MOBILE COMPUTING-(13A05801)

LECTURE NOTES

UNIT-I

WIRELESS LANS AND PANS

1.1 INTRODUCTION

The field of computer networks has grown significantly in the last three decades. An interesting usage of computer networks is in offices and educational institutions, where tens (sometimes hundreds) of personal computers (PCs) are interconnected, to share resources (*e.g.*, printers) and exchange information, using a high-bandwidth communication medium (such as the Ethernet). These privately-owned networks are known as local area networks (LANs) which come under the category of small-scale networks(networks within a single building or campus with a size of a few kilometres). To do away with the wiring associated with the interconnection of PCs in LANs, researchers have explored the possible usage of radio waves and infrared light for interconnection. This has resulted in the emergence of wireless LANs (WLANs), where wireless transmission is used at the physical layer of the network. Wireless personal area networks (WPANs) are the next step down from WLANs, covering smaller areas with low power transmission, for networking of portable and mobile computing devices such as PCs, personal digital assistants (PDAs), which are essentially very small computers designed to consume as little power as possible so as to increase the lifetime of their batteries, cell phones, printers, speakers, microphones, and other consumer electronics.

1.2 FUNDAMENTALS OF WLANS

The terms "node," "station," and "terminal" are used interchangeably. While both portable terminals and mobile terminals can move from one place to another, portable terminals are accessed only when they are stationary. Mobile terminals (MTs), on the other hand, are more powerful, and can be accessed when they are in motion. WLANs aim to support truly mobile work stations.

1.2.1 Technical Issues The differences between wireless and wired networks, the use of WLANs, and the design goals for WLANs.

Differences Between Wireless and Wired Transmission

• **Address is not equivalent to physical location:** In a wireless network, address refers to a particular station and this station need not be stationary. Therefore, address may not always refer to a particular geographical location.

• **Dynamic topology and restricted connectivity:** The mobile nodes may often go out of reach of each other. This means that network connectivity is partial at times.

• **Medium boundaries are not well-defined:** The exact reach of wireless signals cannot be determined accurately. It depends on various factors such as signal strength and noise levels. This means that the precise boundaries of the medium cannot be determined easily.

• **Error-prone medium:** Transmissions by a node in the wireless channel are affected by simultaneous transmissions by neighboring nodes that are located within the direct transmission range of the transmitting node. This means that the error rates are significantly higher in the wireless medium. We need to build a reliable network on top of an inherently unreliable channel. This is realized in practice by having reliable protocols at the MAC layer, which hide the unreliability that is present in the physical layer.

Uses of WLANs

Wireless computer networks are capable of offering versatile functionalities. WLANs are very flexible and can be configured in a variety of topologies based on the application. Some possible uses of WLANs are mentioned below.

• Users would be able to surf the Internet, check e-mail, and receive Instant Messages on the move.

• In areas affected by earthquakes or other such disasters, no suitable infrastructure may be available on the site. WLANs are handy in such locations to set up networks on the fly.

• There are many historic buildings where there has been a need to set up computer networks. In such places, wiring may not be permitted or the building design may not be conducive to efficient wiring. WLANs are very good solutions in such places.

Design Goals

The following are some of the goals which have to be achieved while designing WLANs:

• **Operational simplicity:** Design of wireless LANs must incorporate features to enable a mobile user to quickly set up and access network services in a simple and efficient manner.

• **Power-efficient operation:** The power-constrained nature of mobile computing devices such as laptops and PDAs necessitates the important requirement of WLANs operating with minimal power consumption. Therefore, the design of WLAN must incorporate power-saving features and use appropriate technologies and protocols to achieve this.

• **License-free operation:** One of the major factors that affects the cost of wireless access is the license fee for the spectrum in which a particular wireless access technology operates. Low cost of access is an important aspect for popularizing a WLAN technology. Hence the design of WLAN should consider the parts of the frequency spectrum (*e.g.,* ISM band) for its operation which do not require an explicit licensing.

• **Tolerance to interference:** The proliferation of different wireless networking technologies both for civilian and military applications and the use of the microwave frequency spectrum for non-communication purposes(*e.g.*, microwave ovens) have led to a significant increase in the interference level across the radio spectrum. The WLAN design should account for this and take appropriate measures by way of selecting technologies and protocols to operate in the presence of interference.

• **Global usability:** The design of the WLAN, the choice of technology, and the selection of the operating frequency spectrum should take into account the prevailing spectrum restrictions in countries across the world. This ensures the acceptability of the technology across the world.

• **Security:** The inherent broadcast nature of wireless medium adds to the requirement of security features to be included in the design of WLAN technology.

• **Safety requirements:** The design of WLAN technology should follow the safety requirements that can be classified into the following: (i) interference to medical and other instrumentation devices and (ii) increased power level of transmitters that can lead to health hazards. A well-designed WLAN should follow the power emission restrictions that are applicable in the given frequency spectrum.

• **Quality of service requirements:** Quality of service (QoS) refers to the provisioning of designated levels of performance for multimedia traffic. The design of WLAN should take into consideration the possibility of supporting a wide variety of traffic, including multimedia traffic.

• **Compatibility with other technologies and applications:** The interoperability among the different LANs (wired or wireless) is important for efficient communication between hosts operating with different LAN technologies. In addition to this, interoperability with existing WAN protocols such as TCP/IP of the Internet is essential to provide a seamless communication across the WANs.

1.2.2 Network Architecture This section lists the types of WLANs, the components of a typical WLAN, and the services offered by a WLAN.

Infrastructure Based Versus Ad Hoc LANs

WLANs can be broadly classified into two types, infrastructure networks and adhoc LANs, based on the underlying architecture. Infrastructure networks contain special nodes called *access points* (APs), which are connected via existing networks. APs are special in the sense that they can interact with wireless nodes as well as with the existing wired network. The other wireless nodes, also known as mobile stations, communicate via APs. The APs also act as bridges with other networks. Ad hoc LANs do not need any fixed infrastructure. These networks can be setup on the fly at any place. Nodes communicate directly with each other or forward messages through other nodes that are directly accessible.

Components in a Typical IEEE 802.11 Network

IEEE 802.11 is the most popular WLAN standard that defines the specification for the physical and MAC layers. The success of this standard can be understood from the fact that the revenue from the products based on this standard touched $730 million in the second quarter of the year 2003. The basic components in a typical IEEE 802.11 WLAN are listed. The set of stations that can remain in contact (*i.e*., are associated) with a given AP is called a basic service set (BSS). The coverage area of an AP within which member stations (STAs or MTs) may remain in communication is called the basic service area (BSA). The stations that are a part of a BSS need to be located within the BSA of the corresponding AP. A BSS is the basic building block of the network. BSSs are connected by means of a distribution system(DS) to form an extended network.DS refers to an existing network infrastructure. The implementation of the DS is not specified by the IEEE 802.11 standard. The services of the DS, however, are specified rigidly. This gives a lot of flexibility in the design of the DS. The APs are connected by means of the DS. Portals are logical points through which non-IEEE 802.11 packets (wired LAN packets) enter the system. They are necessary for integrating wireless networks with the existing wired

networks. Just as an AP interacts with the DS as well as the wireless nodes, the portal interacts with the wired network as well as with the DS. The BSSs, DS, and the portals together with the stations they connect constitute the extended service set (ESS). An ad hoc LAN has only one BSS. Therefore, ad hoc LANs are also known as independent basic service sets (IBSSs). It may be noted that the ESS and IBSS appear identical to the logical link control (LLC).

Figure 1.1 gives a schematic picture of what a typical ESS looks like

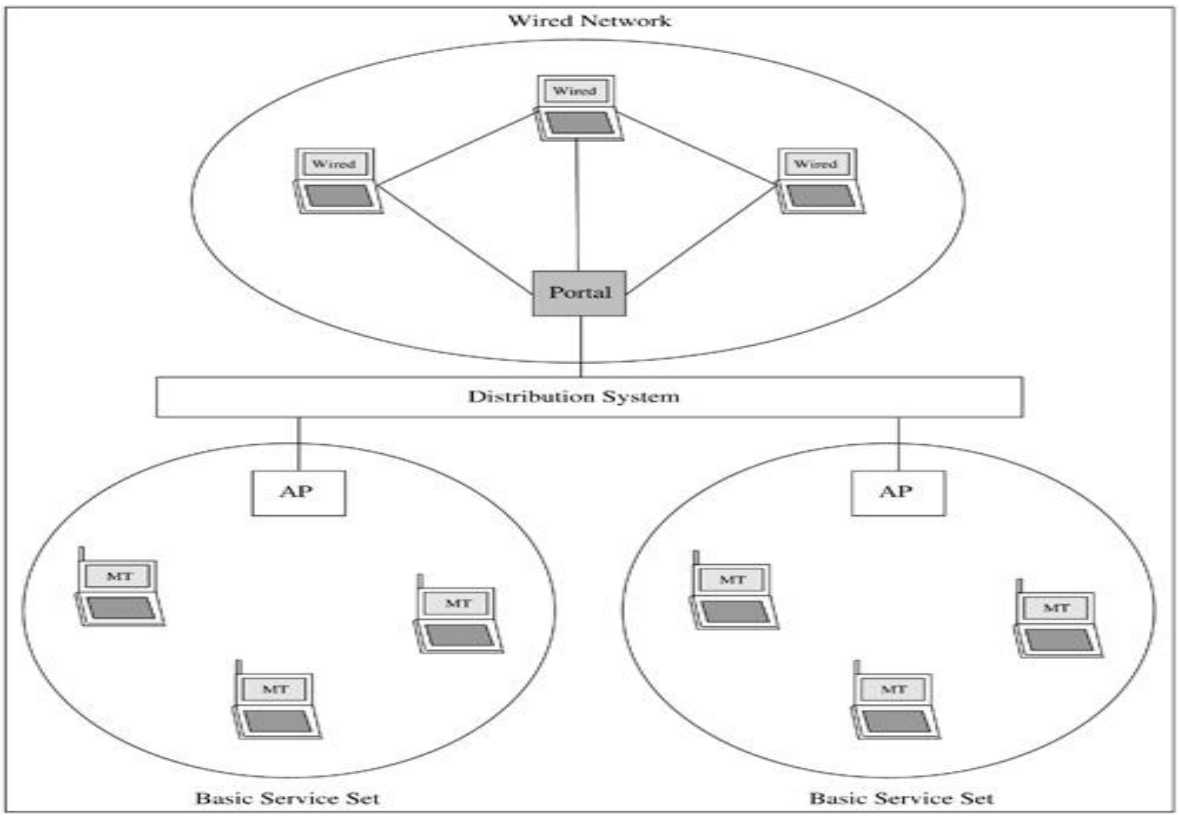


Figure 1.1. Extended Service Set.

Services Offered by a Typical IEEE 802.11 Network

The services offered by a typical IEEE 802.11 network can be broadly divided into two categories: AP services and STA services. The following are the AP services, which are provided by the DS:

• **Association:** The identity of an STA and its address should be known to the AP before the STA can transmit or receive frames on the WLAN. This is done

during association, and the information is used by the AP to facilitate routing of frames.

• **Reassociation:** The established association is transferred from one AP to another using reassociation. This allows STAs to move from one BSS to another.

• **Disassociation:** When an existing association is terminated, a notification is issued by the STA or the AP. This is called disassociation, and is done when nodes leave the BSS or when nodes shut down.

• **Distribution:** Distribution takes care of routing frames. If the destination is in the same BSS, the frame is transmitted directly to the destination, otherwise the frame is sent via the DS.

• **Integration:** To send frames through non-IEEE 802.11 networks, which may have different addressing schemes or frame formats, the integration service is invoked.

The following are the STA services, which are provided by every station, including APs:

• **Authentication:** Authentication is done in order to establish the identity of stations to each other. The authentication schemes range from relatively insecure handshaking to public-key encryption schemes.

• D**eauthentication:** Deauthentication is invoked to terminate existing authentication.

• **Privacy:** The contents of messages may be encrypted (say, by using the WEP algorithm) to prevent eavesdroppers from reading the messages.

• **Data delivery:** IEEE 802.11 naturally provides a way to transmit and receive data. However, like Ethernet, the transmission is not guaranteed to be completely reliable.

1.3 IEEE 802.11 STANDARD

IEEE 802.11 is a prominent standard for WLANs, which is adopted by many vendors of WLAN products. A later version of this standard is the IEEE 802.11b, commercially known as *Wi-Fi* (wireless fidelity). The IEEE 802.11 standard, which deals with the physical and MAC layers in WLANs, was brought out in 1997. It may be observed that IEEE 802.11 was the first WLAN standard that faced the challenge of organizing a systematic approach for defining a standard for wireless wideband local access (small-scale networks capable of transmitting data at high rates). Wireless standards need to have provisions to support mobility of nodes. The IEEE802.11 working group had to examine connection management, link reliability management, and power management — none of which was a concern for other standards in IEEE 802. In addition, provision for security had to be introduced. For all these reasons and because of several competing proposals, it took nearly ten years for the development of IEEE 802.11, which was much longer compared to the time taken for the development of other 802 standards for the wired media. Once the overall picture and the ideas became clear, it took only a reasonable duration of time to develop the IEEE 802.11a and IEEE 802.11b enhancements. Under the IEEE 802.11 standard, MTs can operate in two modes: (i) *infrastructure mode,* in which MTs can communicate with one or more APs which are connected to a WLAN, and (ii) *ad hoc mode,* in which MTs can communicate directly with each other without using an AP.

1.3.1 Physical Layer

IEEE 802.11 supports three options for the medium to be used at the physical level — one is based on infrared and the other two are based on radio transmission. The physical layer is subdivided conceptually into two parts — Physical Medium Dependent sub layer (PMD) and Physical Layer Convergence Protocol (PLCP). PMD handles encoding, decoding, and modulation of signals and thus deals with the idiosyncrasies of the particular medium. The PLCP abstracts the functionality that the physical layer has to offer to the MAC layer. PLCP offers a Service Access Point (SAP) that is independent of the transmission technology, and a Clear Channel Assessment (CCA) carrier sense signal to the MAC layer. The SAP abstracts the channel which can offer up to 1

or 2 Mbps data transmission bandwidth. The CCA is used by the MAC layer to implement the CSMA/CA mechanism.

The three choices for the physical layer in the original 802.11 standard are as follows:

(i) Frequency Hopping Spread Spectrum (FHSS) operating in the license-free 2.4 GHz industrial, scientific, and medical (ISM) band, at data rates of 1 Mbps [using 2-level Gaussian frequency shift keying (GFSK) modulation scheme] and 2 Mbps(using 4-level GFSK);

(ii) Direct Sequence Spread Spectrum (DSSS) operating in the 2.4 GHz ISM band, at data rates of 1 Mbps [using Differential Binary Phase Shift Keying (DBPSK) modulation scheme] and 2 Mbps [using Differential Quadrature Phase Shift Keying (DQPSK)];

(iii) Infrared operating at wavelengths in 850-950 nm range, at data rates of 1 Mbps and 2 Mbps using Pulse Position Modulation (PPM) scheme.

Carrier Sensing Mechanisms

In IEEE 802.3, sensing the channel is very simple. The receiver reads the peak voltage on the cable and compares it against a threshold. In contrast, the mechanism employed in IEEE 802.11 is relatively more complex. It is performed either physically or virtually. As mentioned earlier, the physical layer sensing is through the Clear Channel Assessment (CCA) signal provided by the PLCP in the physical layer of the IEEE 802.11. The CCA is generated based on sensing of the air interface either by sensing the detected bits in the air or by checking the Received Signal Strength (RSS) of the carrier against a threshold. Decisions based on the detected bits are made somewhat more slowly, but they are more reliable. Decisions based on the RSS can potentially create a false alarm caused by measuring the level of interference.

1.3.2 Basic MAC Layer Mechanisms as specified by the IEEE 802.11 standard:

The primary function of this layer is to arbitrate and statistically multiplex the transmission requests of various wireless stations that are operating in an area. This assumes importance because wireless transmissions are inherently broadcast in nature and contentions to access the shared channel need to be resolved prudently in order to avoid collisions, or at least to reduce the number of collisions. The MAC layer also supports many auxiliary functionalities such as offering support for roaming, authentication, and taking care of power conservation. The basic services supported are the mandatory asynchronous data service and an optional real-time service. The asynchronous data service is supported for unicast packets as well as for multicast packets. The real-time service is supported only in infrastructure-based networks where APs control access to the shared medium.

Distributed Foundation Wireless Medium Access Control (DFWMAC)

The primary access method of IEEE 802.11 is by means of a distributed coordination function (DCF). This mandatory basic function is based on a version of carrier sense with multiple access and collision avoidance (CSMA/CA). To avoid the hidden terminal problem, an optional RTS-CTS mechanism is implemented. There is a second method called the Point Coordination Function (PCF) that is implemented to provide real-time services. When the PCF is in operation, the AP controls medium access and avoids simultaneous transmissions by the nodes.

Inter-Frame Spacing (IFS)

Inter-Frame Spacing refers to the time interval between the transmission of two successive frames by any station. There are four types of IFS: SIFS, PIFS, DIFS, and EIFS, in order from shortest to longest. They denote priority levels of access to the medium. Shorter IFS denotes a higher priority to access the medium, because the wait time to access the medium is lower. The exact values of the IFS are obtained from the attributes specified in the Physical Layer

Management Information Base (PHYMIB) and are independent of the station bit rate.

• **Short Inter-Frame Spacing** (**SIFS**) is the shortest of all the IFSs and denotes highest priority to access the medium. It is defined for short control messages such as acknowledgments for data packets and polling responses. The transmission of any packet should begin only after the channel is sensed to be idle for a minimum time period of at least SIFS.

• **PCF Inter-Frame Spacing (PIFS)** is the waiting time whose value lies between SIFS and DIFS. This is used for real-time services.

• **DCF Inter-Frame Spacing (DIFS)** is used by stations that are operating under the DCF mode to transmit packets. This is for asynchronous data transfer within the contention period.

• **Extended Inter-Frame Spacing (EIFS)** is the longest of all the IFSs and denotes the least priority to access the medium. EIFS is used for resynchronization whenever physical layer detects incorrect MAC frame reception.

1.3.3 CSMA/CA Mechanism

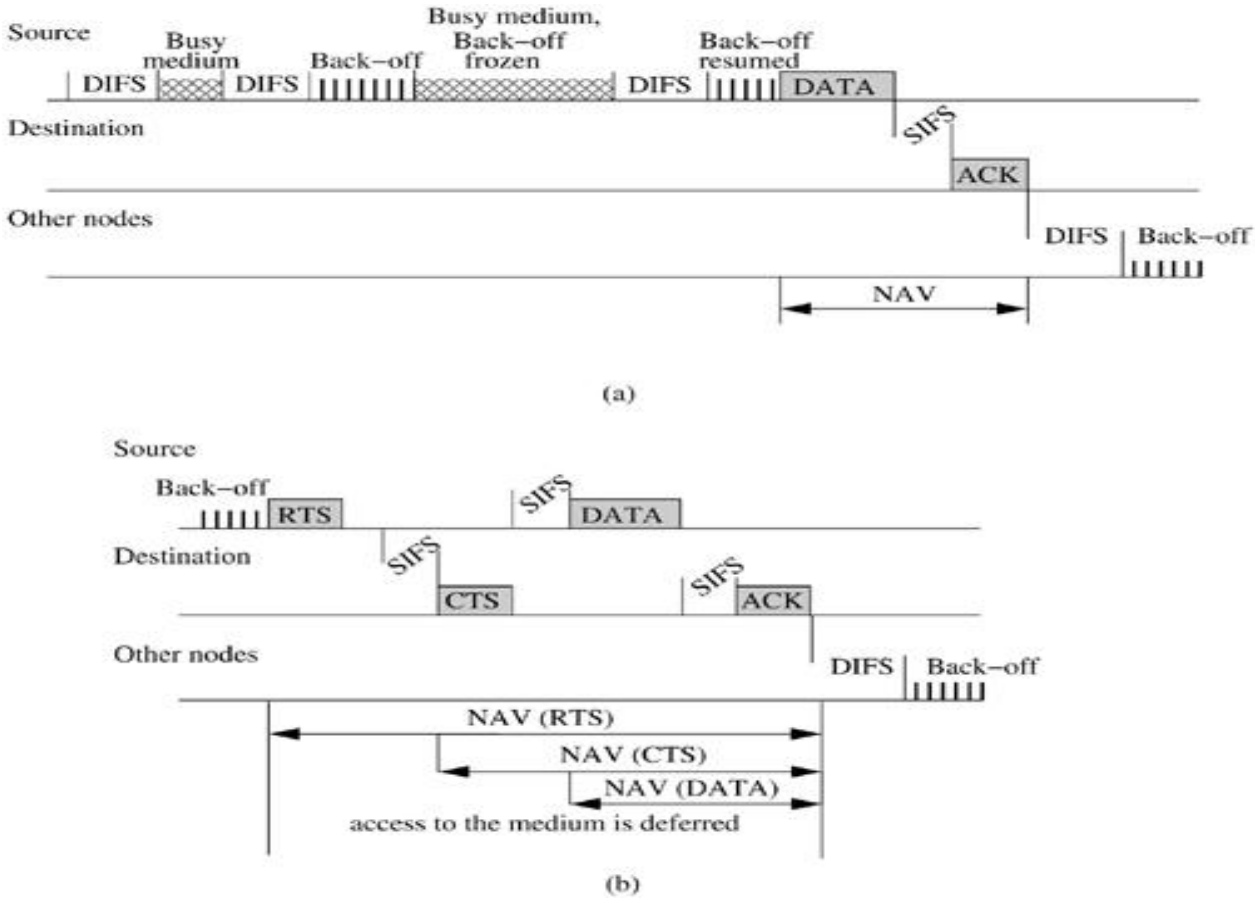
Carrier Sense With Multiple Access And Collision Avoidance (CSMA/CA) is the MAC layer mechanism used by IEEE 802.11 WLANs. Carrier Sense With Multiple Access And Collision Detection (CSMA/CD) is a well-studied technique in IEEE 802.x wired LANs. This technique cannot be used in the context of WLANs effectively because the error rate in WLANs is much higher and allowing collisions will lead to a drastic reduction in throughput. Moreover, detecting collisions in the wireless medium is not always possible. The technique adopted here is therefore one of collision avoidance.

