow-band *control*

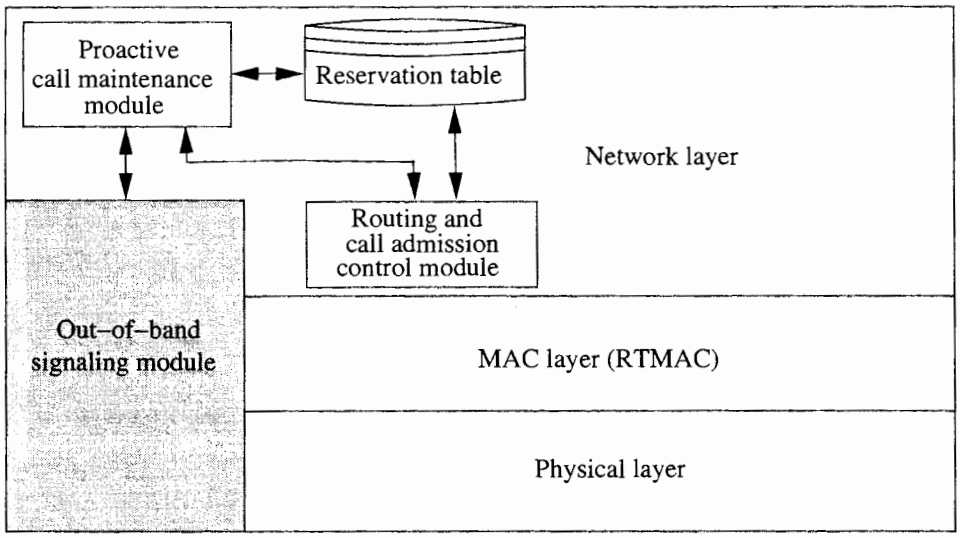


Figure 10.25. Modules in PRTMAC framework.

channel that has a greater transmission range than the data channel. Every node sends out control beacons (short fixed-sized packets) at regular intervals over the control channel. The information carried by the beacons, and the beacon itself, are used by the nodes to gather information about real-time calls. Firstly, the signal strength of the received beacon is used to gain an idea of the relative distance of the node which sent the beacon. Further, the information carried by the beacon is used in predicting *breakaways* and *clashs*. The beacons carry information about each of the calls that the originating node is carrying, and the slots in the superframe that have been reserved for them. Each node originates periodic beacons on the control channel. The beacon has information about all on-going real-time calls at the node. The information includes the start- and end-times of the reservation slot of each call, the sender and the receiver of the call, and the service class (service classes are used to provide differentiated services among the real-time calls existing in the system, for example, the command and control calls in a military communication system may require higher priority than the other calls) to which the call belongs. The range of the control channel must be sufficiently larger than that of the data channel so that all possible events that can cause a call to be interrupted can be discovered well in advance.

Crossover-Time Prediction

Crossover-time is defined as the time at which a node crosses another node's data transmission range r. This event is defined as *crossover*. As apparent from Figures 10.26 (a) and 10.26 (b), there are two different *crossover-times*, namely, *crossover-time-in* and *crossover-time-out*.

The *crossover-time-in* is the expected time at which node B in Figure 10.26 (a) reaches the crossover-point such that a bidirectional link forms between nodes A and B. Figure 10.26 (b) shows the *crossover-time-out*, which occurs at the instant node