The Medium Access Mechanism

The basic channel access mechanism of IEEE 802.11 is shown in Figure 1.2 (a). If the medium is sensed to be idle for a duration of DIFS, the node accesses the

medium for transmission. Thus the channel access delay at very light loads is equal to the DIFS.

Figure 1.2. IEEE 802.11 DCF and RTS-CTS mechanism.



If the medium is busy, the node *backs off,* in which the station defers channel access by a random amount of time chosen within a *contention window*(*CW*). The value of *CW* can vary between CW_{min} and CW_{max}. The time intervals are all integral multiples of slot times, which are chosen judiciously using propagation delay, delay in the transmitter, and other physical layer dependent parameters. As soon as the back-off counter reaches zero and expires, the station can access the medium. During the back-off process, if anode detects a busy channel, it freezes the back-off counter and the process is resumed once the channel becomes idle for a period of DIFS. Each station executes the back-off procedure at least once between every successive transmission.

In the scheme discussed so far, each station has the same chances for transmitting data next time, independent of the overall waiting time for transmission. Such a system is clearly unfair. Ideally, one would like to give stations that wait longer a higher priority service in order to ensure that they are not starved. The back-off timer incorporated into the above mechanism tries to make it fair. Longer waiting stations, instead of choosing another random interval from the contention window, wait only for a residual amount of time that is specified by the back-off timer.

Contention Window Size

The size of the Contention Window (CW) is another important parameter. If the CW is small in size, then the random values will be close together and there is a high probability of packet collision. On the other hand, if the size of CW is very large, there will be some unnecessary delay because of large back-off values. Ideally, one would like the system to adapt to the current number of stations that are contending for channel access. To effect this, the truncated binary exponential back-off technique is used here, which is similar to the technique used in IEEE 802.3. The initial contention window is set to a random value between (0, CWmin) and each time a collision occurs, the CW doubles its size up to a maximum of CWmax. So at high load, the CW size is high and therefore the resolution power of the system is high. At low loads, small CW ensures low access delay. The specified values of CWmin andCWmax for different physical layer specifications are given in Table 1.1.

Table 2.1. IEEE 802.11 parameters

Parameter	802.11 (FHSS)	802.11 (DSSS)	802.11 (IR)	802.11b	802.11a
t_{slot}	50 μsec	20 μsec	8 μsec	20 μsec	9 μsec
SIFS	28 μsec	10 μsec	10 μsec	10 μsec	16 μsec
PIFS	SIFS $+t_{slot}$				
DIFS	SIFS$+(2 \times t_{slot})$				
Operating Frequency	2.4 GHz	2.4 GHz	850-950 nm	2.4 GHz	5 GHz
Maximum Data Rate	2 Mbps	2 Mbps	2 Mbps	11 Mbps	54 Mbps
CWmin	15	31	63	31	15
CWmax	1,023	1,023	1,023	1,023	1,023

Acknowledgments

Acknowledgments (ACKs) must be sent for data packets in order to ensure their correct delivery. For unicast packets, the receiver accesses the medium after waiting for a SIFS and sends an ACK. Other stations have to wait for DIFS plus their backoff time. This reduces the probability of a collision. Thus higher priority is given for sending an ACK for the previously received data packet than for starting a new data packet transmission. ACK ensures the correct reception of the MAC layer frame by using cyclic redundancy checksum (CRC)technique. If no ACK is received by the sender, then a retransmission takesplace. The number of retransmissions is limited, and failure is reported to the higher layer after the retransmission count exceeds this limit.

RTS-CTS Mechanism

The *hidden terminal problem* is a major problem that is observed in wireless networks. This is a classic example of problems arising due to incomplete topology information in wireless networks that was mentioned initially. It also highlights the non-transitive nature of wireless transmission. In some situations, one node can receive from two other nodes, which cannot hear each other. In such cases, the receiver may be bombarded by both the senders, resulting in collisions and reduced throughput. But the senders, unaware of this, may get the impression that the receiver can clearly listen to them without interference from anyone else. This is called the hidden terminal problem. To alleviate this problem, the RTS-CTS mechanism has been devised as shown in Figure 1.2 (b).

How RTS-CTS Works

The sender sends a request to send (RTS) packet to the receiver. The packet includes the receiver of the next data packet to be transmitted and the expected duration of the whole data transmission. This packet is received by all stations that can hear the sender. Every station that receives this packet will set its *Network Allocation Vector* (NAV) accordingly. The NAV of a station specifies the earliest time when the station is permitted to attempt transmission. After waiting for SIFS, the intended receiver of the data packet answers with a clear to send (CTS) packet if it is ready to accept the data packet. The CTS packet contains the duration field, and all stations receiving the CTS packet also set their NAVs. These stations are within the transmission range of the receiver. The set of stations receiving the CTS packet may be different from the set of stations that received the RTS packet, which indicates the presence of some hidden terminals. Once the RTS packet has been sent and CTS packet has been received successfully, all nodes within receiving distance from the sender and from the receiver are informed that the medium is reserved for one sender exclusively. The sender then starts data packet transmission after waiting for SIFS. The receiver, after receiving the packet, waits for another SIFS and sends the ACK.As soon as the transmission is over, the NAV in each node marks the medium as free (unless the node has meanwhile heard some other RTS/CTS) and the process can repeat again. The RTS packet is like any other packet and collisions can occur only at the beginning when RTS or CTS is being sent. Once the RTS and CTS packets are transmitted successfully, nodes that listen to the RTS or the CTS refrain from causing collision to the ensuing data transmission, because of their NAVs which will be set. The usage of RTS-CTS dialog before data packet transmission is a form of *virtual carrier sensing.*

Overhead Involved in RTS-CTS

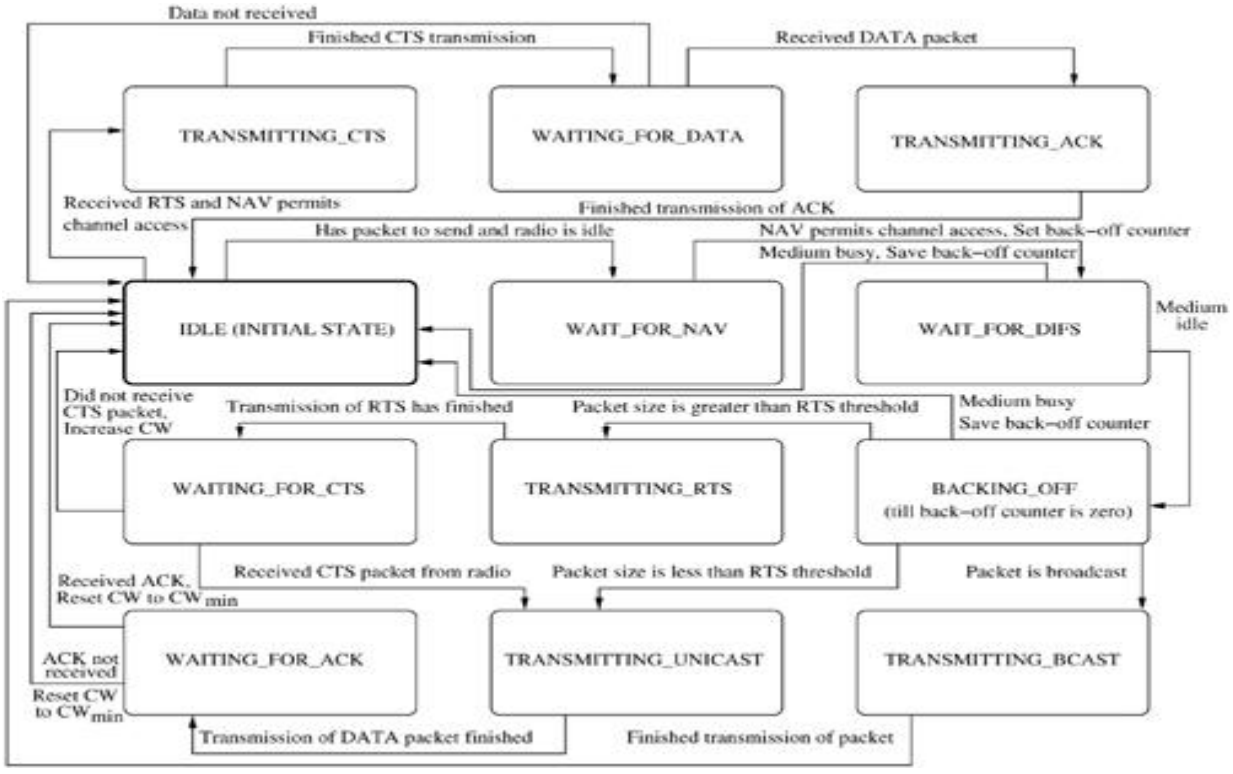
It can be observed that the above mechanism is akin to reserving the medium prior to a particular data transfer sequence in order to avoid collisions during this transfer. But transmission of RTS-CTS can result in non-negligible overhead. Therefore, the RTS-CTS mechanism is used judiciously. An RTS threshold is used to determine whether to start the RTSCTS mechanism or not. Typically, if the frame size is more than the RTS threshold, the RTS-CTS

mechanism is activated and a four-way handshake (*i.e.*, RTS-CTS-DATA-ACK) follows. If the frame size is below the RTS threshold, the nodes resort to a two-way handshake (DATA-ACK).

MAC as a State Machine

Figure 1.3 diagrammatically shows what has been discussed so far. It models the MAC layer as a finite state-machine, and shows the permissible transitions. It must be noted that the state-machine is simplistic and is given only to ease the understanding of the fundamental mechanisms at the MAC layer. The functioning of the finite state-machine is explained in what follows.

Figure 1.3. MAC state transition diagram.



If a node has a packet to send and is in the IDLE state, it goes into the WAIT_FOR_NAV state. After the on-going transmissions (if any) in the neighborhood are over, the node goes to the WAIT_FOR_DIFS state. After waiting for DIFS amount of time, if the medium continues to be idle, the station

enters the BACKING_OFF state. Otherwise, the station sets its back-off counter(if the counter value is zero) and goes back to the IDLE state. During back-off, if the node senses a busy channel, the node saves the back-off counter and goes back to the IDLE state. Otherwise, it goes into one of three states. If the packet type is broadcast, the node enters the TRANSMITTING_BCAST state where it transmits the broadcast packet. If the packet type is unicast and the packet size is less than the RTS threshold, the node enters the TRANSMITTING_UNICAST state and starts transmitting data. If the packet size is greater than the RTS threshold, the node enters the TRANSMITTING_RTS state and starts transmitting the RTS packet. After the RTS transmission is over, the node enters the WAITING_FOR_CTS state. If the CTS packet is not received within a specified time, the node times out and goes back to the IDLE state, and increases the *CW* value exponentially up to a maximum of CW_{max}. If the CTS packet is received, the node enters the TRANSMITTING_UNICAST state and starts transmitting data. After the unicast packet is transmitted, the node enters the WAITING_FOR_ACK state. When the node receives the ACK, it goes back to the IDLE state and reduces the CW value to CW_{min}.If a node receives an RTS packet when in IDLE state and if the NAV of the node indicates that no other on-going transmissions exist, the node enters the TRANSMITTING_CTS state and starts transmitting the CTS packet. After the CTS packet is transmitted, the node enters the WAITING_FOR_DATA state and waits for the data packet from the sender. On receiving the data packet, the node enters the TRANSMITTING_ACK state and starts transmitting the ACK for the data packet. When the ACK has been transmitted, the node goes back to the IDLE state. If the data packet is not received, the receiver returns to the IDLE state.

Fragmentation

Bit error rates in the wireless medium are much higher than in other media. The bit error rate in fiber optics is only about 10^{-9}, whereas in wireless, it is as large as 10^{-4}. One way of decreasing the frame error rate is by using shorter frames. IEEE 802.11 specifies a fragmentation mode where user data packets are split into several smaller parts transparent to the user. This will lead to shorter

frames, and frame error will result in retransmission of a shorter frame. The RTS and CTS messages carry duration values for the current fragment and estimated time for the next fragment. The medium gets reserved for the successive frames until the last fragment is sent. The length of each fragment is the same for all the fragments except the last fragment. The fragments contain information to allow the complete MAC Protocol Data Unit (MPDU, informally referred to as packet) to be reassembled from the fragments that constitute it. The frame type, sender address, destination address, sequence control field, and indicator for more fragments to come are all present in the fragment header. The destination constructs the complete packet by reassembling the fragments in the order of the sequence number field. The receiving station ensures that all duplicate fragments are discarded and only one copy of each fragment is integrated. Acknowledgments for the duplicates may, however, be sent.

1.3.4 Other MAC Layer Functionalities

There are several other functionalities that the MAC layer provides in IEEE 802.11 WLANs. The functionalities are the Point Coordination Function (PCF) which is used for QoS guarantees, Timing Synchronization, Power Management, and Support For Roaming.

Point Coordination Function

The objective of the point coordination function (PCF) is to provide guarantees on the maximum access delay, minimum transmission bandwidth, and other QoS parameters. Unlike the DCF, where the medium contention is resolved in a distributed manner, the PCF works by effecting a centralized contention resolution scheme, and is applicable only in networks where an AP polls the nodes in its BSS. A point coordinator (PC) at the AP splits the access time into super frame periods. The super frame period consists of alternating contention free periods (CFPs) and contention periods (CPs). The PC will determine which station has the right to transmit at any point of time. The PCF is essentially a polled service with the PC playing the role of the polling master. The operation of the PCF may require additional coordination to perform efficient operation in cases where multiple PCs are operating simultaneously such that their

transmission ranges overlap. The IFS used by the PCF is smaller than the IFS of the frames transmitted by the DCF. This means that point-coordinated traffic will have higher priority access to the medium if DCF and PCF are concurrently in action. The PC controls frame transmissions so that contentions are eliminated over a limited period of time, that is, the CFP.

Synchronization

Synchronization of clocks of all the wireless stations is an important function to be performed by the MAC layer. Each node has an internal clock, and clocks are all synchronized by a Timing Synchronization Function (TSF). Synchronized clocks are required for power management, PCF coordination, and Frequency Hopping Spread Spectrum (FHSS) hopping sequence synchronization. Without synchronization, clocks of the various wireless nodes in the network may not have a consistent view of the global time. Within a BSS, quasi periodic beacon frames are transmitted by the AP, that is, one beacon frame is sent every Target Beacon Transmission Time (TBTT) and the transmission of a beacon is deferred if the medium is busy. A beacon contains a time-stamp that is used by the node to adjust its clock. The beacon also contains some management information for power optimization and roaming. Not all beacons need to be heard for achieving synchronization.

Power Management

Usage of power cords restricts the mobility that wireless nodes can potentially offer. The usage of battery-operated devices calls for power management because battery power is expensive. Stations that are always ready to receive data consume more power (the receiver current may be as high as 100 mA). The transceiver must be switched off whenever carrier sensing is not needed. But this has to be done in a manner that is transparent to the existing protocols. It is for this reason that power management is an important functionality in the MAC layer. Therefore, two states of the station are defined: sleep and awake. The sleep state refers to the state where the transceiver cannot receive or send wireless signals. Longer periods in the sleep state mean that the average throughput will be low. On the other hand, shorter periods in the sleep state

consume a lot of battery power and are likely to reduce battery life. If a sender wants to communicate with a sleeping station, it has to buffer the data it wishes to send. It will have to wait until the sleeping station wakes up, and then send the data. Sleeping stations wake up periodically, when senders can announce the destinations of their buffered data frames. If any node is a destination, then that node has to stay awake until the corresponding transmission takes place.

Roaming

Each AP may have a range of up to a few hundred meters where its transmission will be heard well. The user may, however, walk around so that he goes from the BSS of one AP to the BSS of another AP. Roaming refers to providing uninterrupted service when the user walks around with a wireless station. When the station realizes that the quality of the current link is poor, it starts scanning for another AP. This scanning can be done in two ways: active scanning and passive scanning. Active scanning refers to sending a probe on each channel and waiting for a response. Passive scanning refers to listening into the medium to find other networks. The information necessary for joining the new BSS can be obtained from the beacon and probe frames.

1.3.5 Other Issues

Improvements in the IEEE 802.11 standard have been proposed to support higher data rates for voice and video traffic. Also, QoS provisioning and security issues have been addressed in extended versions of the standard.

Newer Standards

The original standards for IEEE 802.11 came out in 1997 and promised a data rate of 1-2 Mbps in the license-free 2.4 GHz ISM band. Since then, several improvements in technology have called for newer and better standards that offer higher data rates. This has manifested in the form of IEEE802.11aand IEEE 802.11b standards, both of which came out in 1999. IEEE 802.11b, an extension of IEEE 802.11 DSSS scheme, defines operation in the 2.4GHz ISM band at data rates of 5.5 Mbps and 11 Mbps, and is trademarked commercially by the Wireless Ethernet Compatibility Alliance (WECA) as Wi-Fi. It achieves

high data rates due to the use of Complimentary Code Keying(CCK). IEEE 802.11a operates in the 5 GHz band (unlicensed national information infrastructure band), and uses orthogonal frequency division multiplexing (OFDM) at the physical layer. IEEE 802.11a supports data rates up to 54 Mbps and is the fast Ethernet analogue to IEEE 802.11b.Other IEEE 802.11 (c, d, and h) task groups are working on special regulatory and networking issues. IEEE 802.11e deals with the requirements of time sensitive applications such as voice and video. IEEE802.11f deals with inter-AP communication to handle roaming. IEEE 802.11g aims at providing the high speed of IEEE 802.11a in the ISM band. IEEE 802.11i deals with advanced encryption standards to support better privacy.

QoS for Voice and Video Packets

In order to offer QoS, delay-sensitive packets (such as voice and video packets) are to be given a higher priority to get ahead of less time-critical (*e.g.,* file transfer) traffic. Several mechanisms have been proposed to offer weighted priority. Hybrid Coordination Function (HCF) can be used where the AP polls the stations in a weighted way in order to offer QoS. Extended DCF is another mechanism which has been proposed where the higher priority stations will choose the random back-off interval from a smaller CW. Performance of WLANs where voice and data services are integrated.

Wired Equivalent Privacy

Security is a very important issue in the design of WLANs. In order to provide a modest level of physical security, the Wired Equivalent Privacy (WEP) mechanism was devised. The name WEP implies that this mechanism is aimed at providing the level of privacy that is equivalent to that of a wired LAN. Data integrity, access control, and confidentiality are the three aims of WEP. It assumes the existence of an external key management service that distributes the key sequence used by the sender. This mechanism relies on the fact that the secret key cannot be determined by brute force. However, WEP has been proven to be vulnerable if more sophisticated mechanisms are used to crack the key. It uses the pseudo-random number key generated by RSA RC4 algorithm which

has been efficiently implemented in hardware as well as in software. This mechanism makes use of the fact that if we take the plain text, XOR (bit-by-bit exclusive OR) it with a pseudo-random key sequence, and then XOR the result with the same key sequence, we get back the plain text.

1.4 HIPERLAN STANDARD

The European counterparts to the IEEE 802.11 standards are the high performance radio LAN(HIPERLAN) standards defined by the European Telecommunications Standards Institute (ETSI). It is to be noted that while the IEEE 802.11 standards can use either radio access or infrared access, the HIPERLAN standards are based on radio access only. The standards have been defined as part of the ETSI Broadband Radio Access Networks (BRAN) project. In general, broadband systems are those in which user data rates are greater than 2Mbps (and can go up to 100s of Mbps). Four standards have been defined for wireless networks by the ETSI.

• HIPERLAN/1 is a wireless radio LAN (RLAN) without a wired infrastructure, based on one-to-one and one-to-many broadcasts. It can be used as an extension to a wired infrastructure, thus making it suited to both ad hoc and infrastructure based networks. It employs the 5.15 GHz and the 17.1 GHz frequency bands and provides a maximum data rate of 23.5 Mbps.

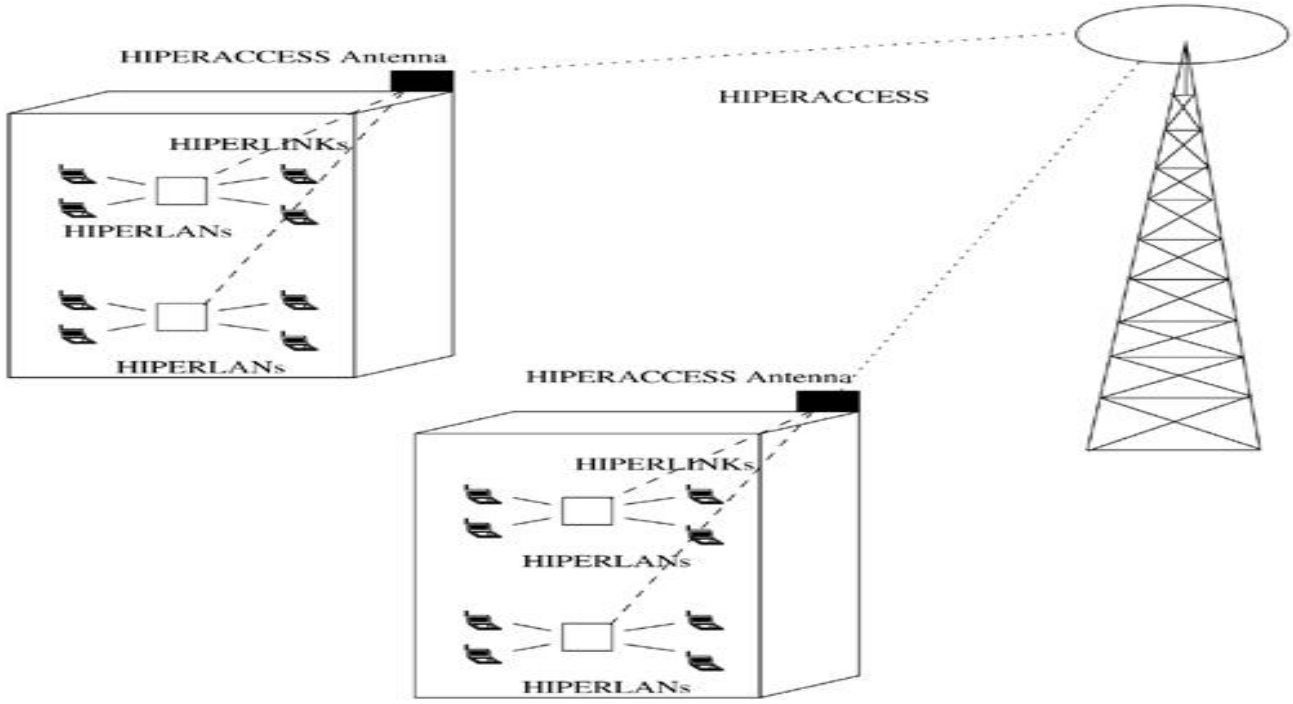
• The HIPERLAN/2 standard intends to provide short-range (up to 200 m)wireless access to Internet Protocol (IP), Asynchronous Transfer Mode (ATM),and other infrastructure-based networks and, more importantly, to integrate WLANs into cellular systems. It employs the 5 GHz frequency band and offers a wide range of data rates from 6 Mbps to 54 Mbps. HIPERLAN/2 has been designed to meet the requirements of future wireless multimedia services.

• ATM networks are connection-oriented and require a connection to set up prior to transfer of information from a source to a destination. All information to be transmitted — voice, data, image, and video — is first fragmented into small, fixed-size packets known as cells. These cells are then switched and routed using packet switching principles.

• HIPERACCESS (originally called HIPERLAN/3) covers "the last mile" to the customer; it enables establishment of outdoor high-speed radio access networks, providing fixed radio connections to customer premises. HIPERACCESS provides a data rate of 25 Mbps. It can be used to connect HIPERLAN/2deployments that are located far apart (up to 5 Km away). It offers point-to multi point communication.

•The HIPERLINK (originally called HIPERLAN/4) standard provides high speed radio links for point-to-point static interconnections. This is used to connect different HIPERLAN access points or HIPERACCESS networks with high-speed links over short distances of up to 150 m. For example, the HIPERLINK can be employed to provide links between different rooms or floors within a large building. HIPERLINK operates on the 17 GHz frequency range. Figure 1.4 shows a typical deployment of the ETSI standards. The standards excluding HIPERLAN/1 are grouped under the BRAN project. The scope of the BRAN has been to standardize the radio access network and the functions that serve as the interface to the infrastructural networks.

Figure 2.10. The ETSI-BRAN systems.



1.4.1 HIPERLAN/1HIPERLAN/1 is a RLAN standard that was introduced by the ETSI in 1995.The standard allows nodes to be deployed either in a pre-arranged or in an adhoc fashion. Apart from supporting node mobility, HIPERLAN/1 provides forwarding mechanisms (multi-hop routing). Thus, coverage is not limited to just the neighboring nodes. Using a clever framing scheme as explained later in this section, HIPERLAN/1 provides a data rate of around 23.5 Mbps without utilizing much power, thus having the capability to support multimedia data and asynchronous data effectively. This data rate is significantly higher than that provided by IEEE 802.11. The HIPERLAN/1 protocol stack is restricted to the two lower-most layers in the OSI reference model: the data link layer (DLL)and the physical layer. The DLL is further divided into the medium access control (MAC) sublayer and the channel access control (CAC) sublayer. The sections that follow describe the standard.

The Physical Layer

The tasks of the physical layer are modulation and demodulation of a radio carrier with a bit stream, forward error-correction mechanisms, signal strength measurement, and synchronization between the sender and the receiver. The standard uses the CCA scheme (similar to IEEE 802.11) to sense whether the channel is idle or busy.

The MAC Sublayer

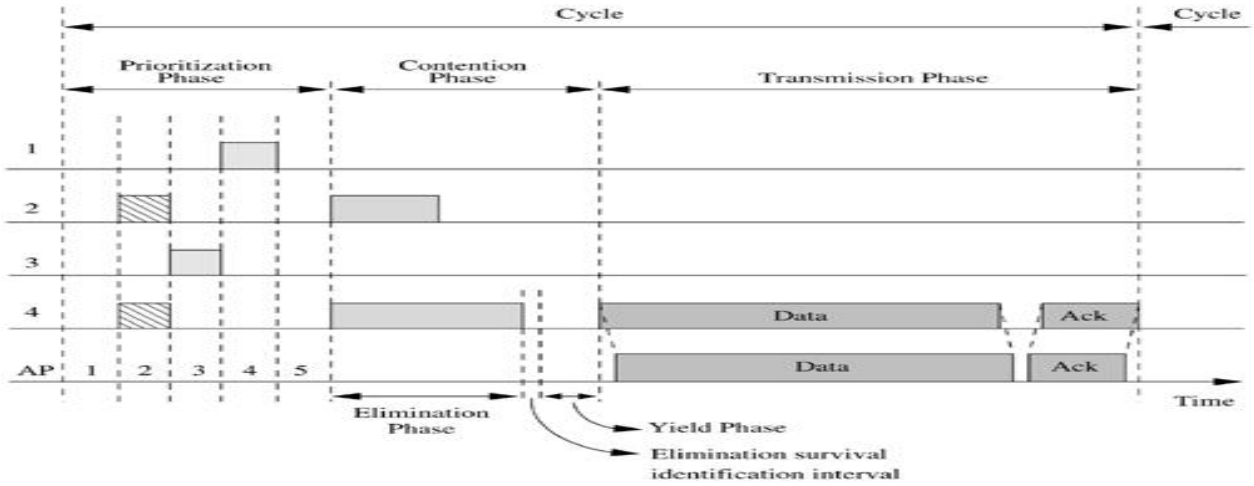
The HIPERLAN/1 MAC (HM) sublayer is responsible for processing the packets from the higher layers and scheduling the packets according to the QoS requests from the higher layers specified by the HMQoS parameters. The MAC sublayer is also responsible for forwarding mechanisms, power conservation schemes, and communication confidentiality through encryption–decryption mechanisms. Because of the absence of an infrastructure, the forwarding mechanism is needed to allow the physical extension of HIPERLAN/1 to go beyond the radio range of a single station. Topology-related data are exchanged between the nodes periodically with the help of special packets, for the purpose of forwarding. In order to guarantee a time-bound service, the HM protocol data unit (HMPDU)selected for channel access has to reflect the user priority and the residual lifetime of the packet (the time remaining for the packet to expire).The

MAC layer computes the channel access priority for each of the PDUs following a mapping from the MAC priority to the channel access mechanism(CAM) priority. One among those PDUs which has the highest CAM priority and the least residual time will be selected for access to the channel.

The CAC Sublayer

The CAC sublayer offers a connectionless data service to the MAC sublayer. The MAC layer uses this service to specify a priority (called the CAM priority)which is the QoS parameter for the CAC layer. This is crucial in the resolution of contention in the CAM.EY-NPMA After a packet with an associated CAM priority has been chosen in the CAC sublayer for transmission, the next phase is to compete with packets of other nodes for channel access. The channel access mechanism is a dynamic, listen-and-then-talk protocol that is very similar to the CSMA/CA used in 802.11 and is called the elimination yield non-pre-emptive multiple access (EYNPMA) mechanism. Figure 1.5 shows the operation of the EY-NPMA mechanism in which the nodes 1, 2, 3, and 4 have packets to be sent to the AP. The CAM priority for nodes 2 and 4 is higher with priority 2 followed by node 3with priority 3, and node 1 with the least priority of 4. The prioritization phase will have k slots where k (can vary from 1 to 5 with $k - 1$ having higher priority than k) refers to the number of priority levels.

Figure 1.5. The operation of EY-NPMA.



The entire process of channel access occurs in the form of channel access cycles. A synchronization interval occurs after the end of every such cycle. This access cycle is comprised of three phases: prioritization, contention, and transmission.

1. **Prioritization:** This phase culls out nodes with packets of the highest CAM priority and lets them participate in the next phase. The prioritization phase consists of two events, namely, priority detection and priority assertion. During the priority detection period, a node listens to the channel for a number of time slots proportional to the CAM priority assigned to the packet that the node wants to send. In Figure 1.5, the nodes 2 and 4 wait for one slot and assert their priority in the second slot as they hold packets with higher priority, and nodes 3 and 1 wait for slots equal to their priority level. By listening to the channel, nodes 3 and 1detect the existence of other nodes with higher priority and hence leave the prioritization phase. If a low-priority node has succeeded in waiting up to this slot, it enters the priority assertion period during which it sends a burst, signaling its selection to the next stage. In this process, the node(s) with the highest CAM priority will finish the prioritization phase first and hence will be selected for the next phase.

2. **Contention:** This phase is to eliminate as many nodes as possible, in order to minimize the collision rate during transmission. This phase extends to a maximum of 13 slots, each of the same width as that of the slots in the prioritization phase. In this phase, the nodes that transmitted a burst in the previous phase, resolve access to the channel by contention. This phase consists of two sub-phases, namely, the elimination phase and the yield phase. Nodes in this phase (nodes 2 and 4 in Figure 1.5) get to transmit a burst for a geometrically distributed number of time slots the probability of a node's transmission extending to a slot length of k slots(where $k < 12$ slots) is 0.5_{k+1}] which is then followed by a sensing period of 1 slot. During this period, if a node detects another node's burst, it stops the contention process (node 2 in Figure 1.5). This period during which each contending node will have to listen to the channel for a slot duration is called the elimination survival identification interval. If the channel is sensed idle during this interval, the node reaches the

yield phase. This period is also called elimination survival verification. This ensures that the node(s) which sent the elimination burst for the maximum number of slots will be chosen for the next phase. The next phase is the yield phase which complements the elimination phase; it involves each node listening to the channel for a number of time slots (up to a maximum of 15 slots, each with duration of the slot duration in the prioritization phase).This is in fact similar to the back-off state in which the probability of backing off for k slots is $0.1 \times 0.9_k$. If the channel is sensed to be idle during these slots, the node is said to be eligible for transmission. The node that waits for the shorter number of slots initiates transmission and other nodes defer their access to the next cycle to begin the process a fresh.

3. **Transmission:** This is the final stage in the channel access where the transmission of the selected packet takes place. During this phase, the successful delivery of a data packet is acknowledged with an ACK packet. The performance of EY-NPMA protocol suffers from major factors such as packet length, number of nodes, and the presence of hidden terminals. The efficiency of this access scheme varies from 8% to 83% with variation of packet sizes from 50 bytes to 2 Kbytes. The above-described channel access takes place during what is known as the channel synchronization condition. The other two conditions during which channel access can take place are

(a) the channel free condition, when the node senses the channel free for some amount of time and then gains access, and

(b) the hidden terminal condition, when a node is eliminated from contention, but still does not sense any data transmission, indicating the presence of a hidden node.

Power Conservation Issues

The HIPERLAN/1 standard has suggested power conservation schemes at both the MAC and the physical layers. At the MAC level, the standard suggests awake/sleep modes similar to the DFWMAC in IEEE 802.11. Two roles defined for the nodes are the p-savers (nodes that want to implement the function) and the p-supporters (neighbors to the p-saver that are deputized to aid the latter's

power conservation). The psaver can receive packets only at predetermined time intervals and is active only during those intervals, in the process saving power. At the physical level, a framing scheme has been adopted to conserve power. The physical burst is divided into High Bit Rate (HBR) and Low Bit Rate (LBR)bursts. The difference between the two bursts lies in the keying mechanisms employed for them – the HBR burst is based on Gaussian Minimum Shift Keying(GMSK) that yields a higher bit rate, but consumes more power than Frequency Shift Keying (FSK) used for the LBR bursts. The LBR burst contains the destination address of the frame and precedes the HBR burst. Any node receiving a packet, first reads the LBR burst. The node will read the HBR burst only if it is the destination for that frame. Otherwise, the burst is simply ignored, thereby saving the power needed to read the HBR burst.

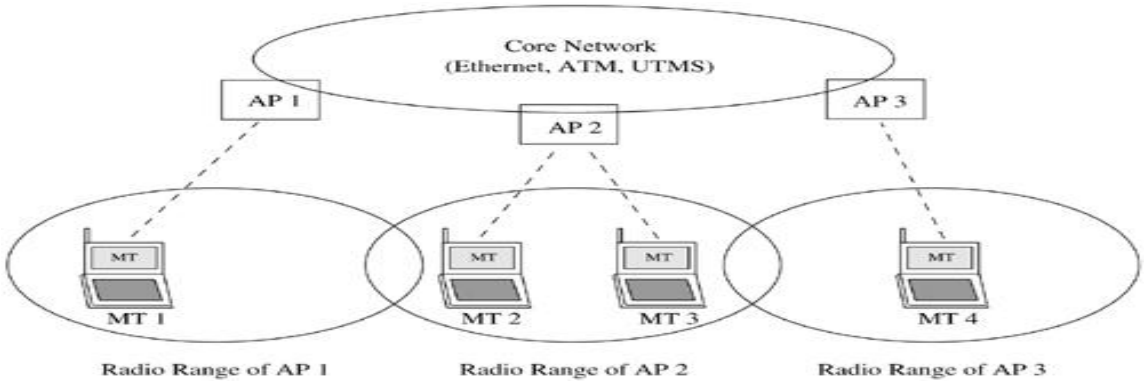
Failure of HIPERLAN/1

In spite of the high data rate that it promised, HIPERLAN/1 standard has always been considered unsuccessful. This is because IEEE Ethernet had been prevalent and hence, for its wireless counterpart too, everybody turned toward IEEE, which came out with its IEEE 802.11 standard. As a result, hardly any manufacturer adopted the HIPERLAN/1 standard for product development. However, the standard is still studied for the stability it provides and for the fact that many of the principles followed have been adopted in the other standards.

1.4.2 HIPERLAN/2 **T**he IEEE 802.11 standard offers data rates of 1 Mbps while the newer standard IEEE802.11a offers rates up to 54 Mbps. However, there was a necessity to support QoS, handoff (the process of transferring an MT from one channel/AP to another), and data integrity in order to satisfy the requirements of wireless LANs. This demand was the motivation behind the emergence of HIPERLAN/2. The standard has become very popular owing to the significant support it has received from cellular manufacturers such as Nokia and Ericsson. The HIPERLAN/2 tries to integrate WLANs into the next-generation cellular systems. It aims at converging IP and ATM type services at a high data rate of 54 Mbps for indoor and outdoor applications. The HIPERLAN/2, an ATM compatible WLAN, is a connection-oriented system, which uses fixed size packets and enables QoS applications easy to implement.

The HIPERLAN/2 network has a typical topology as shown in Figure 1.6. The figure shows MTs being centrally controlled by the APs which are in turn connected to the core network (infrastructure-based network). It is to be noted that, unlike the IEEE standards, the core network for HIPERLAN/2 is not just restricted to Ethernet. Also, the AP used in HIPERLAN/2 consists of one or many transceivers called Access Point Transceivers (APTs) which are controlled by a single Access Point Controller (APC).

Figure 1.6. A typical deployment of HIPERLAN/2.



There are two modes of communication in a HIPERLAN/2 network, which are described by the following two environments:

• **Business environment:** The ad hoc architecture of HIPERLAN/1 has been extended to support a centralized mode of communication using APs. This topology corresponds to business environments. Accordingly, each AP serves a number of MTs.

• **Home environment:** The home environment enables a direct mode of communication between the MTs. This corresponds to an ad hoc architecture that can be operated in a plug-and-play manner. The direct mode of communication is, however, managed by a central control entity elected from among the nodes called the central controller (CC).There are several features of HIPERLAN/2 that have attracted many a cellular manufacturer. These features are part of the discussion on the protocol stack of HIPERLAN/2 below. The HIPERLAN/2 protocol stack consists of the physical layer, convergence layer (CL), and the data link control (DLC) layer.

The Physical Layer

The physical layer is responsible for the conversion of the PDU train from the DLC layer to physical bursts that are suitable for radio transmission. HIPERLAN/2, like IEEE 802.11a, uses OFDM for transmission. The HIPERLAN/2 allows bit rates from 6 Mbps to 54 Mbps using a scheme called link adaptation. This scheme allows the selection of a suitable modulation method for the required bit rate. This scheme is unique to HIPERLAN/2 and is not available in the IEEE standards and HIPERLAN/1. More details on the physical layer can be found in.

The CL

The topmost layer in the HIPERLAN/2 protocol stack is the CL. The functions of the layer are to adapt the requirements of the different higher layers of the core network with the services provided by the lower layers of HIPERLAN/2, and to convert the higher layer packets into ones of fixed size that can be used by the lower layers. A CL is defined for every type of core network supported. In short, this layer is responsible for the network-independent feature of HIPERLAN/2. The CL is classified into two types, namely, the packet-based CL and the cell based CL. The packet-based CL processes variable-length packets (such as IEEE 802.3, IP, and IEEE 1394). The cell-based CL processes fixed sized ATM cells. The CL has two sublayers, namely, the common part (CP) and the service-specific convergence sublayer (SSCS). The CP is independent of the core network. It allows parallel segmentation and reassembly of packets. The CP comprises of two sublayers, namely, the common part convergence sublayer (CPCS) and the segmentation and reassembly (SAR) sublayer. The CPCS processes the packets from the higher layer and adds padding and additional information, so as to be segmented in the SAR. For further information on the CP, readers are referred to. The SSCS consists of functions that are specific to the core network. For example, the Ethernet SSCS has been standardized in for Ethernet core networks. The SSCS adapts the different data formats to the HIPERLAN/2 DLC format. It is also responsible for mapping the QoS requests of the higher layers to the QoS parameters of HIPERLAN/2 such as data rate, delay, and jitter.

The DLC Layer

The DLC layer constitutes the logical link between the AP and the MTs. This ensures a connection-oriented communication in a HIPERLAN/2 network, in contrast to the connectionless service offered by the IEEE standards. The DLC layer is organized into three functional units, namely, the Radio Link Control (RLC) sublayer on the control plane, the error control (EC) sublayer on the user plane, and the MAC sublayer.

The RLC Sublayer

The RLC sublayer takes care of most of the control procedures on the DLC layer. The tasks of the RLC can be summarized as follows.

• **Association Control Function (ACF):** The ACF handles the registration and the authentication functions of an MT with an AP within a radio cell. Only after the ACF procedure has been carried out can the MT ever communicate with the AP.

• **DLC user Connection Control (DCC):** The DCC function is used to control DLC user connections. It can set up new connections, modify existing connections, and terminate connections.

• **Radio Resource Control (RRC):** The RRC is responsible for the surveillance and efficient utilization of the available frequency resources.

It performs the following tasks:

Dynamic frequency selection: This function is not available in IEEE 802.11, IEEE 802.11a, IEEE802.11b, and HIPERLAN/1, and is thus unique to HIPERLAN/2. It allows the AP to select a channel (frequency) for communication with the MTs depending on the interferences in each channel, thereby aiding in the efficient utilization of the available frequencies.

Handoff: HIPERLAN/2 supports three types of handoff, namely, sector handoff(moving to another sector of the same antenna of an APT), radio handoff(handoff between two APTs under the same APC), and network handoff(handoff between two APs in the same network).

Power saving: Power-saving schemes much similar to those in HIPERLAN/1and IEEE 802.11 have been implemented.

Error Control (EC)

Selective repeat (where only the specific damaged or lost frame is retransmitted)protocol is used for controlling the errors across the medium. To support QoS for stringent and delay-critical applications, a discard mechanism can be provided by specifying a maximum delay.

The MAC Sublayer

The MAC protocol is used for access to the medium, resulting in the transmission of data through that channel. However, unlike the IEEE standards and the HIPERLAN/1 in which channel access is made by sensing it, the MAC protocol follows a dynamic time division multiple access/time division duplexing (TDMA/TDD) scheme with centralized control. The protocol supports both AP-MT unicast and multicast transfer, and at the same time MT-MT peer-to-peer communication. The centralized AP scheduling provides QoS support and collision-free transmission. The MAC protocol provides a connection-oriented communication between the AP and the MT (or between MTs).