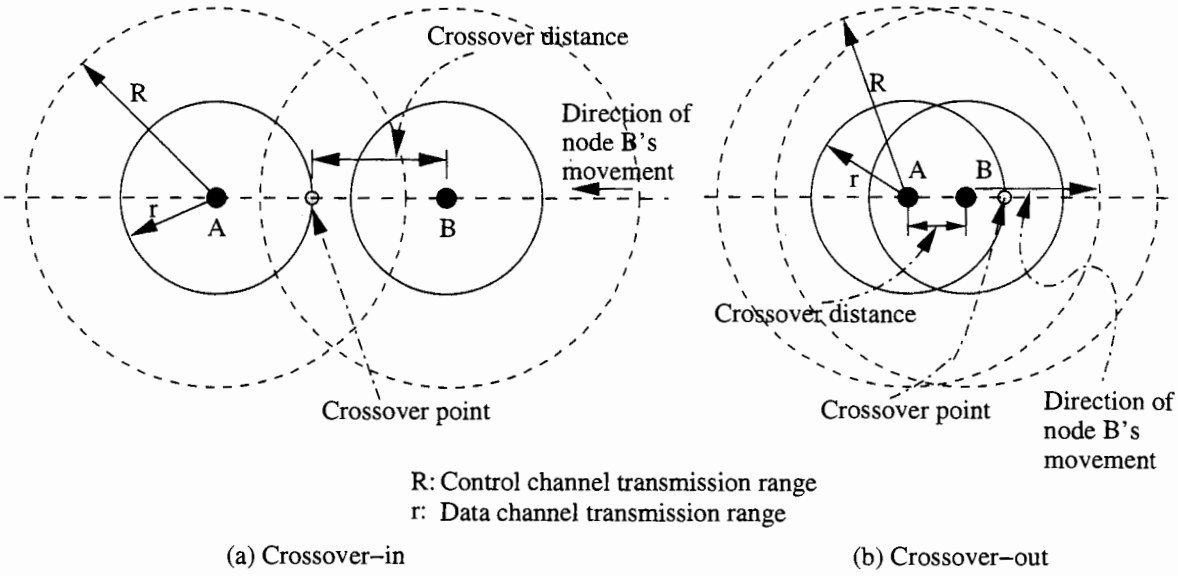


Figure 10.26. Illustration of *crossover-in* and *crossover-out*.

B moves away from node A so that the link between nodes A and B breaks. Each node (say, node A), upon reception of every new beacon from another node (say, node B), predicts the *crossover-time* based on the signal strength history obtained from past beacons, that is, if node B is inside the range of the data channel of node A, node A predicts the *crossover-time-out*, and if node B is outside the range of the data channel of node A, node A predicts the *crossover-time-in*.

The prediction of the *crossover-time-out* of node B with respect to node A is performed by keeping track of the signal strengths of the beacons previously sent by node B to node A (see Figure 10.27). A node stores a fixed number of $< time, signalstrength >$ tuples of the beacons received from any other node. Using this, it generates a polynomial on the variation of signal strength with time. The roots of the polynomial refer to the time at which the signal strength can cross a receiving threshold. When node A predicts that node B is going to cross the data channel range within the next beacon interval, it takes proactive actions described in the next section. If node B is already within the data channel range of node A, then the prediction will be for a *crossover-out* event, and all calls between nodes A and B will be interrupted. If node B is outside the range of node A, then it is a *crossover-in* event, and any packets belonging to existing real-time calls at node A and node B will collide if their reservation times overlap. Note that if the predicted time of entry is beyond the next beacon interval, no action needs to be taken as of now, since the event would be predicted again on receipt of the next beacon. When a node discovers that the *crossover* of another node is imminent, it takes proactive action to safeguard its privileged calls. If it is a *crossover-out*, it checks to see if it has any on-going traffic with the other node. If so, it has to resolve a case of *breakaway*. On the other hand, if it is a case of *crossover-in*, it examines the traffic