Security Issues

Elaborate security mechanisms exist in the HIPERLAN/2 system. The encryption procedure is optional and can be selected by the MT during association. Two strong encryption algorithms are offered, namely, the Data Encryption Standard (DES) and the triple-DES algorithms

1.5 BLUETOOTH

WLAN technology enables device connectivity to infrastructure-based services through a wireless carrier provider. However, the need for personal devices to communicate wirelessly with one another, without an established infrastructure, has led to the emergence of Personal Area Networks (PANs). The first attempt to define a standard for PANs dates back to Ericsson's Bluetooth project in 1994 to

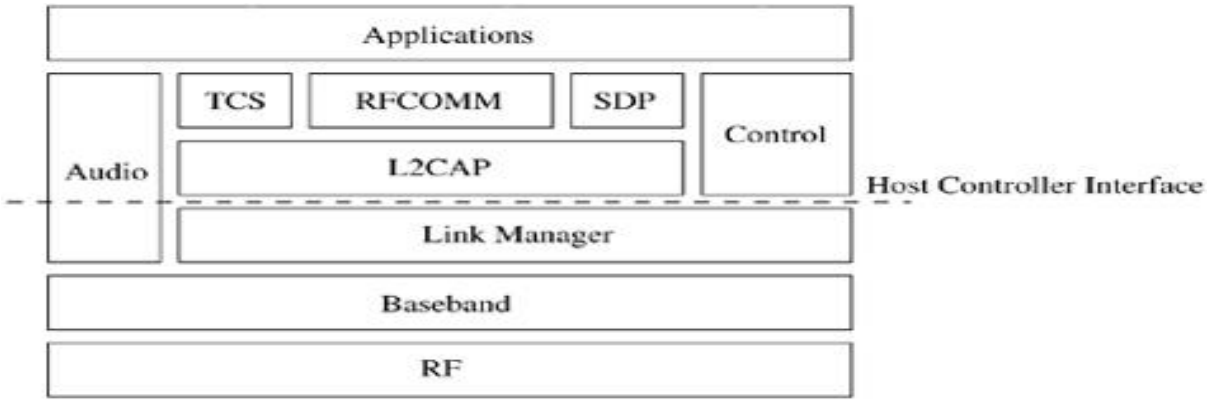
enable communication between mobile phones using low power and low-cost radio interfaces. In May 1998, several companies such as Intel, IBM, Nokia, and Toshiba joined Ericsson to form the Bluetooth Special Interest Group (SIG) whose aim was to develop a *de facto* standard for PANs. Recently, IEEE has approved a Bluetooth-based standard (IEEE 802.15.1) for wireless personal area networks (WPANs). The standard covers only the MAC and the physical layers while the Bluetooth specification details the whole protocol stack. Bluetooth employs radio frequency (RF) technology for communication. It makes use of frequency modulation to generate radio waves in the ISM band. Low power consumption of Bluetooth technology and an offered range of up to ten meters has paved the way for several usage models. One can have an interactive conference by establishing an ad hoc network of laptops. Cordless computer, instant postcard [sending digital photographs instantly (a camera is cordlessly connected to a mobile phone)], and three-in-one phone [the same phone functions as an intercom (at the office, no telephone charge), cordless phone (at home, a fixed-line charge), and mobile phone (on the move, a cellular charge)] are other indicative usage models.

1.5.1 Bluetooth Specifications

The Bluetooth specification consists of two parts: core and profiles. The core provides a common data link and physical layer to application protocols, and maximizes reusability of existing higher layer protocols. The profiles specifications classify Bluetooth applications into thirteen types. The protocol stack of Bluetooth performs the functions of locating devices, connecting other devices, and exchanging data. It is logically partitioned into three layers, namely, the transport protocol group, the middleware protocol group, and the application group. The transport protocol group consists of the radio layer, baseband layer, link manager layer, logical link control and adaptation layer, and the host controller interface. The middleware protocol group comprises of RFCOMM, SDP, and IrDA (IrOBEX and IrMC). The application group consists of applications (profiles) using Bluetooth wireless links, such as the modem dialer and the Web-browsing client.

Figure1.7 shows the protocol stack of Bluetooth.

Figure 1.7. Bluetooth protocol stack.



1.5.2 Transport Protocol Group This group is composed of the protocols designed to allow Bluetooth devices to locate each other and to create, configure, and manage the wireless links. Design of various protocols and techniques used in Bluetooth communications has been done with the target of low power consumption and ease of operation. This shall become evident in the design choice of FHSS and the master–slave architecture. The following sections study the various protocols in this group, their purpose, their modes of operation, and other specifications.

Radio (Physical) Layer

The radio part of the specification deals with the characteristics of the transceivers and design specifications such as frequency accuracy, channel interference, and modulation characteristics. The Bluetooth system operates in the globally available ISM frequency band and the frequency modulation is GFSK. It supports 64 Kbps voice channels and asynchronous data channels with a peak rate of 1 Mbps. The data channels are either asymmetric (in one direction) or symmetric (in both directions). The Bluetooth transceiver is a FHSS system operating over a set of *m* channels each of width 1 MHz. In most of the countries, the value of *m* is 79. Frequency hopping is used and hopsare made at a rapid rate across the possible 79 hops in the band, starting at 2.4GHz and stopping at 2.480 GHz. The choice of frequency hopping has been made to provide protection against interference. The Bluetooth air interface is based on a nominal antenna power of 0 dBm (1mW) with extensions for operating at up to 20 dBm (100 mW) worldwide. The nominal link range is from 10 centimetres to

10 meters, but can be extended to more than 100 meters by increasing the transmit power (using the 20 dBm option). It should be noted here that a WLAN cannot use an antenna power of less than 0 dBm (1 mW) and hence an 802.11 solution might not be apt for power-constrained devices.

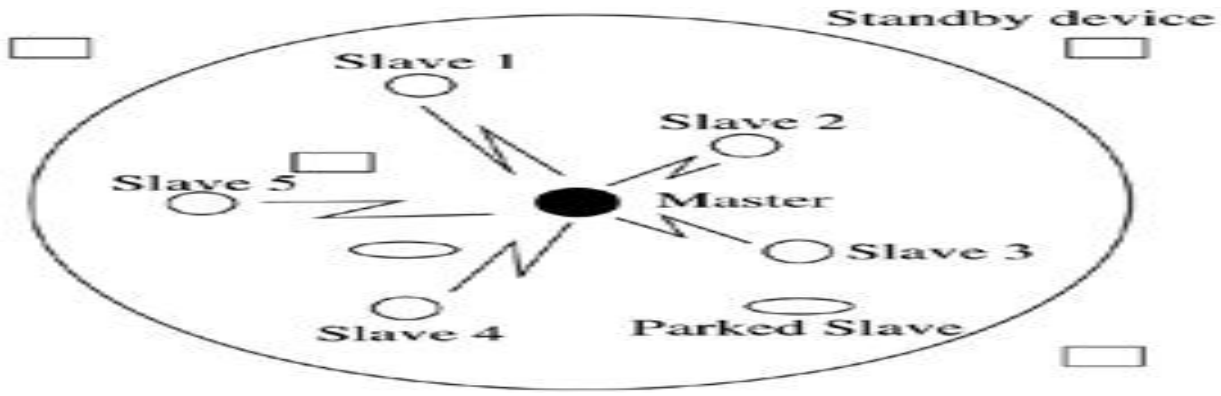
Baseband Layer

The key functions of this layer are frequency hop selection, connection creation, and medium access control. Bluetooth communication takes place by adhoc creation of a network called a *piconet.* The address and the clock associated with each Bluetooth device are the two fundamental elements governing the formation of a piconet. Every device is assigned a single 48-bit address which is similar to the addresses of IEEE 802.xx LAN devices. The address field is partitioned into three parts and the lower address part (LAP) is used in several baseband operations such as piconet identification, error checking, and security checks. The remaining two parts are proprietary addresses of the manufacturing organizations. LAP is assigned internally by each organization. Every device also has a 28-bit clock (called the *native clock*) that ticks 3,200 times per second or once every 312.5 μs. It should be noted that this is twice the normal hopping rate of 1,600 hops per second.

Piconet

The initiator for the formation of the network assumes the role of the *master* (of the piconet). All the other members are termed as *slaves* of the piconet. A piconet can have up to seven active slaves at any instant. For the purpose of identification, each active slave of the piconet is assigned a locally unique active member address AM_ADDR. Other devices could also be part of the piconet by being in the parked mode (explained later). A Bluetooth device not associated with any piconet is said to be in standby mode. Figure 1.8 shows a piconet with several devices.

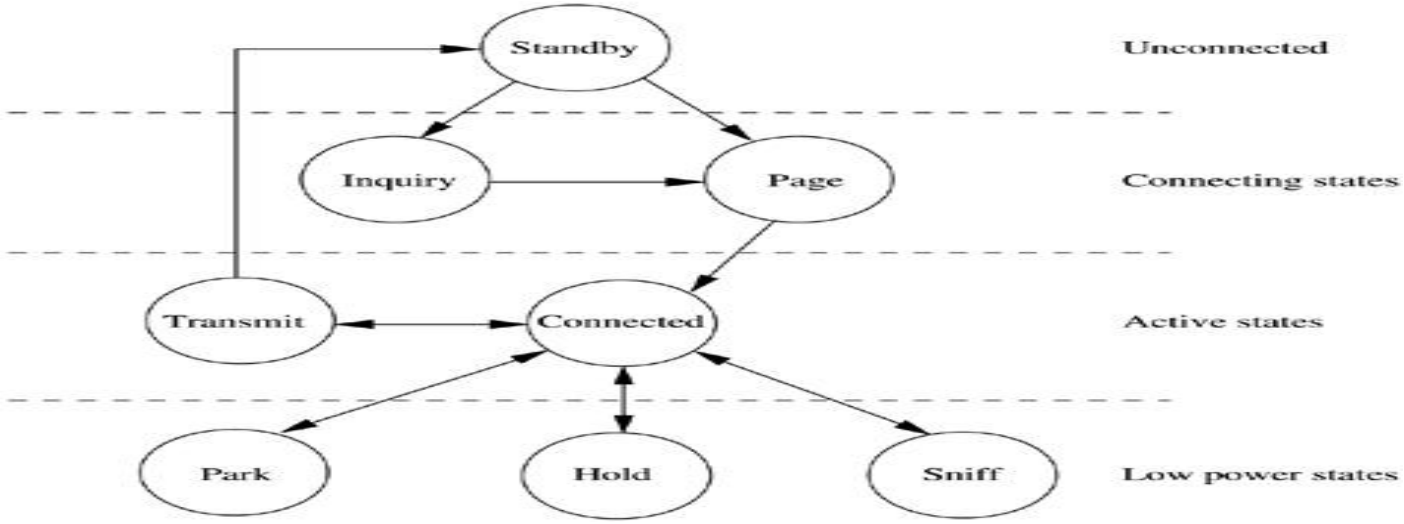
Figure 1.8. A typical piconet.



Operational States

Figure 1.9 shows the state diagram of Bluetooth communications. Initially, all the devices would be in the standby mode. Then some device (called the master)could begin the inquiry and get to know the nearby devices and, if needed, join them into its piconet. After the inquiry, the device could formally be joined by paging, which is a packet-exchange process between the master and a prospective slave to inform the slave of the master's clock. If the device was already inquired, the master could get into the page state bypassing the inquiry state. Once the device finishes getting paged, it enters the connected state. This state has three power-conserving sub-states – hold, sniff, and park (described later in this section). A device in the connected state can participate in the data transmission.

Figure 1.9. Operational states.



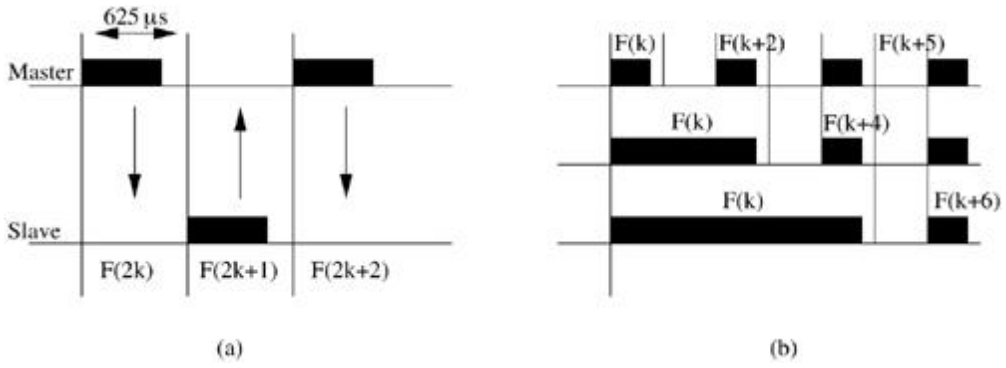
Frequency Hopping Sequences

It is evident (in any wireless communication) that the sender and the receiver should use the same frequency for communication to take place. A Frequency Selection Module (FSM) is present in each device to select the next frequency to be used under various circumstances. In the connected state, the clock and the address of the device (master) completely determine the hopping sequence. Different combination of inputs (clock, address) are used depending on the operational state. During the inquiry operation, the address input to FSM is a common inquiry address. This common address is needed because at the time of inquiry no device has information about the hopping sequence being followed. The address of the paged device is fed as input to the FSM for the paging state.

Communication Channel

The channel is divided into time slots, each 625 μs in length. The time slots are numbered according to the Bluetooth clock of the piconet master. A time division duplex (TDD) scheme is used where master and slave alternately transmit. The master starts its transmission in even-numbered time slots only, and the slave starts its transmission in odd-numbered time slots only. This is clearly illustrated in Figure 1.10 (a). The packet start shall be aligned with the slot start. A Bluetooth device would determine slot parity by looking at the least significant bit (LSB) in the bit representation of its clock. IfLSB is set to 1, it is the possible transmission slot for the slave. A slave in normal circumstances is allowed to transmit only if in the preceding slot it has received a packet from the master. A slave should know the master's clock and address to determine the next frequency (from the FSM). This information is exchanged during paging.

Figure 1.10. Transmission of packets over a channel.



Packet-Based Communication

Bluetooth uses packet-based communication where the data to be transmitted is fragmented into packets. Only a single packet can be transmitted in each slot. A typical packet used in these communications has three components: access code, header, and payload. The main component of the access code is the address of the piconet master. All packets exchanged on the channel are identified by the master's identity. The packet will be accepted by the recipient only if the access code matches the access code corresponding to the piconet master. This also helps in resolving conflicts in the case where two piconets are operating currently on the same frequency. A slave receiving two packets in the same slot can identify its packet by examining the access code. The packet header contains many fields such as a three-bit active slave address, a one-bit ACK/NACK for ARQ scheme [Automatic Repeat reQuest — anytime an error is detected, a negative acknowledgment (NACK) is returned and the specified frames are retransmitted], a four-bit packet type to distinguish payload types, and an eight-bit header error check code to detect errors in the header. Depending on the payload size, one, three, or five slots may be used for the packet transmission. The hop frequency which is used for the first slot is used for the remainder of the packet. While transmitting packets in multiple slots, it is important that the frequencies used in the following time slots are those that are assigned to those slots, and that they do not follow the frequency sequence that should have normally applied. This is illustrated in Figure 1.10 (b). When a device uses five slots for packet transmission, the next packet transmission is allowed in F(k+6) and not in F(k+2). Also note that the receiving time slot becomes F(k+5) as opposed to F(k+1). On this slotted channel, both

synchronous and asynchronous links are supported. Between a master and a slave there is a single asynchronous connectionless link(ACL) supported. This is the default link that would exist once a link is established between a master and a slave. Whenever a master would like to communicate, it would, and then the slave would respond. Optionally, a piconet may also support synchronous connection oriented (SCO) links. SCO link is symmetric between master and slave with reserved bandwidth and regular periodic exchange of data in the form of reserved slots. These links are essential and useful for high-priority and time-bound information such as audio and video.

Inquiry State

As shown in Figure 1.9, a device which is initially in the standby state enters the inquiry state. As its name suggests, the sole purpose of this state is to collect information about other Bluetooth devices in its vicinity. This information includes the Bluetooth address and the clock value, as these form the crux of the communication between the devices. This state is classified into three substates: inquiry, inquiry scan, and inquiry response. A potential master sends an inquiry packet in the inquiry state on the inquiry hop sequence of frequencies. This sequence is determined by feeding a common address as one of the inputs to the FSM. A device (slave) that wants to be discovered will periodically enter the inquiry scan state and listen for these inquiry packets. When an inquiry message is received in the inquiry scan state, a response packet called the Frequency Hopping Sequence (FHS) containing the responding device address must be sent. Devices respond after a random jitter to reduce the chances of collisions.

Page State

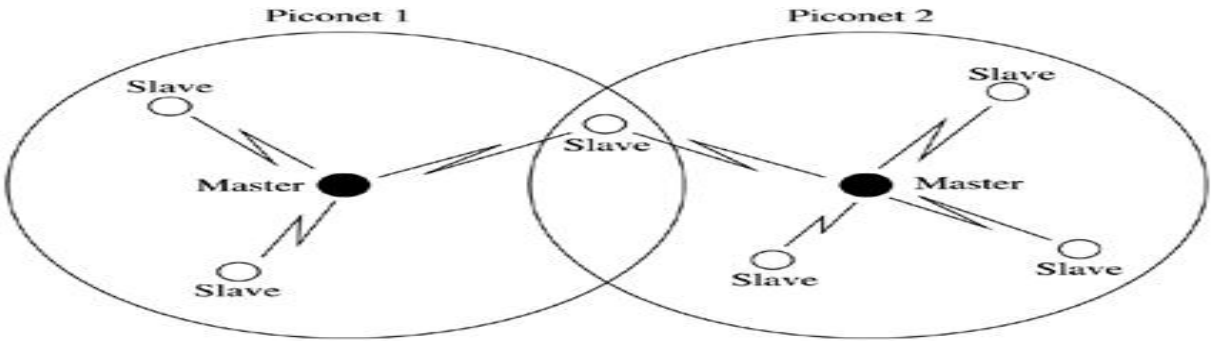
A device enters this state to invite other devices to join its piconet. A device could invite only the devices known to itself. So normally the inquiry operation would precede this state. This state also is classified into three sub-states: page, page scan, and page response. In the page mode, the master estimates the slave's clock based on the information received during the inquiry state, to determine where in the hop sequence the slave might be listening in the page scan mode. In order to account for inaccuracies in estimation, the master also transmits the

page message through frequencies immediately preceding and succeeding the estimated one. On receiving the page message, the slave enters the slave page response substate. It sends back a page response consisting of its ID packet which contains its Device Access Code (DAC). Finally, the master (after receiving the response from a slave) enters the page response state and informs the slave about its clock and address so that the slave can go ahead and participate in the piconet. The slave now calculates an offset to synchronize with the master clock, and uses that to determine the hopping sequence for communication in the piconet.

Scatternets and Issues

Piconets may overlap both spatially and temporally, that is, many piconets could operate in the same area at the same time. Each piconet is characterized by a unique master and hence the piconets hop independently, each with its own channel hopping sequence as determined by the respective master. In addition, the packets carried on the channels are preceded by different channel access codes as determined by the addresses of the master devices. As more piconets are added, the probability of collisions increases, and a degradation in performance results, as is common in FHSS systems. In this scenario, a device can participate in two or more overlaying piconets by the process of time sharing. To participate on the proper channel, it should use the associated master device address and proper clock offset. A Bluetooth unit can act as a slave in several piconets, but as a master in only a single piconet. A group of piconets in which connections exist between different piconets is called a *scatternet* (Figure 1.11).

Figure 1.11. A typical scatternet.



When a device changes its role and takes part in different piconets, it is bound to lead to a situation in which some slots remain unused (for synchronization).This implies that complete utilization of the available bandwidth is not achieved. An interesting proposition at this juncture would be to unite the timings of the whole of the scatternet. But this may lead to an increase in the probability of packets colliding. Another important issue is the timing that a device would be missing by participating in more than one piconet. A master that is missing from a piconet (by momentarily becoming a slave in another piconet) may miss polling slaves and must ensure that it does not miss beacons from its slaves. Similarly, a slave(by becoming a master or slave in another piconet) that is missing from a piconet could appear to its master to have gone out of range or to be connected through a poor-quality link.

Link Manager Protocol

Link manager protocol (LMP) is responsible for setting and maintaining the properties of the Bluetooth link. Currently, the major functionality of this layer is power management and security management. It also provides minimal QoS support by allowing control over parameters such as delay and delay jitter. Normally, a paging device is the default master of the piconet, but, depending on the usage scenario, the roles of the master and a slave could be switched and this is coordinated by exchange of LMP packets.

Power Management

The Bluetooth units can be in several modes of operation during the connection state, namely, active mode, sniff mode, hold mode, and park mode. These modes are now described.

• **Active mode:** In this mode, the Bluetooth unit actively participates in the piconet. Various optimizations are provided to save power. For instance, if the master informs the slave when it will be addressed, the slave may sleep until then. The active slaves are polled by the master for transmissions.

• **Sniff mode:** This is a low-power mode in which the listening activity of the slave is reduced. The LMP in the master issues a command to the slave to enter

the sniff mode, giving it a sniff interval, and the slave listens for transmissions only at these fixed intervals.

• **Hold mode:** In this mode, the slave temporarily does not support ACL packets on the channel (possible SCO links will still be supported). In this mode, capacity is made available for performing other functions such as scanning, paging, inquiring, or attending another piconet.

• **Park mode:** This is a very low-power mode. The slave gives up its active member address and is given an eight-bit parked member address. The slave, however, stays synchronized to the channel. Any messages to be sent to a parked member are sent over the broadcast channel characterized by an active member address of all zeros. Apart from saving power, the park mode helps the master to have more than seven slaves (limited by the three-bit active member address space) in the piconet.

Bluetooth Security

In Bluetooth communications, devices may be authenticated and links may be encrypted. The authentication of devices is carried out by means of a challenge response mechanism which is based on a commonly shared secret link key generated through a user-provided personal identification number (PIN). The authentication starts with the transmission of an LMP challenge packet and ends with the verification of result returned by the claimant. Optionally, the link between them could also be encrypted.

Logical Link Control and Adaptation Protocol (L2CAP)

This is the protocol with which most applications would interact unless a host controller is used. L2CAP supports protocol multiplexing to give the abstraction to each of the several applications running in the higher layers as if it alone is being run. Since the data packets defined by the baseband protocol are limited in size, L2CAP also segments large packets from higher layers such as RFCOMM or SDP into multiple smaller packets prior to their transmission over the channel. Similarly, multiple received baseband packets may be reassembled into

a single larger L2CAP packet. This protocol provides QoS on certain parameters such as peak bandwidth, latency, and delay variation when the link is established between two Bluetooth units.

Host Controller Interface

This is the optional interface layer, provided between the higher (above LMP) and lower layers of the Bluetooth protocol stack, for accessing the Bluetooth hardware capabilities. Whenever the higher layers are implemented on the motherboard of a host device, this layer is needed. Such an approach could prove beneficial as the spare capacity of the host device (say, a personal computer) could be utilized. The specification defines details such as the different packet types as seen by this layer. Command packets that are used by the host to control the device, event packets that are used by the device to inform the host of the changes, and data packets come under this category.

2.5.3 Middleware Protocol Group

The basic functionality of the middleware protocol group is to present to the application layers a standard interface that may be used for communicating across the transport layer, that is, the applications need not know the transport layer's complexities, they can just use the application programming interfaces(APIs) or higher level functions provided by the middleware protocols. This group consists of the RFCOMM layer, service discovery protocol (SDP), IrDA interoperability protocols, telephony control specification (TCS), and audio. The RFCOMM layer presents a virtual serial port to applications using the serial interface. Any application which is using the serial port can work seamlessly on Bluetooth devices. RFCOMM uses an L2CAP connection to establish a link between two devices. In the case of Bluetooth devices, there is no device which will be static and hence services offered by the other devices have to be discovered. This is achieved by using the service discovery protocol (SDP) of the Bluetooth protocol stack. Service discovery makes the device self configured without manual intervention. The IrDA interoperability protocol is not for communication between Bluetooth devices and Infrared devices. It is only for the existing IrDA applications to work on

Bluetooth devices without any changes. The main protocols in the IrDA set are IrOBEX (IrDA object exchange) for exchanging objects between two devices and IrMC (infrared mobile communications) for synchronization. Audio is the distinguishing part of Bluetooth. Audio is given the highest priority and is directly carried over the baseband at 64 Kbps so that a very good quality of voice is provided. Another important point to note here is that audio is actually not a layer of the protocol stack, but only a specific packet format that can be transmitted directly over the SCO links of the baseband layer. Telephony control is implemented using the telephony control specification –binary (TCS-BIN) protocol. TCS defines three major functional areas: call control, group management, and connectionless TCS. Call control is used to setup calls which can be subsequently used to carry voice and data traffic. TCS operates in both point-to-point and point-to-multipoint configurations. One of the main concepts of TCS is that of the wireless user group (WUG). Group management enables multiple telephone extensions, call forwarding, and group calls. For example, consider multiple handsets and a single base set. When a call comes in to the base set, all the multiple handsets can receive this call. In a similar fashion, calls can also be forwarded. The functionalities of TCS include *configuration distribution* and *fastintermember access.* Configuration distribution is the mechanism used to find the information about the other members in a group. Fast inter member access is a method for two slaves to create a new piconet. AWUG member uses the information from the configuration distribution and determines another member which it wants to contact. Then it sends the device's information to the master, which forwards it to this device. The contacted device then responds with its device address and clock information and places itself in a page scan state. Then the master contacts the device initiating the communication. This device now pages the contacted device and forms a new piconet. This explains how a new piconet is formed between two slaves with the help of the master. In all the above cases, a connection-oriented channel is established. To exchange simple information such as adjusting volume or signaling information, establishing such a channel is overkill and hence connectionless TCS has been provided for having a connectionless channel.

1.5.4 Bluetooth Profiles

These profiles have been developed to promote interoperability among the many implementations of the Bluetooth protocol stack. Each Bluetooth profile specification has been defined to provide a clear and transparent standard that can be used to implement a specific user end function. Two Bluetooth devices can achieve a common functionality only if both devices support identical profiles. For example, a cellular phone and a headset both have to support the Bluetooth headset profile for the headset to work with the phone. The Bluetooth profiles spring up from the usage models. In all, 13 profiles have been listed and these can be broadly classified into the following four categories:

1. **Generic profiles:** The Generic access profile, which is not really an application, provides a way to establish and maintain secure links between the master and the slaves. The service discovery profile enables users to access SDP to find out which applications (Bluetooth services)are supported by a specific device.

2. **Telephony profiles:** The cordless telephony profile is designed for three in-one phones. The Intercom profile supports two-way voice communication between two Bluetooth devices within range of each other. The Headset profile specifies how Bluetooth can provide a wireless connection to a headset (with earphones/microphones) for use with a computer or a mobile phone.

3. **Networking profiles:** The LAN Access profile enables Bluetooth devices to either connect to a LAN through APs or form a small wireless LAN among themselves. The dial-up networking profile is designed to provide dial-up connections via Bluetooth-enabled mobile phones. The FAX profile, very similar to the dial-up networking profile, enables computers to send and receive faxes via a Bluetooth-enabled mobile phone.

4. **Serial and object exchange profiles:** The serial port profile emulates a serial line (RS232 and USB serial ports) for (legacy) applications that require a serial line. The other profiles, generic object exchange, object push, file transfer, and synchronization, are for exchanging objects between two wireless devices. Bluetooth is the first wireless technology which has actually tried to attempt to make all the household consumer electronics devices follow one particular

communication paradigm. It has been partially successful, but it does have its limitations. Bluetooth communication currently does not provide support for routing. It should be noted that some research efforts are under way to accommodate this in the Bluetooth specification. Once the routing provision is given, inter-piconet communication could be enhanced. The issues of handoffs also have not yet been dealt with till now. Although master–slave architecture has aided low cost, the master becomes the bottleneck for the whole piconet in terms of performance, fault tolerance, and bandwidth utilization. Most importantly, Bluetooth communication takes place in the same frequency band as that of WLAN and hence robust coexistence solutions need to be developed to avoid interference. The technology is still under development. Currently, there are nearly 1,800 adopter companies which are contributing toward the development of the technology.

1.6 HOME RF

Wireless home networking represents the use of the radio frequency (RF)spectrum to transmit voice and data in confined areas such as homes and small offices. One of the visionary concepts that home networking intends to achieve is the establishment of communication between home appliances such as computers, TVs, telephones, refrigerators, and air conditioners. Wireless home networks have an edge over their wired counterparts because features such as flexibility (enabling of file and drive sharing) and interoperability that exist in the wired networks are coupled with those in the wireless domain, namely, simplicity of installation and mobility. The HIPERLAN/2, as mentioned earlier, has provisions for direct communication between the mobile terminals (the home environment). The home environment enables election of a central controller (CC) which coordinates the communication process. This environment is helpful in setting up home networks. Apart from this, an industry consortium known as the Home RF Working Group has developed a technology that is termed HomeRF. This technology intends to integrate devices used in homes into a single network and utilize RF links for communication. HomeRF is a strong competitor to Bluetooth as it operates in the ISM band.

Technical Features The HomeRF provides data rates of 1.6 Mbps, a little higher than the Bluetooth rate, supporting both infrastructure-based and ad hoc communications. It provides a guaranteed QoS delivery to voice-only devices and best-effort delivery for data-only devices. The devices need to be plug-and-play enabled; this needs automatic device discovery and identification in the network. Atypical HomeRF network consists of resource providers (through which communication to various resources such as the cable modem and phone lines is effected), and the devices connected to them (such as the cordless phone, printers, and file servers). The HomeRF technology follows a protocol called the shared wireless access protocol (SWAP). The protocol is used to set up a network that provides access to a public network telephone, the Internet (data), entertainment networks (cable television, digital audio, and video), transfer and sharing of data resources (such as disks and printers), and home control and automation. The SWAP has been derived from the IEEE 802.11 and the European digitally enhanced cordless telephony (DECT) standards. It employs a hybrid TDMA/CSMA scheme for channel access. While TDMA handles isochronous transmission (similar to synchronous transmission, isochronous transmission is also used for multimedia communication where both the schemes have stringent timing constraints, but isochronous transmission is not as rigid as synchronous transmission in which data streams are delivered only at specific intervals), CSMA supports asynchronous transmission (in a manner similar to that of the IEEE 802.11 standard), thereby making the actual framing structure more complex. The SWAP, however, differs from the IEEE 802.11specification by not having the RTS-CTS handshake since it is more economical to do away with the expensive handshake; moreover, the hidden terminal problem does not pose a serious threat in the case of small-scale networks such as the home networks. The SWAP can support up to 127 devices, each identified uniquely by a 48-bitnetwork identifier. The supported **devices** can fall into one (or more) of the following four basic types:

• Connection point that provides a gateway to the public switched telephone network (PSTN), hence supporting voice and data services.

• Asynchronous data node that uses the CSMA/CA mechanism to communicate with other nodes.

• Voice node that uses TDMA for communication.

• Voice and data node that can use both CSMA/CA and TDMA for channel access. Home networking also needs strong security measures to safeguard against potential eavesdroppers. That is the reason why SWAP uses strong algorithms such as Blowfish encryption. HomeRF also includes support for optional packet compression which provides a trade-off between bandwidth and power consumption. Because of its complex (hybrid) MAC and higher capability physical layer, the cost of HomeRF devices is higher than that of Bluetooth devices. HomeRF Version 2.0, released recently, offers higher data rates (up to 10 Mbps by using wider channels in the ISM band through FHSS).

Infrared The infrared technology (IrDA) uses the infrared region of the light for communication . Some of the characteristics of these communications are as follows:

• The infrared rays can be blocked by obstacles, such as walls and buildings.

• The effective range of infrared communications is about one meter. But when high power is used, it is possible to achieve better ranges.

• The power consumed by infrared devices is extremely low.

• Data rates of 4 Mbps are easily achievable using infrared communications.

• The cost of infrared devices is very low compared to that of Bluetooth devices.

Although the restriction of line of sight (LoS) is there on the infrared devices, they are extremely popular because they are cheap and consume less power. The infrared technology has been prevalent for a longer time than Bluetooth wireless communications. So it has more widespread usage than Bluetooth. Table1.2 compares the technical features of Bluetooth, HomeRF, and IrDAtechnologies.

Table 1.2. Illustrative comparison among Bluetooth, HomeRF, and IrDA technologies

Feature	Bluetooth	HomeRF	IrDA
Peak Data Rate	1 Mbps	1.6 Mbps	4 Mbps
Data Network Support	via PPP*	TCP/IP	TCP/IP
Voice Network Support	via SCO	via IP & PSTN	via IP
Range	< 10 meters	> 50 meters	< 10 meters
Power Consumption	0.25 - 100 mW	100 - 500 mW	10 mW (nominal)

*The point-to-point protocol is an Internet standard protocol for transporting IP datagrams over point-to-point links.

WIRELESS INTERNET

1.7 INTRODUCTION

The advent of the Internet has caused a revolutionary change in the use of computers and the search for information. The Internet has affected the traditional way of information exchange and now almost every city, every town, and every street has access to the Internet. Some basic concepts about the Internet and some fundamental issues that are encountered when a transition is made from the wired domain to the wireless domain and the MobileIP framework are discussed. Some of the problems faced during a transition from the wired domain to the wireless domain arise due to the fact that the protocols that work very well in the former may perform poorly in the latter. TCP is a perfect example of this. The key issues involved in TCP for wireless networks and an analysis of the current set of proposals to enhance the performance of TCP in the wireless domain are also presented. The classical wired networks have given rise to a number of application protocols such as TELNET, FTP, and SMTP. The wireless application protocol (WAP) architecture aims at bridging the gap at the application level, between the wireless users and the services offered to them.

1.8 WHAT IS WIRELESS INTERNET?

Wireless Internet refers to the extension of the services offered by the Internet to mobile users, enabling them to access information and data irrespective of their location. The inherent problems associated with wireless domain, mobility of nodes, and the design of existing protocols used in the Internet, require several

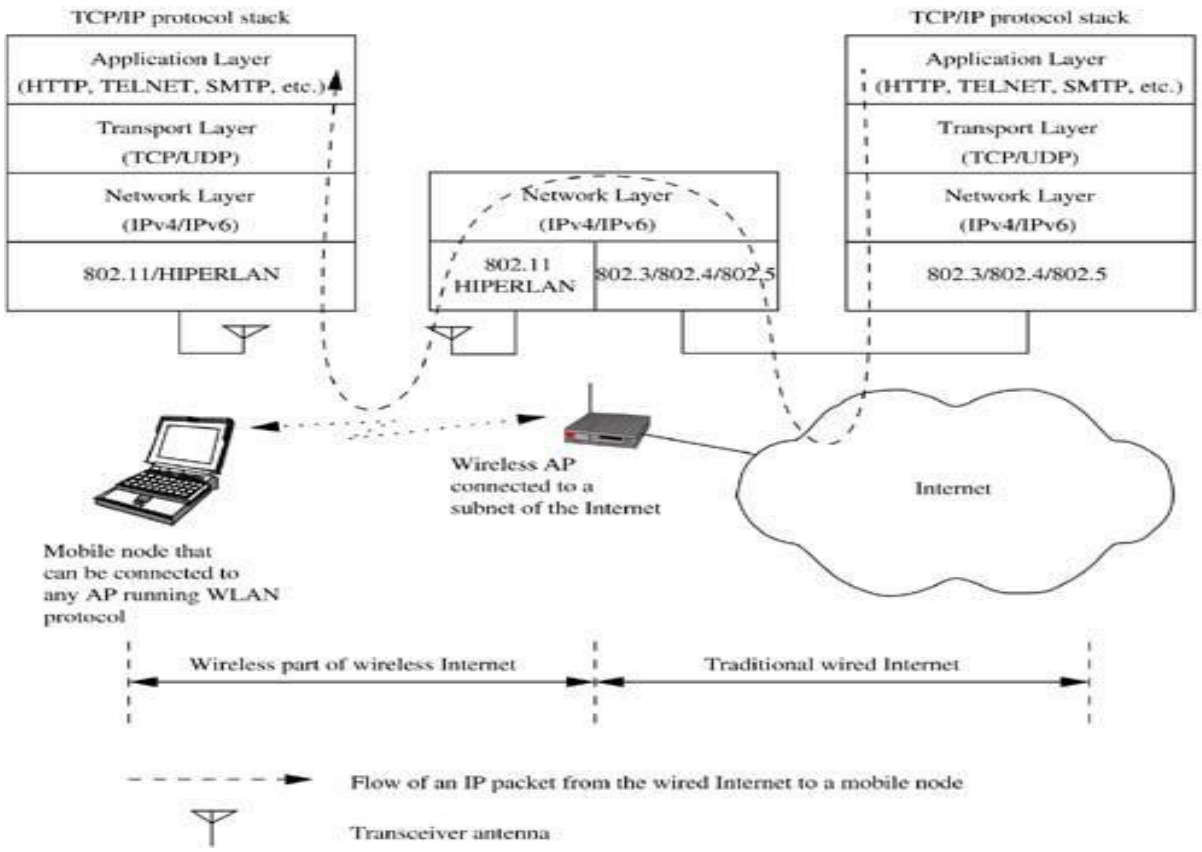
solutions for making the wireless Internet a reality. An illustration of wireless Internet with its layered protocol stack at wired and wireless parts is shown in Figure 1.12. The major issues that are to be considered for wireless Internet are the following.

• Address mobility

• Inefficiency of transport layer protocols

• Inefficiency of application layer protocols

Figure 1.12. An illustration of wireless Internet.

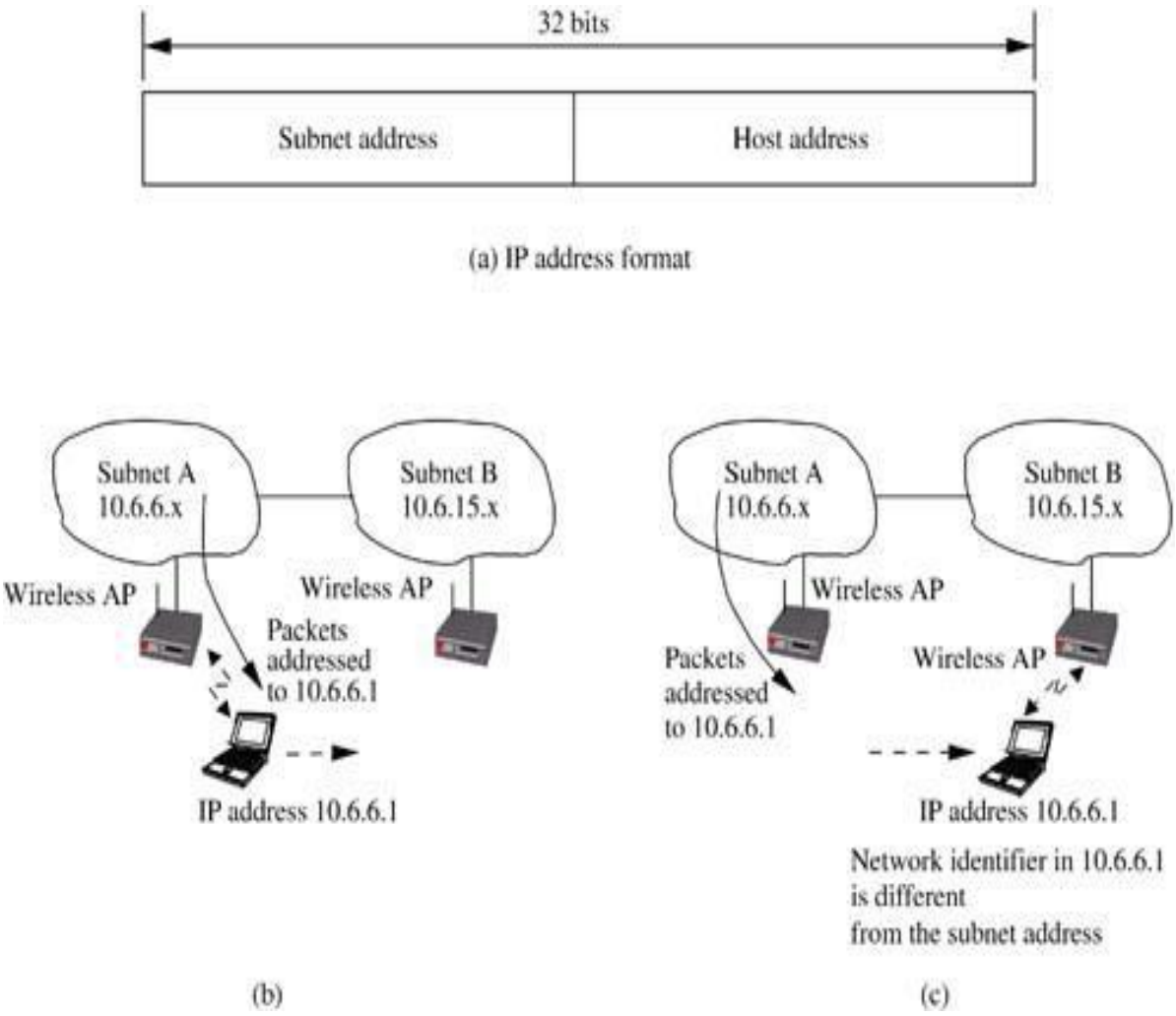


1.8.1 Address Mobility

The network layer protocol used in the Internet is Internet protocol (IP) which was designed for wired networks with fixed nodes. IP employs a hierarchical addressing with a globally unique 32-bit address which has two parts, network

identifier and host identifier, as shown in Figure 1.13 (a). The network identifier refers to the subnet address to which the host is connected. This addressing scheme was used to reduce the routing table size in the core routers of the Internet, which uses only the network part of the IP address for making routing decisions. This addressing scheme may not work directly in the wireless extension of the Internet, as the mobile hosts may move from one subnet to another, but the packets addressed to the mobile host may be delivered to the old subnet to which the node was originally attached, as illustrated in Figures 1.13 (b) and 1.13 (c). Hence the traditional IP addressing is not supportive of address mobility which is essential in wireless Internet. Figure 1.13 shows the mobility of a node (with IP address 10.6.6.1) attached to subnet A (subnet address 10.6.6.x) moving over to another subnet B with address 10.6.15.x. In this case, the packets addressed to the node will be routed to the subnet A instead of the subnet B, as the network part in the mobile node's address is10.6.6.x (see Figure 1.13 (c)). MobileIP$_2$ is a solution that uses an address redirection mechanism for this address mobility issue in wireless Internet.

Figure 1.13. The address mobility problem.



1.8.2 Inefficiency of Transport Layer Protocols

The transport layer is very important in the Internet as it ensures setting up and maintaining end-to-end connections, reliable end-to-end delivery of data packets, flow control, and congestion control. TCP is the predominant transport layer protocol for wired networks, even though UDP, a connectionless unreliable transport layer protocol, is used by certain applications. Wireless Internet requires efficient operation of the transport layer protocols as the wireless medium is inherently unreliable due to its time-varying and environment-dependent characteristics. Traditional TCP invokes a congestion control algorithm in order to handle congestion in the networks. If a data packet or an ACK packet is lost, then TCP assumes that the loss is due to congestion

and reduces the size of the congestion window by half. With every successive packet loss the congestion window is reduced, and hence TCP provides a degraded performance in wireless links. Even in situations where the packet loss is caused by link error or collision, the TCP invokes the congestion control algorithm leading to very low throughput. The identification of the real cause that led to the packet loss is important in improving the performance of the TCP over wireless links. Some of the solutions for the transport layer issues include indirect-TCP (ITCP), snoop TCP, and mobile TCP.

1.8.3 Inefficiency of Application Layer Protocols

Traditional application layer protocols used in the Internet such as HTTP, TELNET, simple mail transfer protocol (SMTP), and several markup languages such as HTML were designed and optimized for wired networks. Many of these protocols are not very efficient when used with wireless links. The major issues that prevent HTTP from being used in wireless Internet are its stateless operation, high overhead due to character encoding, redundant information carried in the HTTP requests, and opening of anew TCP connection with every transaction. Wireless bandwidth is limited and much more expensive compared to wired networks. Also, the capabilities of the handheld devices are limited, making it difficult to handle computationally and bandwidth-wise expensive application protocols. Wireless Application Protocol(WAP) and optimizations over traditional HTTP are some of the solutions for the application layer issues.