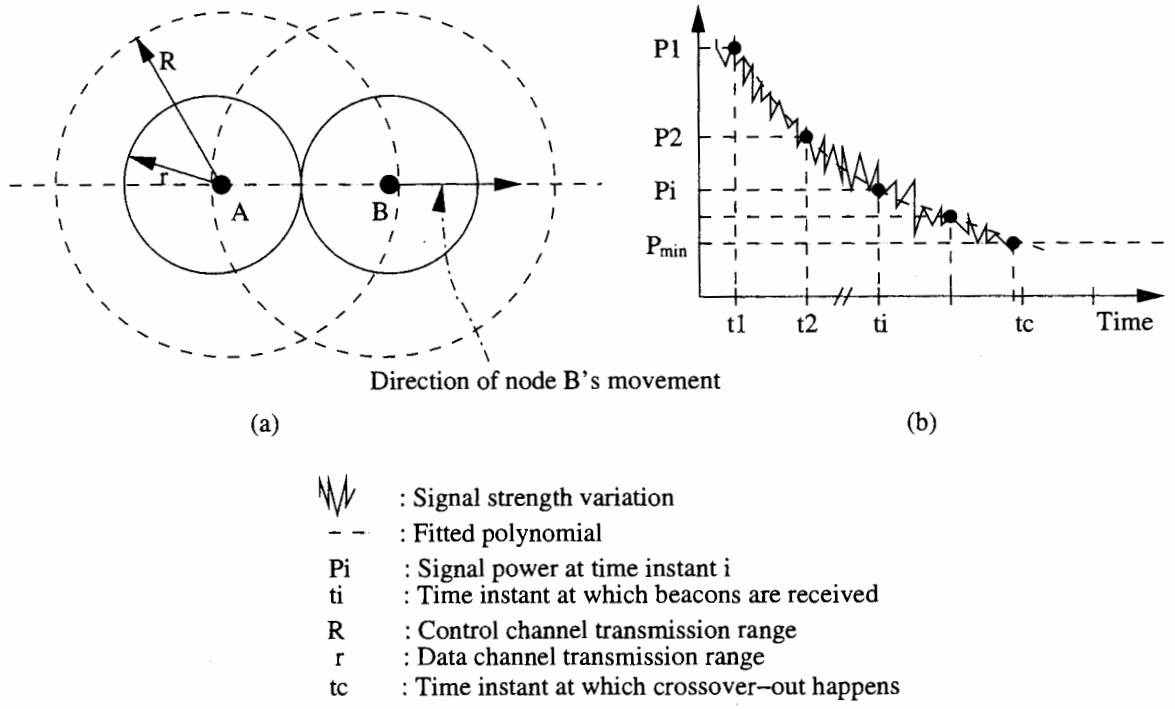


Figure 10.27. Illustration of prediction *crossover-time-out*: (a) Node B moves away from node A. (b) Signal strength variation of received beacons with time.

information of the other node and its own reservation tables to check for possible overlaps. If there is any such overlap, it has to resolve a case of *clash*.

Handling Breakaways

Figure 10.28 illustrates the handling of a broken path due to *breakaway*. The event of *breakaways* can be handled in two different ways: First is the local reconfiguration and second is the end-to-end reconfiguration.

The path reconfiguration is performed when a link breaks as in Figure 10.28 (a), where the path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ between nodes A and E is illustrated and the link formed between nodes C and D is broken. The local reconfiguration is also illustrated in Figure 10.28 (a), where the intermediate node, the link with whose downstream node is broken, holds the responsibility of finding a path to the destination node E. The intermediate node (node C) originates fresh route probe packets to obtain a path with reservation to the destination node. The end-to-end reconfiguration method is depicted in Figure 10.28 (b), where node C originates *RouteError* to the sender node in order to obtain a new path to the destination node. In this case, the reservation done in the old path may entirely be released. In PRTMAC a combination of the above two types is attempted, which is described as follows: Node C checks to see if its routing tables have another path toward the destination node (say, node F). If there exists such a node, then node C makes reservations on the link C-F for the on-going call. When the call is interrupted and

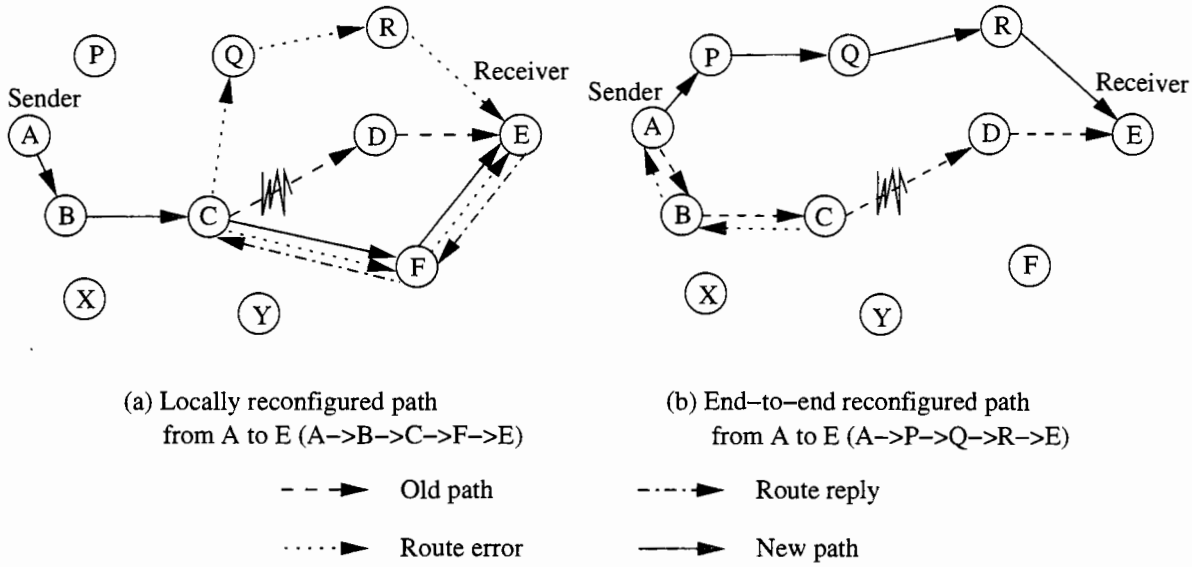


Figure 10.28. Illustration of route reconfiguration schemes for a path affected by breakaway.

reconfigured locally a number of times, as expected in an ad hoc wireless network, the end-to-end reconfiguration is attempted.

Handling Clashs

Figure 10.29 (a) illustrates how two nodes can reside safely within range of each other if the reserved slots do not overlap with each other. If the reservation slots *clash* for the two nodes, as indicated in Figure 10.29 (b), then PRTMAC handles it in such way that the flow between, say, node N and node C is assigned to a new slot (#5), as shown in Figure 10.30.

In the absence of any measures taken to resolve a *clash*, both the calls that experience a *clash* will be reconfigured from the source to the destination, resulting in degradation of performance. PRTMAC prevents such an occurrence to the extent possible by proactively shifting one of the calls to a new slot, so that the two calls do not *clash*. This benefit of *clash* resolution is more important when a higher priority call *clash*s with a lower priority call.

For the proactive slot shifting of one of the calls to happen, there is a need to decide unambiguously the node that will perform this reconfiguration. Since PRTMAC wants a high-priority call to continue undisturbed as long as possible, in case of a *clash* between a high-priority call and a low-priority call, it is the responsibility of the node having the low-priority call to reconfigure it to a new slot. If both the calls that *clash* are of high-priority, the one with the lower FlowID has to be reconfigured, and if both such calls have the same FlowID, the node with the lower node address (expected to be unique throughout the network) will have to reconfigure its call. The FlowID is a unique number associated with a call at a given node.

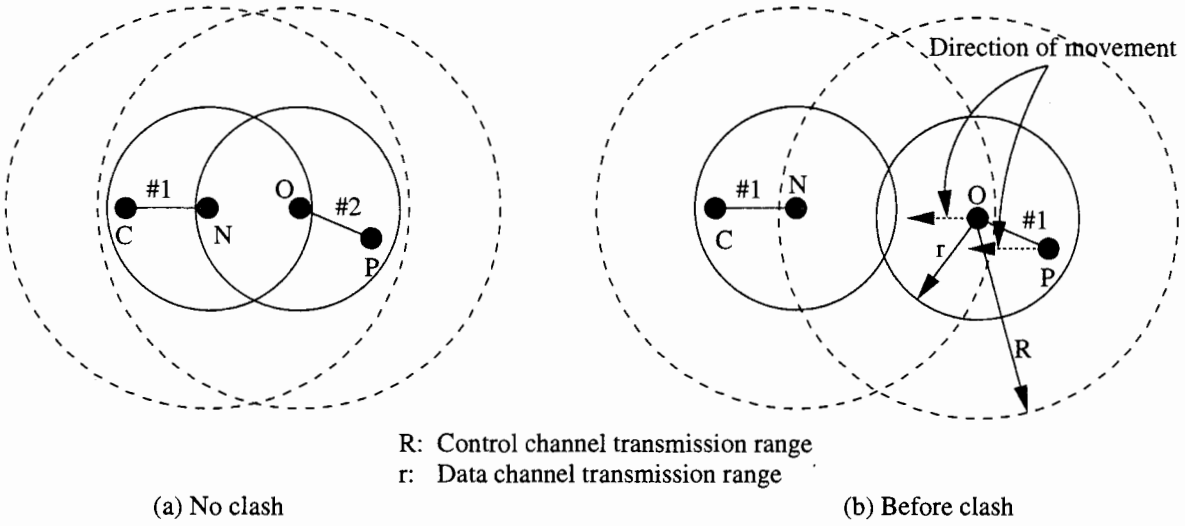


Figure 10.29. (a) Nodes N and O are within range, but reservation is safe. (b) Nodes N and O are out of range but are going to *clash* as nodes P and O are moving toward them.

As illustrated in Figure 10.30, the node whose responsibility it is to reconfigure the call is denoted by node N, the other node, whose call *clash*s with node N's call, is denoted by node O, and the counterpart of node N in its call by node C. Node N goes through its reservation tables and its neighbor reservation table corresponding to node C and tries to come up with a free reservation slot in both nodes N and C large enough to accommodate the call to be shifted. If it succeeds in finding such a free slot, the existing reservations for the call must be dropped and new reservations must be made for the call in the free slot. This is achieved when the originator of the call frees the earlier reservation and issues a request for the reservation of the slots belonging to the free slot.