1.9 MOBILE IP

Each computer connected to the Internet has a unique IP address, which helps not only in identifying the computer on the network but also routing the data to the computer. The problem of locating a mobile host in a mobile domain is now imminent as the IP address assigned can no longer be restricted to a region. The first conceivable solution to the above problem would be to change the IP address when the host moves from one subnet to another. In this way, its address is consistent with the subnet it is currently in. The problems with changing the IP address as the host moves is that TCP identifies its connection with another terminal based on the IP address. Therefore, if the IP address itself

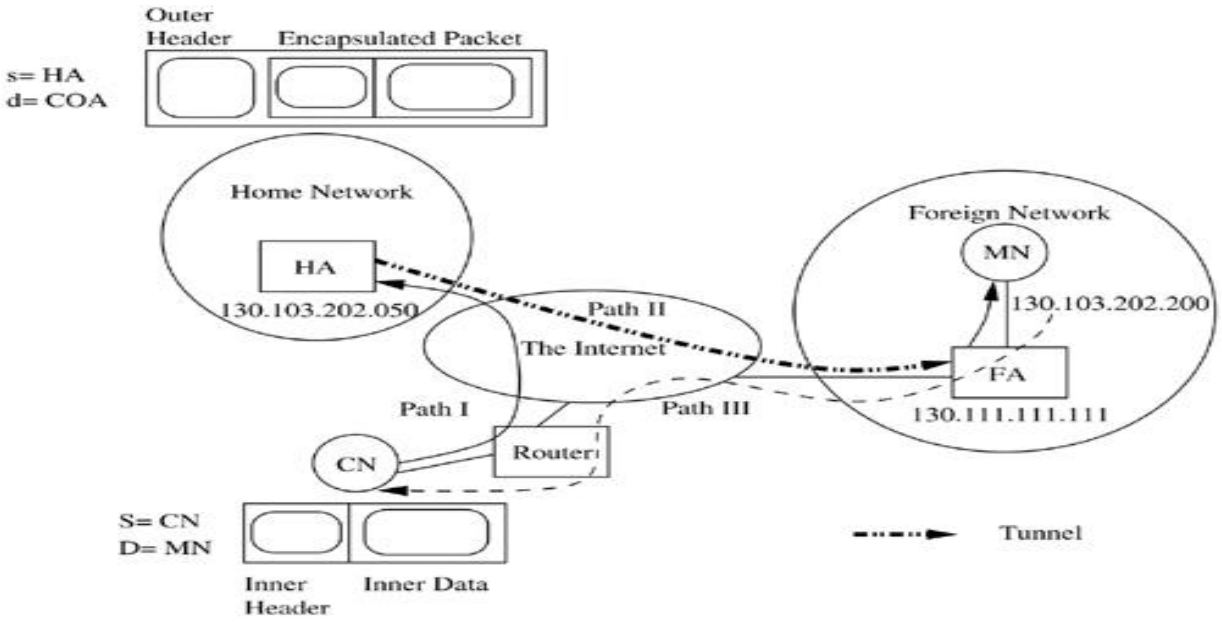
changes, the TCP connection must be reestablished. Another method would be to continue to use the same IP address and add special routing entries for tracking the current location of the user. This solution is practical if the number of mobile users is small. The quick-fix solutions are inadequate, but they give valuable insight into the nature of the mobility problem and offer certain guidelines for the actual solution. Before providing the solution to the problem, some issues of utmost importance need to be enumerated. These are as follows:

• **Compatibility:** The existing wired Internet infrastructure is well-established today and it is economically impractical to try to alter the way it is working.

• **Scalability:** Wireless communication is the technology for the future, so the solution should be scalable to support a large number of users.

• **Transparency:** The mobility provided should be transparent in the sense that the user should not feel a difference when working in a wireless domain or in a wired one. In Figure 1.14, mobile node (MN) is a mobile terminal system (end user) or a mobile router. It is the host for which the mobility support is to be provided. At the other end of the network is the system with which MN communicates. This is referred to as the correspondent node (CN), which may be a fixed or a mobile node. In this section, CN is considered to be a fixed, wired node. The node or router to which the MN is connected, which currently enjoys all the network facilities, is known as the foreign agent (FA). The subnet to which the MN's IP address belongs is the home network, and the router or node under whose domain this IP address lies is the home agent (HA).

Figure 1.14. Routing in MobileIP.



Suppose MN is currently in the subnet 130.111.*, hence as shown in the figure, 130.111.111.111 becomes the FA for MN. If CN sends a packet to MN, it reaches the HA of MN (130.103.202.050) along Path I.HA cannot find MN in the home network, but if it knows the location of MN, it can send the packet along Path II by creating a *tunnel*, as explained later.

1.9.1 MobileIP

The essence of the MobileIP scheme is the use of the old IP address but with a few additional mechanisms, to provide mobility support. MN is assigned another address, the care of address (COA).

The COA can be one of the following types:

1. **Foreign agent-based COA:** The address of the FA to which the MN is connected can be used to locate the MN. The COA of the MN in this case is the address of its current FA.

2. **Colocated COA:** In this case MN acquires a topologically correct IP address. In effect, each MN now has two IP addresses assigned to it. In this case the CN sends data to the old IP address. The HA receives this packet and *tunnels* it to the MN using the new IP address. In the case of FA-based COA, the FA decapsulates the packet and forwards it to MN, while in the case of colocated

COA, it is decapsulated at MN. The HA encapsulates the data packet inside another packet addressed to the COA of MN. This is known as *encapsulation* and the mechanism is known as *tunneling*. Path II in Figure 1.14 shows the tunnel using the FA-based COA. Though the problem is solved, it has been done with a high degree of inefficiency.

Registration with the HA

This section discusses how the COA of an MN is communicated to its HA. This is done through the process of *registration*. When an MN moves to a new location, it tries to find the FA. This is done using the agent advertisement packet or agent solicitation packet. Registration involves authentication and authorization of the MN by the HA. In case of the colocated COA, there is no intermediate FA. MN simply sends the registration request to its HA, which authenticates it and sends back a registration reply.

Reverse Tunneling

It appears that there should not be any problem for the MN in sending a packet to the CN following path III. However, there are other practical constraints that play an important role here.

1. **Ingress filtering:** There are some routers which filter the packets going out of the network if the source IP address associated with them is not the subnet's IP address. This is known as ingress filtering where the MN's packet may get filtered in the foreign network if it uses its home IP address directly.

2. **Firewalls:** As a security measure, most firewalls will filter and drop packets that originate from outside the local network, but appear to have a source address of a node that belongs to the local network. Hence if MN uses its home IP address and if these packets are sent to the home network, then they will be filtered.

3. **Time to live (TTL):** The MN should be able to communicate transparently with all the CNs that it can communicate with while at home. Hence, in case of *triangular routing*, the TTL for the packets must be reduced only by one, up to the point where the packet is tunnelled home. Firewalls and ingress filtering

have made a simple solution complicated. Therefore, to avoid these problems the idea of *reverse tunneling* is used, that is, MN encapsulates its packets using the source address of the encapsulated packet as its COA and destination as HA. The routing of packets from MN to CN takes place via the non-shortest path (as shown in Figure 1.14),that is, MN to HA to CN or vice versa is called *triangular routing*. This method, though not efficient, does work in practice.

1.9.2 Simultaneous Bindings Simultaneous bindings is a feature of MobileIP that allows an MN to register more than one COA at the same time, that is, the HA allows MN to register more than one COA. MN can also deregister a specific COA. In such a situation, the HA must send multiple duplicated encapsulated data packets, one to each COA. The idea behind the use of simultaneous binding is to improve the reliability of data transmission.

1.9.3 Route Optimization The packets sent to and from the HA are routed on non-optimal paths, hence the need for optimizations. The CN is assumed to be mobility-aware, that is, it has the capability to de-encapsulate the packets from the MN and send packets to the MN, bypassing the HA. The following are some of the concepts related to optimization strategies.

• **Binding cache:** The CN can keep the mapping of MN's IP address and COA in a cache. Such a cache is called a binding cache. Binding cache is used by the CN to find the COA of the MN in order to optimize the path length. Like any other cache, this may follow the update policies such as least recently used and first-in-first-out.

• **Binding request and binding update:** The CN can find the binding using a binding request message, to which the HA responds with a binding update message.

• **Binding warning:** In some cases, a handoff may occur, but CN may continue to use the old mapping. In such situations, the old FA sends a binding warning message to HA, which in turn informs the CN about the change, using a binding update message.

1.9.4 MobileIP Variations – The 4 × 4 ApproachAs discussed in Section 1.9.1, MobileIP is a general-purpose solution to the mobility problem over IPv4. It uses encapsulation as a primary technique and thus introduces a huge overhead (approximately 20 bytes per packet). In the MobileIP scheme, the MN is dependent on the FA to provide a COA. The presence of the FA in all transactions prevents the MN from being able to perform any kind of optimization, and it is unable to forgo the MobileIP support even when it is not required. The key factors that affect any optimization scheme are the permissiveness of the network and the capabilities of the communicating nodes. In the following strategy presented, it is presumed that the MN does not depend on the FA for any support and it is able to acquire a COA from the subnet that it is present in.

Goals of Optimizations

Any optimization scheme should try to ensure guaranteed delivery, low latency, and low overhead. Deliverability is to be understood in terms of the traditional datagram network that provides only a best-effort service. The latency issue mainly deals with the route that is being followed by the packet from the source to the destination, either in terms of the hop count or the delay. The overhead in the MobileIP scheme is essentially the packet encapsulation overhead.

The 4 × 4 Approach

The strategy presented here provides four options for packets directed from the MN to the CN (OUT approaches) and four more options for packets directed from the CN to the MN (IN approaches). The set of options can be provided as a4×4 matrix to the hosts, which can decide on the appropriate combination depending on the situation. The IN and OUT strategies are summarized in Tables 1.4 and 1.5, respectively. s and d represent the outer source and destination in the encapsulated packet while S and D represent the inner source and destination of the packet (refer to Figure 1.14). Indirect transmission refers to the routing of packets between the CN and MN involving the HA, whereas direct transmission bypasses the HA. In Table 1.4 the four IN strategies are listed along with the respective source and destination fields, and the

assumptions made and restrictions on usage of the strategies. For example, INIE uses the traditional MobileIP mechanism and works in all network environments irrespective of security considerations, while IN-DT is applicable for short-term communication wherein the mobility support is compromised. In Table 1.5, the four OUT strategies are listed. For example, OUT-IE uses the traditional MobileIP reverse tunneling mechanism and works in all networks cenarios, while OUT-DH avoids encapsulation overhead but can be used only when the MN and CN are in the same IP subnet.

Table 1.4. The IN strategies in 4 × 4 approachTable

IN Strategy	s	d	S	D	Notes	Acceptable Combinations
Incoming Indirect Encapsulated (IN-IE)	IP address of HA	COA of the MN	IP address of the CN	Home IP address of the MN	1. Highest overhead 2. Guaranteed delivery 3. Uses tunneling 4. CN need not be mobility aware	OUT-IE OUT-DE OUT-DH
Incoming Direct Encapsulated (IN-DE)	IP address of CN	COA of the MN	IP address of the CN	Home IP address of the MN	1. CN is mobility aware 2. No tunneling	OUT-DE OUT-DH
Incoming Uses Home Address (IN-DH)	Not applicable	Not applicable	IP address of the CN	Home IP address of the MN	1. No encapsulation 2. Usable when there are no security constraints at intervening routers 3. MN and CN on same subnet	OUT-DH only
Incoming Direct Uses Temporary Address (IN-DT)	Not applicable	Not applicable	IP address of the CN	COA of MN	1. MN cannot receive packets addressed to its original IP address 2. Useful for short-term communication	OUT-DT only

1.5. The OUT strategies in 4×4 approach

OUT Strategy	s	d	S	D	Notes	Acceptable Combinations
Outgoing Indirect Encapsulated (OUT-IE)	COA of the MN	IP address of HA	Home IP address of the MN	IP address of the CN	1.Highest overhead 2. Guaranteed delivery 3. No ingress filtering 4. CN need not be mobility aware	IN-IE only
Outgoing Direct Encapsulated (OUT-DE)	COA of the MN	IP address of CN	Home IP address of the MN	IP address of the CN	1. CN is mobility aware 2. No tunneling	IN-IE IN-DE
Outgoing Direct Home Address (OUT-DH)	Not applicable	Not applicable	Home IP address of the MN	IP address of the CN	1. No encapsulation 2. Usable when there are no security constraints at the intervening routers 3. MN and CN on the same subnet	IN-IE IN-DE IN-DH
Outgoing Direct uses Temporary Address (OUT-DT)	Not applicable	Not applicable	COA of MN	IP address of the CN	1. MN cannot receive packets addressed to its original IP address 2. Useful for short-term communication	IN-DT only

Comparison and Evaluation of the Strategies

Having seen the four approaches for each of the two directions of packet transfer, different combinations of these strategies can be considered. Though there seem to be 16 combinations, some of them are inapplicable and some are redundant. There are also restrictions on when the approaches are valid; the characteristics of the situation will determine which approach to choose. The choice of a particular strategy can be made on a per session basis or on a packet to-packet basis, as desired by the entities involved in the conversation. Tables 1.4 and 1.5 also show the acceptable combinations of the strategies.

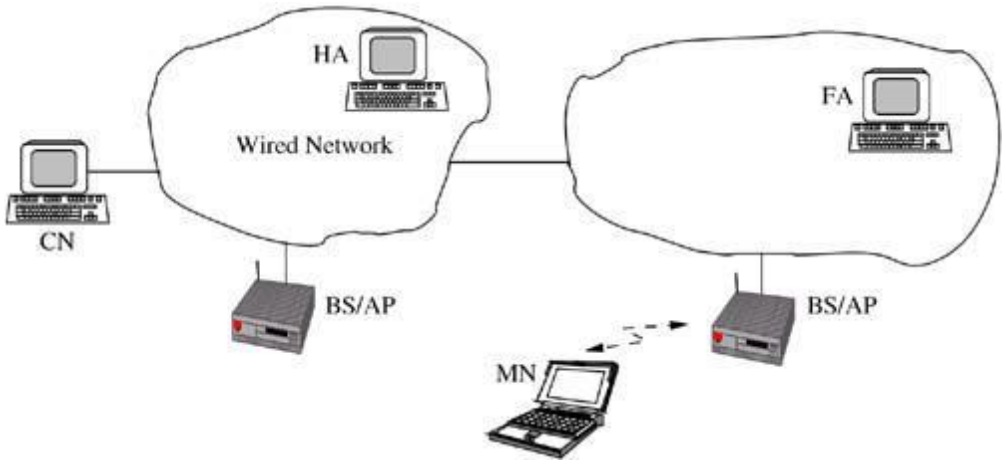
1.9.5 Handoffs

A handoff is required when the MN is moving away from the FA it is connected to, and as a result the signals transmitted to and from the current FA become weak. If the MN can receive clearer signals from another FA, it breaks its connection with the current FA and establishes a connection with the new one. The typical phases involved in handoff are measuring the signal strength, decisions regarding where and when to hand off, and the establishment of a new connection breaking the old one.

Classification of Handoffs

The issues in handoffs are on the same lines as those in cellular networks. Handoffs can be classified in three ways based on functionalities of the entities involved, signaling procedure, and number of active connections. Function-based classification is based on the roles of the MN and FA during the handoff. Figure 1.15 shows the MN, BS, FA, and CN in the handoff scenario.

Figure 1.15. Entities in wireless Internet handoff scenario.



Here, handoffs can be classified into four categories as follows:

1. **Mobile initiated handoff:** In this case, the handoff is managed by the MN. The MN measures the signal strength, decides the target base station (BS), and triggers the handoff.

2. **Mobile evaluated handoff:** This is similar to the previous case except that the decision on the handoff lies within the network, perhaps with the BS.

3. **Network initiated handoff:** In this case, the network (BS) decides where the MN should be handed over. Also, only the network measures the signal strength of the uplink and the MN has very little role to play.

4. **Mobile assisted handoff:** The MN assists the network in the network initiated scenario by measuring the downlink signal strength. This is typically to avoid a *black hole* scenario. A black hole scenario occurs when the channel properties tend to be asymmetric. (Usually wireless channels are assumed to have the same properties in both uplink and downlink, but in certain circumstances the throughput on one of the directions may be significantly less than the other. This scenario is referred to as a black hole.)The second kind of classification is based on the number of active connections, where the handoffs are classified into two types: the hard handoff (only one active connection to the new or the old FA) and the soft handoff (has two active connections during the handoff).Signaling procedure-based handoffs are classified into two types depending on which FA (old FA or new FA) triggers the handoff along with MN.

• **Forward handoff:** In this case, MN decides the target BS and then requests the target BS to contact the current BS to initiate the handoff procedure.

• **Backward handoff:** In this case, MN decides the target BS and then requests the current BS to contact the new one.

Fast Handoffs

A typical handoff takes a few seconds to break the old connection and establish the new one. This delay may be split into three components : delay in detection of a need for a handoff, layer2 handoff (a data link connection that needs to be established between the new FA and MN), and layer3 handoff or registration with HA. The first two components cannot be avoided; however, the delay due to the third can be reduced. Also, if the above operations are parallelized, the total delay will be reduced. Two techniques called pre- and post-registration handoffs are employed to perform the above operations. The difference lies in the order in which the operations are performed. In the case of the pre-registration handoff, the registration with the HA takes place before the handoff while the MN is still attached to the old FA, while in the case of the post-registration handoff, registration takes place after the MN is connected to the new FA. In this case, the MN continues to use the old FA, tunnelling data via the new FA until the process of registration is completed.

1.9.6 IPv6 Advancements

The various optimizations provided over IPv4 (IP version 4) in order to avoid the inefficiencies in routing MN's data were discussed in Section 1.9.4. IPv6(IP version 6) has a built-in support for mobility to a great extent. The features are listed below:

• Route optimization is a built-in feature of IPv6.

• IPv6 has fields for specifying both new (COA) and home (IP) address. So problems that lead to reverse tunneling can be avoided.

• The problem of *ingress filtering* is also solved due to the above.

• Control packets such as those used in route optimization can be piggy-backed onto the data packets.

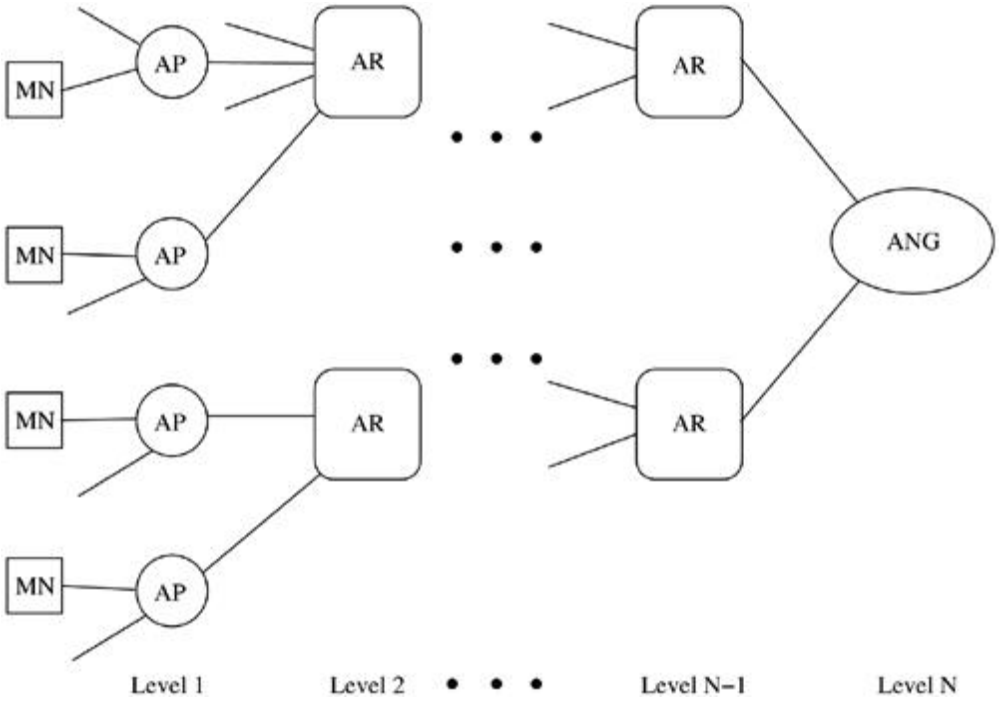
• *Detection of black holes:* Sometimes it might happen that the signals of one of the links (uplink or downlink) become weak while the other link has a good signal strength. Such a phenomenon is known as a *black hole* because data can go in one direction but cannot come out in the other. In such cases, a handoff may be required. IPv6 allows both MN and BS to detect the need for a handoff due to creation of black holes.

• IPv6 avoids overheads due to encapsulation because both the COA and the original IP address are included in the same packet in two different fields. Apart from these, IPv6 allows 2^{128} addresses, thereby solving the IP address shortage problem, and includes advanced QoS features. It also supports encryption and decryption options to provide authentication and integrity.