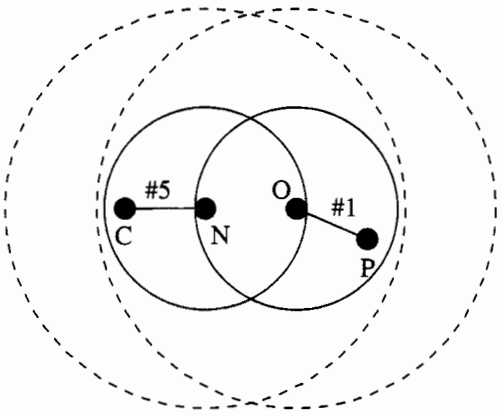


Figure 10.30. *Clash* handling reassigns the flow through the link N-C to a new slot.

If both the calls that *clash* have high priority and node N cannot come up with a slot free enough to accommodate the call, it informs node O about its failure in shifting the call. Now node O executes the above process with its counterpart and tries to shift the call. If one of the calls that *clash* is a high-priority call and the other a low-priority one, and the node that has a low-priority call (here it is node N) is unable to find a new slot to shift the call, the low-priority call undergoes end-to-end reconfiguration. This is to ensure that the low-priority call would not hinder the high-priority calls.

Differentiated Services Provisioning in PRTMAC

PRTMAC also provides class-based service differentiation among the priority classes. It supports the following three classes:

- *Class 1* is the class with the highest privilege. Calls belonging to this class are high-priority real-time calls. PRTMAC attempts to ensure that a Class 1 call is not interrupted as far as possible. If a situation occurs in which a Class 1 call can be sustained only at the expense of calls belonging to the other classes, PRTMAC utilizes the possibility and preempts the low-privilege call to preserve the Class 1 call.

- *Class 2* is the next privileged class. Calls belonging to this class are real-time calls with bandwidth reservation. A Class 2 call is sustained provided that there are no events in the network (such as *clash*s or *breakaway*s) that could cause the call to be terminated. Class 2 calls may be preempted in order to service Class 1 calls. A Class 2 call has an end-to-end bandwidth reservation at the time of accepting the call. If such a reservation is not possible, then the call is not accepted.

- The *Best-effort* class is the least privileged class. There are no guarantees made regarding any parameters corresponding to best-effort traffic.

The following are three major instances where priority is given to a Class 1 call over a Class 2 call:

- Preempting Class 2 calls for Class 1 calls while a new call is admitted.

- Handling of *clash*s and *breakaway*s for Class 1 calls.

- Prioritizing path-reconfiguration attempts for Class 1 calls.

During a new call admission, PRTMAC tries to admit a Class 1 call, even if it requires preempting an existing Class 2 call. In addition to privileged admission, upon detection of an imminent *clash* of reservation, PRTMAC provides higher priority to the Class 1 calls in order to protect them from being affected by the *clash*.

In addition to the above, during the reconfiguration process, the number of attempts made for reconfiguring a broken Class 1 call is higher than that of Class 2 calls. Among the equal priority calls, the differentiation is provided based on the

node addresses and FlowIDs. Hence, by deliberately providing node addresses to the designated persons or nodes based on their rank or position in military establishments, and by choosing the FlowIDs for the calls originated by them, PRTMAC can provide very high chances of supporting guaranteed services. For example, the leader of the military group can be given the highest node address, and if the system is so designed that the FlowIDs for the calls he generates are higher than those for the rest of the nodes, then none of the other calls can terminate his calls if he decides to originate a Class 1 call.

Advantages and Disadvantages

PRTMAC is appropriate in providing better real-time traffic support and service differentiation in high mobility ad hoc wireless networks such as military networks formed by high-speed combat vehicles, fleets of ships, and fleets of aircrafts where the power resource is not a major concern. In ad hoc wireless networks, formed by low-power and resource-constrained handheld devices, having another channel may not be an economically viable solution.

10.7 SUMMARY

In this chapter, several solutions proposed in the literature for providing QoS support for applications in ad hoc wireless networks have been described. First, the issues and challenges in providing QoS in ad hoc wireless networks were discussed. Then the classifications of the existing QoS approaches under several criteria such as interaction between routing protocol and resource reservation signaling, interaction between network and MAC layer, and information update mechanism were discussed. The data link layer solutions such as cluster TDMA, IEEE 802.11e, and DBASE and the network layer solutions such as ticket-based probing, predictive location-based QoS routing, trigger-based QoS routing, QoS enabled AODV, bandwidth routing, on-demand routing, asynchronous QoS routing, and multipath QoS routing were described. Finally, QoS frameworks for ad hoc wireless networks such as INSIGNIA, INORA, SWAN, and PRTMAC were described.

10.8 PROBLEMS

1. What are the limitations of the IEEE 802.11 MAC protocol that prevent it from supporting QoS traffic?

2. Express various inter-frame spaces (IFSs) of the IEEE 802.11e MAC protocol in terms of $SIFS$ and *slottime*.

3. Compare and contrast the hybrid coordinator (HC) of the IEEE 802.11e MAC protocol with the point coordinator (PC) of the IEEE 802.11 MAC protocol.

4. What are the advantages of having transmission opportunities (TXOPs) in the IEEE 802.11e MAC protocol?

5. Compare and contrast the IEEE 802.11e MAC protocol with the DBASE protocol.

6. Discuss how a source node determines how many number of tickets (green and yellow tickets) are to be issued for a session in delay-constrained TBP protocol.

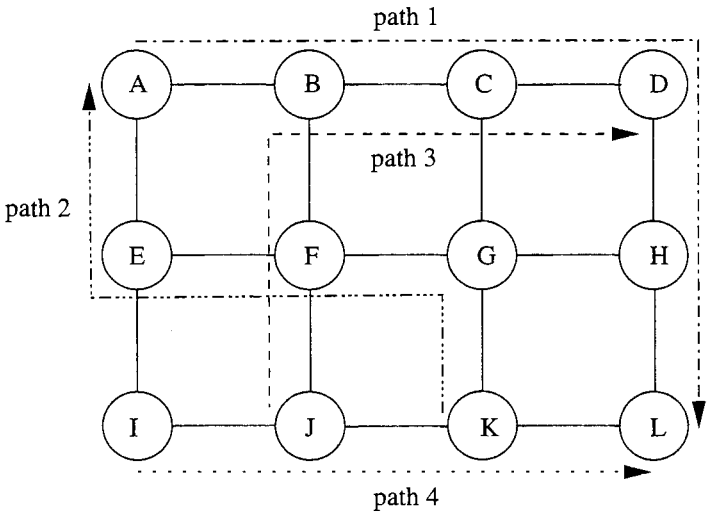
7. Discuss how a node estimates its expected location and under what circumstances the node generates a *Type2* update message in PLBQR protocol.

8. Consider the network topology shown in Figure 10.31 (a). Assume that free slots available at various nodes are as given in Figure 10.31 (b).

   (a) Using the hop-by-hop path bandwidth calculation algorithm proposed in the BR protocol, calculate the end-to-end path bandwidth for the paths given below.

      i. PATH1: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow H \rightarrow L$
      ii. PATH2: $K \rightarrow G \rightarrow F \rightarrow E \rightarrow A$
      iii. PATH3: $J \rightarrow F \rightarrow B \rightarrow C \rightarrow D$
      iv. PATH4: $I \rightarrow J \rightarrow K \rightarrow L$

   (b) Further assume that four call setup packets are generated in the order given below. After admitting or rejecting a particular call, the next call is issued. Discuss which of these calls are admitted and rejected.



(a)

node ID	free slots
A	{1,2,4,6,7,8}
B	{1,2,5,6,7}
C	{2,4,6,7,8}
D	{1,4,5,6,7,8}
E	{1,3,5,6,7}
F	{1,2,4,6,7,8}
G	{1,3,5,6,7}
H	{2,3,5,6,8}
I	{1,2,4,6,8}
J	{1,5,6,7,8}
K	{2,3,5,6,8}
L	{1,3,4,5,8}

(b)

Figure 10.31. Network topology and slot table for Problem 8.

i. Node A issues a call setup packet having a 2-slots bandwidth requirement. The call setup packet traverses along the PATH1.

ii. Node K issues a call setup packet having a 1-slot bandwidth requirement. The call setup packet traverses along the PATH2.

iii. Node J issues a call setup packet having a 2-slots bandwidth requirement. The call setup packet traverses along the PATH3.

iv. Node I issues a call setup packet having a 2-slots bandwidth requirement. The call setup packet traverses along the PATH4.