1.9.7 IP for Wireless Domains

MobileIP is only a solution to the mobility of IP address problem, it is not a specific solution for wireless, especially cellular domains. The following discussion addresses certain protocols that are IP-based and suited for the wireless domain as well. In particular, we consider an approach which is terminal independent, that is, an approach aimed at giving a uniform service to both hosts that have the MobileIP capability as well as legacy hosts. The terminal independent mobility for IP (TIMIP) strategy is based on two main protocols for the wireless networks, namely, HAWAII and CellularIP. Figure 1.16 gives the hierarchy of routers in the HAWAII, CellularIP, and TIMIP architectures. The access point (AP) is a router that is at the first level of the hierarchy and this is in direct communication with the MN over the wireless interface. Access Routers (AR) are interior routers in the tree. The Access Network Gateway (ANG) is the router at the root of the tree that acts as the interface between the wireless (TIMIP) domain and the core wired IP network.

Figure 1.16. Hierarchical routers.



HAWAII

HAWAII stands for Handoff Aware Wireless Access Internet Infrastructure. The infrastructure identifies two categories of mobility to be handled, micro mobility(intra-domain) and macro mobility (inter-domain), where domain refers to a part of the network under the control of a single authority, such as AR and ANG. The objective of the infrastructure is to solve the QoS and efficiency issues that are not addressed by MobileIP.

CellularIP

CellularIP offers an alternative to the handoff detection problem by using the MAC layer information based on the received signal strengths to detect handoffs, instead of using the network layer information. The routing nodes maintain both a paging cache and a routing cache; the routing cache is a mapping between an MN's IP address and its current location in the Cellular IPdomain. The paging cache is preferred for nodes that receive or send packets relatively infrequently, and it is maintained by paging update packets sent by the MN whenever it crosses between two APs. The routing cache will be updated whenever the MN has a packet to send. The MN will send the packet to the closest AP and this will update all routing caches all the way up to the ANG. It

is to be noted that during a handoff or just after the handoff, packets meant for the MN will be routed to both the old as well as the current AP in charge of the MN for a time interval equal to the routing cache timeout.

TIMIP

In the Terminal Independent Mobility for IP (TIMIP) approach, the emphasis is on providing a uniform service to both MobileIP-capable MNs as well as the legacy terminals. The MobileIP capability of the legacy terminals will be provided by the ANG. The ANG will keep track of the information regarding each of the MNs in its domain such as the MN's MAC and IP addresses, the MobileIP capabilities, and the authentication parameters. Whenever an MN arrives in the TIMIP domain, a routing path has to be created in the domain so that all packets intended for this host can be efficiently routed. This will cause a trigger of updates to ensure route reconfiguration in the entire hierarchy. The ARs not involved in the route will be unaware of the new path to the MN. As a result, the default strategy for any packet in the TIMIP domain, that is, for any IP address that is unknown at a particular AP or AR, will be to route it to the ANG.

• **Micromobility:** Whenever the MN moves within the same TIMIP domain, it is referred to as micromobility. The route updates and the corresponding acknowledgments will propagate up the hierarchy until the crossover AR is reached. The old path needs to be deleted in all the routing tables of the nodes. Now the crossover AR will send a route update packet addressed to the MN, and this packet will propagate down the tree until the old AP in charge of the MN is reached.

• **Macromobility:** Similar to CellularIP and HAWAII, TIMIP relies purely on MobileIP to support macromobility. The ANG acts as the MobileIP proxy on behalf of the MN that does not have MobileIP capability, and does all the MobileIP signaling that the MN would have normally done. For the normal MobileIP capable MNs, however, the ANG performs the role of a FA. The TIMIP approach also provides for seamless mobility through the context

transfer framework. The context transfer essentially ensures that the data loss during handoff is minimized and this is transparent to the MN and the CN.

1.9.8 Security in MobileIP

The wireless domain is inherently insecure. Any data that needs to be transmitted has to be broadcast and anyone who can hear this can read it irrespective of the destination address.

Security Problems

The common security problems that may arise in wireless networks are as follows:

• **Registration request by a malicious node:** This is a problem because amalicious node can pose as a legitimate MN and use the MN's IP address forregistration, thereby enjoying all the facilities meant for the MN.

• **Replay attacks:** Many times the above problem is solved by making the registration process encrypted. Though this appears to avoid the first problem, the malicious node may copy the MN's registration packet, which is encrypted when the MN tries to register with the FA. Though this packet cannot be decoded by this malicious node, it can certainly use this packet for registering itself as the MN at a later point of time, and hence enjoy all the facilities at the cost of the MN.

• **Tunnel hijacking:** In this case, the malicious node uses the tunnel built by the MN to break through the firewalls.

• **FA can itself be a malicious node.** The MN and HA share the same security association and use the *message digest5* (MD5) with 128-bit encryption. To circumvent the problem of replay attacks the MN and HA use a shared random number[4] (called Nonce) and this random number is sent along with the encrypted registration request. On registration, the HA verifies the random number and issues a new random number to be used for the next registration. Hence, even if the packet is copied by the malicious node, it becomes useless for a replay

attack, as at the time of the next registration the random number would have changed anyway.

1.9.9 MRSVP - Resource Reservation

The following section describes a reservation protocol used to provide real-time services to mobile users. A major problem is that mobility affects the QoS adversely. Hence there is a need for advance reservations to be made on behalf of a mobile host at future locations that it is likely to visit. We notice that the current RSVP[5] structure is far from adequate and examine the proposed scheme.

Overview

The usual QoS parameters are delay, loss, throughput, and delay jitter. Whenever an MN moves across from one agent to another, there is obviously a change in the data flow path due to the handoff. The delay is likely to change due to the change in the data flow path and also due to the fact that the new location may vary from the old location with respect to congestion characteristics. Again, if the new location is highly congested, the available bandwidth is less, hence the throughput guarantees that were provided earlier may be violated. In addition, under extreme cases there may be temporary disconnections immediately following a handoff, which causes significant data loss during the transit.

Requirements of a Mobility-Aware RSVP

A fundamental requirement is that an MN must be able to make advance reservations along data flow paths to and from locations that it is likely to visit in the lifetime of a particular connection or session. Such a protocol has to have information that we refer to as the MSPEC, which is the set of locations from which the MN requires reservations. The definition of the MSPEC may be either statically done or there may be additional options to update it dynamically while the flow is active. A hypothetical MRSVP has two types of reservations: ACTIVE and PASSIVE. An ACTIVE reservation is a normal RSVP-like reservation that is on the data flow path from the current location of the MN. A PASSIVE reservation is made along all paths to and from other locations in the MSPEC of the MN. The path along which the reservation will be made is the

path specified by the MobileIP protocol. Passive reservations become active reservations whenever there is a active sender or receiver involved in that data flow path. The paths along which passive reservations have been made can be used by other flows with weaker QoS guarantees, but appropriate action needs to be taken when the passive flow turns into an active one.

MRSVP – Implementation

We have to identify proxy agents (PAs) that will make reservations on behalf of mobile senders and receivers. There are two types of PAs: remote and local. A local proxy agent (LPA) is that to which the MN is currently attached. Every other agent in the MSPEC will be a remote proxy agent (RPA).The sender periodically generates ACTIVE PATH messages, and for a mobile sender the PAs will send PASSIVE PATH messages along the flow path to the destination. Similarly, the PAs for a mobile receiver send the PASSIVE RESV messages while the receiver itself sends the ACTIVE RESV message. The framework also defines additional messages such as JoinGroup, RecvSpec, SenderSpec, and SenderMSpec . The key issues in the implementation are as follows:

• The identification of proxy agents (local and remote) that will perform the reservations on behalf of an MN.

• The identification of flow anchors (proxy agents), a Sender Anchor when the MN is a sender and a Receiver Anchor when the MN is a receiver, that will act as fixed points in the flow path.

• The establishment of both active and passive reservations (by the remote proxy agents) for the MN according to the MSPEC.

• The actual message sequences that lead to the reservation depends on the type of the flow and the strategy adopted. The MRSVP scheme is an initial approach to providing QoS guarantees within the MobileIP framework. The scheme considers both unicast as well as multicast traffic for all types of senders and receivers. The significant contribution of the approach is the notion of PASSIVE reservations that exist virtually on future routers that the MN's data

flow is likely to use, but will turn into real flows when theMN moves into the new domain.

1.10 TCP IN WIRELESS DOMAIN

The topics discussed so far addressed the network layer modifications that are necessary to make an efficient transition from the wired to the wireless domain. The wireless domain is not only plagued by the mobility problem, but also by high error rates and low bandwidth. Obviously there needs to be a higher layer abstraction that would perform the error recovery and flow control. The traditional TCP, which guarantees in-order and reliable delivery, is the classical wired networks transmission protocol. Since the transition to the wireless domain should be compatible with the existing infrastructure, there is need for modifications of the existing protocols. This is the correct approach rather than resorting to a completely new set of protocols.

1.10.1 TraditionalTCP

TCP provides a connection-oriented, reliable, and byte stream service. The term connection-oriented means the two applications using TCP must establish a TCP connection with each other before they can exchange data. It is a full duplex protocol, meaning that each TCP connection supports a pair of byte streams, one flowing in each direction. TCP includes a flow-control mechanism for each of these byte streams that allows the receiver to limit how much data the sender can transmit. TCP also implements a congestion-control mechanism.TCP divides the data stream to be sent into smaller segments and assigns sequence numbers to them. The sequence number helps the receiver to provide the higher layers with in-order packet delivery, and also detect losses. The sliding window mechanism employed by TCP guarantees the reliable delivery of data, ensures that the data is delivered in order, and enforces flow control between the sender and the receiver. In the sliding-window process, the sender sends several packets before awaiting acknowledgment of any of them, and the receiver acknowledges several packets at a time by sending to the transmitter the relative byte position of the last byte of the message that it has received successfully. The number of packets to be sent before the wait for

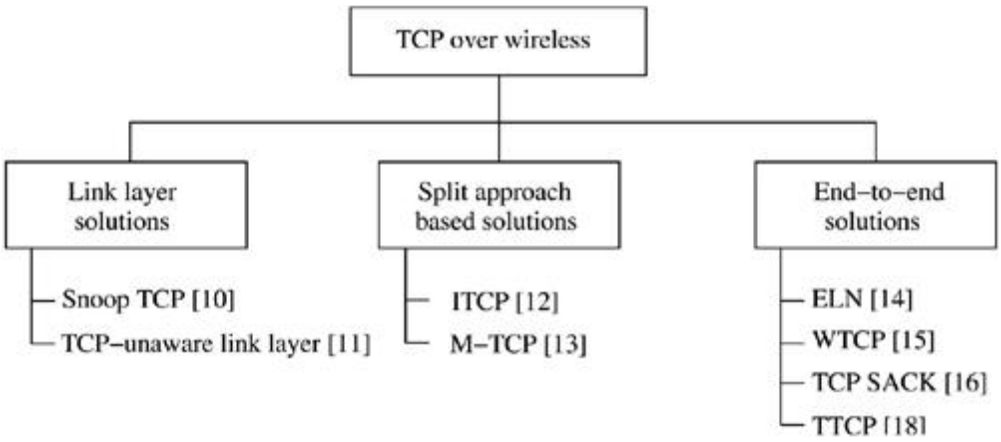
acknowledgment (window size) is set dynamically, that is, it can change from time to time depending on network conditions. Because the major cause of packet loss in the wired domain is congestion, TCP assumes that any loss is due to congestion. The TCP congestion control mechanism works as below. Initially, the TCP sender sets the congestion window to the size of one maximum TCP segment [also known as maximum segment size (MSS)]. The congestion window gets doubled for each successful transmission of the current window. This process continues until the size of the congestion window exceeds he size of the receiver window or the TCP sender notices a timeout for any TCP segment. The TCP sender interprets the timeout event as network congestion, initializes a parameter called *slow start threshold* to half the current congestion window size, and resets the congestion window size to one MSS. It then continues to double the congestion window on every successful transmission and repeats the process until the congestion window size reaches the slow start window threshold. Once the threshold is reached, the TCP sender increases the congestion window size by one MSS for each successful transmission of the window. This mechanism whereby the congestion window size is brought down to one MSS each time network congestion is detected and then is incremented as described above is referred to as *slow start.* Another important characteristic of TCP is fast retransmit and recovery. If the receiver receives packets out of order, it continues to send the acknowledgment for the last packet received in sequence. This indicates to the sender that some intermediate packet was lost and the sender need not invoke the congestion control mechanism. The sender then reduces the window size by half and retransmits the missing packet. This avoids the slow start phase.

1.10.2 TCP Over Wireless

The adaptation of TCP to congestion causes a lot of problems in the wireless domain. The wireless domain has high packet loss and variable latency, which may cause TCP to respond with slow start. Bandwidth utilization is further reduced due to retransmission of lost packets. One of the earliest suggested alternatives for improving the performance of TCP over wireless networks was to ensure that the link layer corrected all the errors itself over the wireless

interface, thereby eliminating the need for error handling at the TCP layer. One of the suggestions in this category is the use of forward error correction (FEC) to correct small errors. FEC is a means of error control coding wherein redundancy is encoded into the sent message or binary stream to allow self-correction at the receiver. The main objective of these techniques is to hide errors from TCP as far as possible. However, FEC incurs overhead even when there are no errors as there must be the redundant parity bits to allow error detection and correction. The alternative is to use adaptive schemes, which are dynamic in the sense that when the error rate or error probability is found to be higher than usual, the redundancy introduced into the transmitted stream is also correspondingly increased. Under normal circumstances, the overhead is kept to a minimum. The other form of link layer recovery is to use retransmissions at the link layer. This incurs the overhead only on error. However, the link level recovery mechanism may cause head-of the-line blocking, wherein the recovery mechanisms employed for one data stream consume the network resources and prevent others from being able to transmit packets. Some researchers have advocated the use of the retransmit when FEC capability is exceeded. The most accepted role of the link layer strategy would be one in which the link layer helps TCP error recovery by providing "almost in order delivery" of packets. Not all connections can benefit from link level retransmission as it is dependent on the nature of the applications. Several alternatives have been proposed to alter the existing TCP protocol to suit the wireless domain. The simplest idea would be to design a new TCP protocol for the wireless domain, but this will be incompatible with the wired domain. The following sections discuss various approaches to improve TCP performance in the wireless domain. Figure 1.17 provides a classification of the existing approaches.

Figure 1.17. Classification of approaches for TCP over wireless.



1.10.3 Snoop TCP

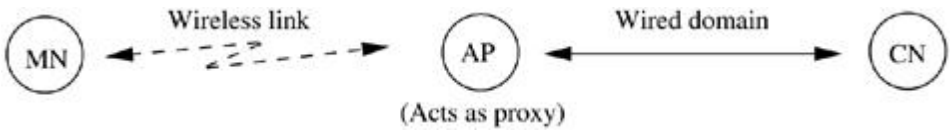
The central idea used in snoop TCP is to buffer the data as close to MN as possible in order to minimize the time for retransmission. The BS just snoops the packets being transmitted in both directions and recognizes the acknowledgments. The BS buffers the packets transmitted but does not acknowledge on behalf of MN. It simply removes the packet from the buffer when it sees an acknowledgment. If BS gets a duplicate acknowledgment (DUPACK) or no acknowledgment for quite some time, then it retransmits from the buffer after discarding the duplicate acknowledgment. This is to avoid unnecessary retransmissions from CN. The BS does not send acknowledgments to the CN on behalf of the MN, in order to retain the end-to-end semantics that traditional TCP provides. When the data transmission is from MN to CN, if the BS detects a gap in the sequence numbers acknowledged by the CN, it sends a NACK or negative acknowledgment to the MN to indicate loss over the wireless link.

1.10.4 TCP-Unaware Link Layer This strategy particularly aims at simulating the behavior of the snoop- TCP protocol without requiring the link layer at the BS to be TCP-aware (hence the name TCP-unaware link layer even though TCP requires some information from the link layer). The usage of delayed DUPACKs imitates snoop- TCP without requiring the link layer at BS to be TCP-aware. At the BS, as in snoop-TCP, link layer retransmission is used to perform local error recovery. But unlike snoop-TCP, where retransmissions are

triggered by TCP DUPACKs, here retransmissions are triggered by link level ACKs. The MN reduces the interaction between the link layer and TCP using delayed DUPACKs. The advantages of this scheme are that the link layer need not be TCP-aware, it can be used even if headers are encrypted, which is not possible in snoop-TCP, which needs to look into the headers to see the sequence numbers, and it works well for small round trip times (RTTs) over the wireless link. The most significant disadvantage of this mechanism is that the optimum value of DUPACK delay is dependent on the wireless link, and this value is crucial in determining the performance.

1.10.5 Indirect TCP This approach involves splitting of the TCP connection into two distinct connections, one TCP connection between the MN and BS[6] and another TCP connection between the BS and the CN. [6] In this context of TCP over wireless, the terms BS and AP are used interchangeably. Such a division splits the TCP connection based on the domain, the wireless domain, and the wired domain. The traditional TCP can be used in the wired part of the connection and some optimized version of TCP can be used in the wireless counterpart. In this case, the intermediate agent commonly known as the access point (AP) acts as a proxy for MN. The indirect TCP (ITCP) mechanism is shown in Figure 1.18. Loss of packets in the wireless domain, which would otherwise cause a retransmission in the wired domain, is now avoided by using a customized transport protocol between the AP and MN which accounts for the vagaries of the wireless medium. The AP acknowledges CN for the data sent to MN and buffers this data until it is successfully transmitted to MN. MN acknowledges the AP alone for the data received. Handoff may take a longer time as all the data acknowledged by AP and not transmitted to MN must be buffered at the new AP.

Figure 1.18. Indirect TCP.



4.4.6 MobileTCP

The most common problem associated with the wireless domain is that quite often the connection between MN and BS is lost for small intervals of time. This typically happens when MN moves behind a huge building or MN enters offices where the signals are filtered. In such cases, the sender will keep transmitting and times out eventually. In case of ITCP, the data buffered at AP may grow too large in size. It may also lead to slow start. In such situations the sender needs to be informed. This situation is handled in mobile TCP (M-TCP) by the supervisory host (the node in the wired network that controls a number of APs) which advertises the window size to be one, thus choking the sender and hence avoiding slow start. Connection may be resumed when MN can be contacted again. When the supervisory host receives a TCP packet, it forwards it to the M-TCP client. Upon reception of an ACK from M-TCP client, the supervisory host forwards the ACK to the TCP sender. Hence M-TCP maintains the end-to-end TCP semantics even though the TCP connection is split at the supervisory host. When the M-TCP client undergoes a temporary link break, the supervisory host avoids forwarding the ACK of the last byte to the sender and hence the sender TCP goes to the persist state by setting the window size to zero. This avoids retransmission, closing of the congestion window, and slow start at the sender.

1.10.7 Explicit Loss Notification Typically, the problem with TCP lies in the fact that it does not know the exact cause for packet loss, and hence has to invariably assume congestion loss. An ideal TCP simply retransmits the lost packets without any congestion control mechanism. The MAC layer, however, can identify the reason for the packet loss. Once the MAC layer detects that either a handoff is about to occur or realizes that the actual cause of the packet loss is not congestion, then it immediately informs the TCP layer of the possibility of a non-congestion loss. The crux of the strategy is to detect loss at MN and send an explicit loss notification (ELN) to the sender. The sender does not reduce window size on receiving the ELN as this message implies that there was an error and not congestion. This technique avoids slow start and can handle encrypted data. However, the protocol layer software at the MAC layer of MN needs to be changed. Further, the information conveyed by the MAC layer may not always be reliable.

1.10.8 WTCP

WTCP aims at revamping the transport protocol for the wireless domain using

(a) rate-based transmission at the source,

(b) inter-packet separation at the receiver as the congestion metric,

(c) mechanisms for detecting the reason for packet loss, and

(d) bandwidth estimation, as some of the underlying principles. A unique characteristic of WTCP is the attempt to separate the congestion control and reliability mechanisms. WTCP uses separate sequence numbers for congestion control and reliability mechanisms in order to distinguish the two. The reliability mechanism involves a combination of selective and cumulative acknowledgments, and takes into account the reverse-path characteristics for determining the ACK frequency.

1.10.9 TCP SACK

The selective retransmission strategy is more complex and requires more buffer space at the end-points. Hence TCP traditionally uses cumulative acknowledgments and the go-back-N strategy. Using selective retransmit reduces the overhead of retransmission on errors and therefore cannot be ruled out for use in wireless domains. The TCP with selective ACK scheme (TCP SACK) improves TCP performance by allowing the TCP sender to retransmit packets based on the selective ACKs provided by the receiver.

1.10.10 Transaction-OrientedTCP

The TCP connection setup and connection tear-down phases involve a huge overhead in terms of time and also in terms of the number of packets sent. This overhead is very costly, especially if the size of the data is small. An alternative for such transactions is transaction-oriented TCP (TTCP). The motivation behind this approach is to integrate the call setup, the call tear-down, and the actual data transfer into a single transaction, thereby avoiding separate packets for connecting and disconnecting. However, the flip side to the strategy is that changes must be made to TCP, which goes against some of the fundamental

objectives that the changes to TCP must be transparent and must not affect the existing framework. Table 1.6 shows a summary of the various approaches discussed so far. The next section briefly describes the impact of mobility on the performance of TCP.

Table 1.6. Summary of proposed protocols to improve the performance of TCP over wireless

Feature	Snoop TCP	TCP-Unaware Link Layer	Mobile TCP	ITCP	ELN	WTCP	TCP SACK	TTCP
Changes in:								
AP	Yes	Yes	Yes	Yes	No	No	No	No
CN	No	No	No	No	Yes	Yes	Yes	Yes
MN	Yes	No	Yes	Yes	Yes	Yes	No	No
Retransmitting Node	AP	AP	NA*	AP	NA	NA	NA	NA
Single Point Failure	No	No	No	Yes (AP)	No	No	No	No
Handoff Latency	Low	Low	Low	Low	High	High	High	High
Security	Breach at AP	No breach	NA	Breach at AP	No breach	No breach	No breach	No breach
End-to-End Semantics	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Retransmissions by Intermediate Nodes	Yes	Yes	No	Yes	No	No	No	No
Slow Start	Yes	Yes	No	NA	No	No	No	Yes
Buffer at AP	Yes	Yes	No	Yes	No	No	No	No

*Not Applicable

1.10.11 Impact of Mobility Handoffs occur in wireless domains when an MN moves into a new BS's domain (a cell in the cellular context). If the link layer ensures reliable delivery and guarantees zero loss during a handoff, then TCP will be totally unaware of the handoff and no measures need to be taken at the transport layer to support handoff. The only exception to this is when the handoff latency is too large and exceeds the TCP timeout; then the transparency of handoffs to TCP is lost.