9. Consider the network topology shown in Figure 10.32. Construct T and T_{LCF} for path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$. What is the maximum end-to-end bandwidth for that path?

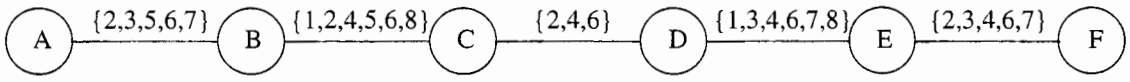


Figure 10.32. Network topology for Problem 9.

10. What are the pros and cons of using the hop-by-hop path bandwidth calculation algorithm proposed in the BR and OQR protocols over the approach used during the unipath discovery operation of the OLMQR protocol for the end-to-end path bandwidth calculation?

11. Write an algorithm for SWAN AIMD rate control mechanism.

12. Compare the admission control mechanisms of INSIGNIA and SWAN frameworks.

13. Assume that a session S requires X units of bandwidth and an intermediate node I has Y ($Y < X$) units of bandwidth available at that node. Discuss how node I responds after receiving a *RouteRequest* packet for the session S in INSIGNIA and INORA frameworks.

14. Assume that a QoS session is being initiated by the source node A to the destination node I with six units of bandwidth requirement. Let the DAG created by TORA protocol for this QoS session be as shown in Figure 10.33. In this figure, the label on each link specifies the available bandwidth on that link. Discuss how the QoS session is admitted (or rejected) in case of coarse feedback mechanism and class-based fine feedback mechanism for the following two cases:

(a) $BW_{EH} = BW_{FI} = 3$ and $BW_{HI} = 10$.

(b) $BW_{FI} = 2$, $BW_{EH} = 3$, and $BW_{HI} = 4$.

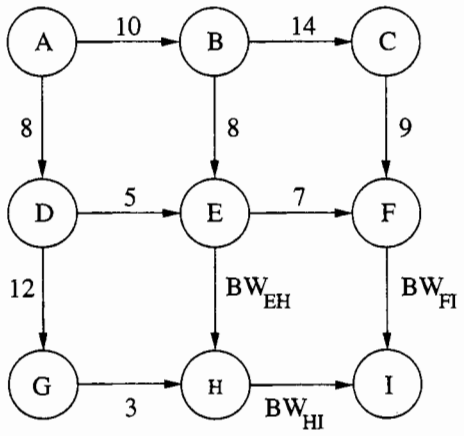


Figure 10.33. Topology for Problem 14.

15. In a military-vehicular ad hoc wireless network using PRTMAC, formed by 500 nodes distributed uniformly in a battlefield area of 1,000 m $\times$ 1,000 m, calculate the number of nodes contending for the data channel and for the control channel. The transmission range of the data channel is 250 m.

16. In Problem 15, find the probability that a beacon gets collided, when the beacons are generated periodically with a period of $P_b = 10$ seconds. Assume the beacon length to be equal to 1 ms.

17. In a PRTMAC system, assume that the beacon transmissions are carried out in the control channel with a period of 5 seconds, with an accurate prediction mechanism. Calculate the probability that an impending collision from another node traveling with a constant velocity of 20 m/s goes undetected. Assume that the transmission range of the data channel is 240 m and the probability of a beacon getting collided is 0.2.

BIBLIOGRAPHY

[1] P. Karn, "MACA: A New Channel Access Method for Packet Radio," *Proceedings of ARRL/CRRL Amateur Radio* 9^{th} *Computer Networking Conference 1990*, pp. 134-140, September 1990.

[2] IEEE Standards Board, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *The Institute of Electrical and Electronics Engineers, Inc.*, 1997.

[3] M. Gerla and J. T. C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *ACM/Baltzer Wireless Networks Journal*, vol. 1, no. 3, pp. 255-265, October 1995.

[4] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," *Proceedings of the European Wireless 2002*, vol. 1, pp. 32-39, February 2002.

[5] M. Veeraraghavan, N. Cocker, and T. Moors, "Support of Voice Services in IEEE 802.11 Wireless LANs," *Proceedings of IEEE INFOCOM 2001*, vol. 1, pp. 488-497, April 2001.

[6] M. A. Visser and M. E. Zarki, "Voice and Data Transmission Over an 802.11 Wireless Network," *Proceedings of IEEE PIMRC 1995*, vol. 2, pp. 648-652, September 1995.

[7] IEEE 802.11 TGe, "EDCF Proposed Draft Text," *TR-01/131r1*, March 2001.

[8] IEEE 802.11 TGe, "Hybrid Coordination Function (HCF)–Proposed Updates to Normative Text of D0.1," *TR-01/110r1*, March 2001.

[9] IEEE 802.11 TGe, "HCF Ad Hoc Group Recommendation–Normative Text to EDCF Access Category," *TR-02/241r0*, March 2001.

[10] IEEE 802.11 TGe, "Proposed Normative Text for AIFS–Revisited," *TR-01/270r0*, February 2003.

[11] S. Sheu and T. Sheu, "DBASE: A Distributed Bandwidth Allocation/Sharing/Extension Protocol for Multimedia over IEEE 802.11 Ad Hoc Wireless LAN," *Proceedings of IEEE INFOCOM 2001*, vol. 3, pp. 1558-1567, April 2001.

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

[13] 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.

[14] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1488-1504, August 1999.

[15] S. H. Shah and K. Nahrstedt, "Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks," *Proceedings of IEEE ICC 2002*, vol. 2, pp. 1022-1027, May 2002.

[16] S. De, S. K. Das, H. Wu, and C. Qiao, "Trigger-Based Distributed QoS Routing in Mobile Ad Hoc Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 22-35, July 2002.

[17] C. E. Perkins, E. M. Royer, and S. R. Das, "Quality of Service for Ad Hoc On-Demand Distance Vector Routing," *IETF Internet Draft, draft-ietf- manet-aodvqos-00.txt*, July 2000.

[18] C. R. Lin and J. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1426-1438, August 1999.

[19] C. E. 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.

[20] C. R. Lin, "On-Demand QoS Routing in Multi-Hop Mobile Networks," *Proceedings of IEEE INFOCOM 2001*, vol. 3, pp. 1735-1744, April 2001.

[21] Y. Chen, Y. Tseng, J. Sheu, and P. Kuo, "On-Demand, Link-State, Multipath QoS Routing in a Wireless Mobile Ad Hoc Network," *Proceedings of European Wireless 2002*, pp. 135-141, February 2002.

[22] V. Vidhyashankar, B. S. Manoj, and C. Siva Ram Murthy, "Slot Allocation Schemes for Delay-Sensitive Traffic Support in Asynchronous Wireless Mesh Networks," *Proceedings of HiPC 2003*, LNCS 2913, pp. 333-342, December 2003.

[23] B. S. Manoj, V. Vidhyashankar, and C. Siva Ram Murthy, "Slot Allocation Strategies for Delay-Sensitive Traffic Support in Asynchronous Ad Hoc Wireless Networks," to appear in *Journal of Wireless Communications and Mobile Computing*, 2004.

[24] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454-1466, August 1999.

[25] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," *IETF RFC1633*, June 1994.

[26] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *IETF RFC2475*, December 1998.

[27] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reSerVation Protocol (RSVP) – Version 1 Functional Specification," *IETF RFC 2205*, September 1997

[28] H. Xiao, K. G. Seah, A. Lo, and K. C. Chua, "A Flexible Quality of Service Model for Mobile Ad Hoc Networks," *Proceedings of IEEE Vehicular Technology Conference*, vol. 1, pp. 445-449, May 2000.

[29] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts," *ACM/Baltzer Wireless Networks Journal*, vol. 7, no. 1, pp. 5-19, January 2001.

[30] S. B. Lee, A. Gahng-Seop, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 374-406, April 2000.

[31] D. Dharmaraju, A. R. Chowdhury, P. Hovareshti, and J. S. Baras, "INORA–A Unified Signalling and Routing Mechanism for QoS Support in Mobile Ad Hoc Networks," *Proceedings of ICPPW 2002*, pp. 86-93, August 2002.

[32] H. Ahn, A. T. Campbell, A. Veres, and L. Sun, "Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, pp. 192-207, September 2002.

[33] T. Sandeep, V. Vivek, B. S. Manoj, and C. Siva Ram Murthy, "PRTMAC: An Enhanced Real-Time Support Mechanism for Tactical Ad Hoc Wireless Networks," *Technical Report*, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, June 2001. (A shorter version of this report has been accepted for presentation at *IEEE RTAS 2004*.)

[34] T. Bheemarjuna Reddy, I. Karthigeyan, B. S. Manoj, and C. Siva Ram Murthy, "Quality of Service Provisioning 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, July 2003.