Fast Retransmit/Recovery

The usual problem associated with handoffs is that the handoff may lead to packet loss during transit, either as a result of the intermediary routers' failure to allocate adequate buffers or their inability to forward the packets meant for the MN to the new BS. The result of the packet loss during handoff is slow start. The solution involves artificially forcing the sender to go into fast retransmission mode immediately, by sending duplicate acknowledgments after the handoff, instead of going into slow start. The advantage of the strategy is its simplicity and the fact that it requires minimal changes to the existing TCP

structure. However, the scheme does not consider the fact that there may be losses over the wireless links.

Using Multicast

Multicast has been suggested to improve the performance of TCP in the presence of handoffs. The idea is similar to the one used in MRSVP, where the MN is required to define a group of BSs that it is likely to visit in the near future. These include the current cell (or the current BS) the MN is attached to and also the cells (BSs) likely to be visited by it. These BSs are then directed to join the multicast group, the address being the unique multicast address assigned to the MN. Packets destined for MN will have to be subsequently readdressed to the multicast group. In the implementation, only one BS is actually in contact with the MN and is responsible for transmitting the packets to it. If the rest of the BSs in the multicast group are able to buffer the packets addressed to the multicast address, then the loss of packets during the handoff can be significantly minimized. There is a trade-off between buffer allocation at the BSs and the loss during handoff. In practical situations, the number of buffers allocated can be minimized by buffering only when a handoff is likely to occur.

1.11 WAP

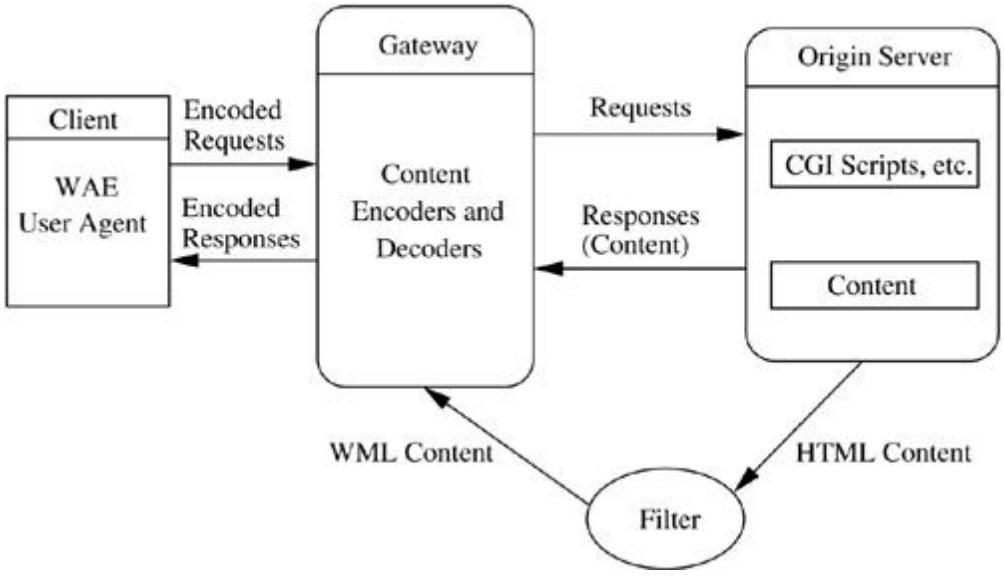
WAP stands for wireless application protocol. This name is a misnomer, because WAP represents a suite of protocols rather than a single protocol. WAP has today become the *de facto* standard for providing data and voice services to wireless handheld devices. WAP aims at integrating a simple lightweight browser also known as a micro-browser into handheld devices, thus requiring minimal amounts of resources such as memory and CPU at these devices. WAP tries to compensate for the shortfalls of the wireless handheld devices and the wireless link (low bandwidth, low processing capabilities, high bit-error rate, and low storage availability) by incorporating more intelligence into the network nodes such as the routers, Web servers, and BSs. The primary objectives of the WAP protocol suite are independence from the wireless network standards, interoperability among service providers, overcoming the shortfalls of the wireless medium (such as low bandwidth, high latency, low

connection stability, and high transmission cost per bit), overcoming the drawbacks of handheld devices (small display, low memory, limited battery power, and limited CPU power), increasing efficiency and reliability, and providing security, scalability, and extensibility.

1.11.1 The WAPModel

WAP adopts a client-server approach. It specifies a proxy server that acts as an interface between the wireless domain and core wired network. This proxy server, also known as a WAP gateway, is responsible for a wide variety of functions such as protocol translation and optimizing data transfer over the wireless medium. Figure 4.8 illustrates the client-server model that WAP employs. The WAP-enabled handset communicates with a Web content server or an origin server [that may provide hypertext markup language (HTML)/common gateway interface (CGI) content] via a WAP gateway. It is at the WAP gateway that the convergence of the wireless and wired domains actually occurs. The gateway receives WAP requests from the handset, and these have to be converted into suitable HTTP requests to be sent to the origin server. If the origin server cannot provide the required information in wireless markup language (WML) form, then there must be an additional filter between the server and the gateway to convert the HTML content into WAP compatible WML content. The gateway may additionally perform functions such as caching and user agent profiling as part of some optimization measures. This is also known as *capability and preference information.* By means of user agent profiling, the MN specifies its characteristics such as hardware characteristics, software capabilities, and user preferences, to the server so that the content can be formatted appropriately to be displayed correctly.

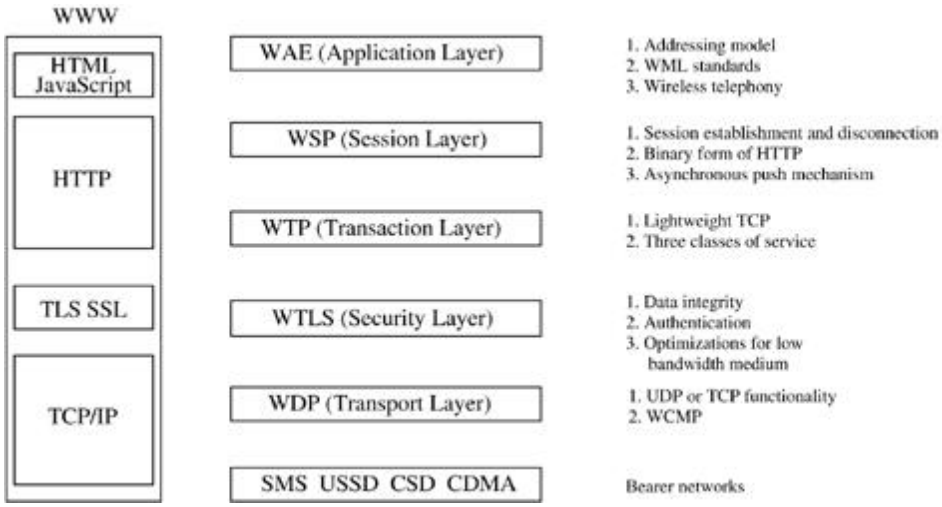
Figure 1.19. The WAP client-server model.



1.11.2 The WAP Protocol Stack

The WAP protocol stack is designed in a layered fashion that allows the architecture to provide an environment that is both extensible and scalable for application development. The WAP architecture allows other services to access the WAP stack at well-defined interfaces. Figure 4.9 gives an overview of the different layers in the WAP protocol suite and also their basic functionalities. This section provides a brief description of some of the important layers in the protocol stack.

Figure 1.20. The WAP protocol stack.



The Wireless Application Environment

The wireless application environment (WAE) has a number of components that address specific issues in the application environment. The WAE provides for an addressing model for accessing both the WWW URLs and other resources specific to the wireless domain using uniform resource identifiers (URIs). The WAE uses WML as the standard markup language, which can be construed as an efficient binary encoded form of the traditional HTML. The WAE also provides a compact scripting language analogous to JavaScript. The WAE also provides for a set of telephony applications through the wireless telephony application interface (WTAI).

Wireless Session Protocol

The Wireless Session Protocol (WSP) establishes a reliable session between the client and the server and also ensures that the session is released in an orderly manner. The push mechanism is a fundamental component of the WAP programming model aimed at reducing the number of requests made by the client to the server. A data server will asynchronously push the information to the registered client(s) efficiently using this mechanism. This is especially useful in multicast and broadcast applications. The WSP provides the equivalent of HTTP in the WWW domain. The core of the WSP design is a binary form of HTTP. A session may be suspended to save power at the clients, but the session reestablishment follows only a small procedure that avoids the overhead of starting a full-fledged session afresh.

Wireless Transaction Protocol

The Wireless Transaction Protocol (WTP) can for all practical purposes be viewed as a lightweight version of TCP. A transaction is defined as a request/response cycle. The WTP has no explicit setup and tear-down phases like TCP, as this would cause a tremendous overhead. There are no security options at the transaction layer in the WAP stack. WTP defines three categories or classes of service: 1. Class 0: Unreliable send (push model) with no ACK. There is no retransmission in case the message is lost. This is essentially a connection-less service. 2. Class 1: Reliable push service, where a request is sent and the responder sends the data as an implicit acknowledgment to the

request. The responder maintains this state for some time to handle possible retransmissions. 3. Class 2: This is the classical request-data-ACK cycle providing a two-way reliable service.

Wireless Transport Layer Security

The objective of the Wireless Transport Layer Security (WTLS) is to provide transport layer security between the WAP client and a WAP server. WTLS is based on the industry standard transport layer security (TLS) protocol with certain features such as datagram support, optimized handshake, and dynamic key refreshing. The primary objectives of WTLS are data integrity, privacy, authentication, and denial of service (DoS) protection. WTLS has capabilities to detect and reject data that is not successfully verified; this protects servers from DoS attacks.

Wireless Datagram Protocol

The Wireless Datagram Protocol (WDP) defines the WAP's transport layer in the protocol suite. The WDP has an adaptation layer that is bearer-specific that helps optimize the data transfer specific to a particular bearer service (such as SMS, USSD, CSD, and CDMA). If the underlying bearer service uses the IP standard user datagram protocol (UDP), then there is no necessity for a separate functionality at the WDP layer as UDP itself is used. The wireless control message protocol (WCMP) is responsible for providing the error-handling mechanisms analogous to Internet control message protocol (ICMP).

1.11.3WAP 2.0 and i-mode

The i-mode (information-mode) system, developed in Japan and a major competitor to WAP, has three main components: a transmission system, a handset, and a language for designing Web pages. The transmission system consists of the existing mobile phone network (which is circuit-switched) and a new packet-switched network. Voice transmission uses the existing mobile phone network while data transmission uses the packet-switched network and is billed based on the number of packets transmitted as opposed to connection time. i-mode uses a subset of HTML called as cHTML (compactHTML). In

contrast, WAP 2.0 was developed by the WAP Forum and is likely to use packet-switched network. WAP 2.0 has new features such as multimedia messaging, pull (request for data, then receive the data) as well as push model (asynchronous data transfer, without requiring explicit request messages, such as stock prices), integrated telephony, interoperability with WAP 1.0, and support for plug-ins in the browser. Unlike i-mode, WAP 2.0 charges the users based on connection time.

1.12 OPTIMIZING WEB OVER WIRELESS

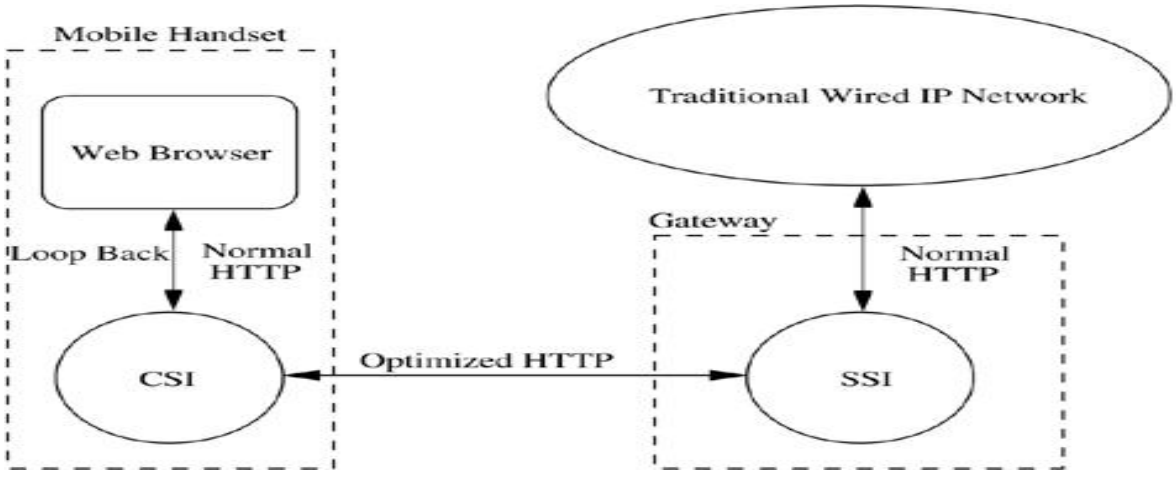
The limitations of wireless networks that provide the motivation for such optimizations are low bandwidth, low reliability, high latency, and high cost per byte transferred. Integrating Web access over wireless devices would have to take into account the drawbacks of the wireless medium and the capabilities of the devices. Systems such as WebExpress are aimed at optimizing routine repetitive browsing; many of the mechanisms suggested may not be suitable for random browsing (*i. e*., there are no perceivable trends in the Web accesses). Web browsers must offer a good interface for the wireless devices, keeping in mind the network, memory, processing power, and power consumption constraints.

1.12.1HTTP Drawbacks

The main protocol on which the Web operates today is the hypertext transfer protocol (HTTP), which is optimized mainly for the wired world. It has a lot of overhead, but it is acceptable when the network bandwidth is an inexpensive resource as in typical wired networks compared to wireless networks. HTTP has drawbacks such as high connection overhead (a new TCP socket is opened for every newHTML object), redundant capabilities transmission (information regarding the browser capabilities is included in every HTTP request), and verbosity (HTTP is ASCII-encoded and hence inherently verbose). The WebExpress system suggests that an Intercept model be applied for Web access over wireless interfaces. This allows the number of requests sent over the wireless channel to be optimized, and also avoids the connection setup overhead over the wireless interface. There are two main entities that are introduced into

the system: the client side interface (CSI) and the server side interface (SSI). The CSI appears as a local Web proxy co-resident with the Web browser on the wireless rendering device, say, a mobile phone or a PDA. The communication between the CSI and the Web browser takes place through the loopback feature of the TCP/IP suite (wherein the host sends a packet to itself using an IP address like 127.0.0.1). The communication between the CSI and the SSI is the only interaction over the wireless network and this uses a reduced HTTP, as discussed later. The SSI communicates with the Web server over the wired network. The SSI could typically be resident at the network gateway or the FA in MobileIP. The intercept model (Figure 4.10) is transparent to browsers and servers, and is also insensitive to changes in HTTP/HTML technology.

Figure 1.21. The intercept model.



1.12.2 Optimizations Four main categories of optimizations that can improve the performance of Web access systems over wireless channels can be identified. These are:

• **Caching:** Current caching technologies are suited for wired applications. Cache objects are either purged at the end of the session or may persist across sessions. But it is advantageous to have cached data persist across browser sessions, as this increases cache hit ratios. Appropriate cache coherency methods are added to detect and change old information.

• **Differencing:** For transaction processing (involving forms) caching techniques do not help as different replies to the same application server are

often different. Still, the fact that these replies tend to be similar can be exploited to reduce the network traffic over the wireless interface. A base object carries fundamental features that do not change across transactions and is created and maintained by both the client and server interfaces. Whenever a new transaction takes place, the server computes the difference stream and only the difference stream is transmitted.

• **Protocol reduction:** This approach aims at reducing the overhead of repeated setup and tear-down of TCP/IP connections for each Web-object to be transmitted. This can be eliminated by establishing a single TCP/IP connection between the CSI and the SSI that will persist for the entire session. The connection setup/tear-down overhead is on the local and wired connections only.

• **Header reduction:** HTTP requests are prefixed with headers that indicate to the origin server the rendering capabilities of the browser and also the various content formats handled by it. The alternative to this is that the CSI sends this information in the first request and SSI records this information. For every subsequent request sent by the CSI, the SSI automatically inserts this capability list into each packet meant for the origin server.

UNIT-II

ADHOC WIRELESS NETWORKS

2.1 INTRODUCTION The principle behind ad hoc networking is multi-hop relaying, which traces its roots back to 500 B.C. Darius I (522-486 B.C.), the king of Persia, devised an innovative communication system that was used to send messages and news from his capital to the remote provinces of his empire by means of a line of shouting men positioned on tall structures or heights. This system was more than 25 times faster than normal messengers available at that time. The use of ad hoc voice communication was used in many ancient/tribal societies with a string of repeaters of drums, trumpets, or horns. In 1970, Norman Abramson and his fellow researchers at the University of Hawaii invented the ALOHA net, an innovative communication system for linking together the universities of the Hawaiian islands. ALOHA net utilized single-hop wireless packet switching and a multiple access solution for sharing a single channel. Even though ALOHA net was originally implemented for a fixed single-hop wireless network, the basic idea was compelling and applicable to any environment where access to a common resource had to be negotiated among a set of uncoordinated nodes. The success and

novelty of ALOHA net triggered widespread interest in different directions of computer communication, including the work that led to the development of Ethernet by Robert Metcalfe and the packet radio network (PRNET) project sponsored by the defence advanced research projects agency (DARPA). The PRNET project was aimed at developing a packet wireless network for military applications. Even though the initial attempt had a centralized control, it quickly evolved into a distributed multi-hop wireless communication system that could operate over a large geographical area. Each mobile node had a broadcast radio interface that provided many advantages such as the use of a single channel, simpler channel management techniques, and the ease of supporting mobility. PRNET used a combination of ALOHA and carrier sense multiple access (CSMA) for access to the shared radio channel. The radio interface employed the direct-sequence (DS) spread spectrum scheme. The system was designed to self-organize, self-configure, and detect radio connectivity for the dynamic operation of a routing protocol without any support from fixed infrastructure. The major issues that the PRNET project faced include those of obtaining, maintaining, and utilizing the topology information, error and flow control over the wireless links, reconfiguration of paths to handle path breaks arising due to the mobility of nodes and routers, processing and storage capability of nodes, and distributed channel sharing. The successful demonstrations of the PRNET proved the feasibility and efficiency of infrastructureless networks and their applications for civilian and military purposes. DARPA extended the work on multi-hop wireless networks through the survivable radio networks (SURAN) project that aimed at providing ad hoc networking with small, low-cost, low-power devices with efficient protocols and improved scalability and survivability (the ability of a network to survive the failure of network nodes and links). During the 1980s, research on military applications was extensively funded across the globe. Realizing the necessity of open standards in this emerging area of computer communication, a working group within the Internet Engineering Task Force (IETF), termed the mobile ad hoc networks (MANET) working group , was formed to standardize the protocols and functional specifications of ad hoc wireless networks. The vision of the IETF effort in the MANET working group is to provide improved standardized routing functionality to support self-organizing mobile networking infrastructure. In 1994, the Swedish communication equipment maker Ericsson proposed to develop a short range, low-power, low-complexity, and inexpensive radio interface and associated communication protocols referred to as *Bluetooth* for ubiquitous connectivity among heterogeneous devices, as discussed in Section 2.5. This effort was later taken over by a Special Interest Group (SIG) formed by several major computer and telecommunication vendors such as 3Com, Ericsson, IBM, Intel, Lucent, Microsoft, Motorola, Nokia, and Toshiba. The Bluetooth SIG aims at delivering a universal solution for connectivity among heterogeneous devices. This is one of the first commercial realizations of ad hoc wireless networking. Bluetooth standardizes the single-hop point-to-point wireless link that helps in exchanging voice or data, and formation of *piconets* that are formed by a group of nodes in a smaller geographical region where every node can reach every other node in the group within a single-hop. Multiple piconets can form a *scatternet*, which necessitates the use of multi-hop routing protocols. Even though ad hoc wireless networks are

expected to work in the absence of any fixed infrastructure, recent advances in wireless network architectures reveal interesting solutions that enable the mobile ad hoc nodes to function in the presence of infrastructure. Multi-hop cellular networks (MCNs) and self-organizing packet radio ad hoc networks with overlay (SOPRANO) are examples of such types of networks. These hybrid architectures (which combine the benefits of cellular and ad hoc wireless networks) improve the capacity of the system significantly. Even with all the promises that are offered by ad hoc wireless networks, successful commercial deployment requires realistic solutions to different problems, including support for QoS provisioning and real-time applications, pricing, cooperative functioning, energy-efficient relaying, load balancing, and support for multicast traffic.

2.1.1 Cellular and Ad Hoc Wireless Networks

Figure 2.1 shows a representation of different wireless networks. The current cellular wireless networks (depicted in Figure 2.2) are classified as the infrastructure dependent networks. The path setup for a call between two nodes, say, node *C* to node *E,* is completed through the base station as illustrated in Figure 2.2.

Figure 2.1. Cellular and ad hoc wireless networks.

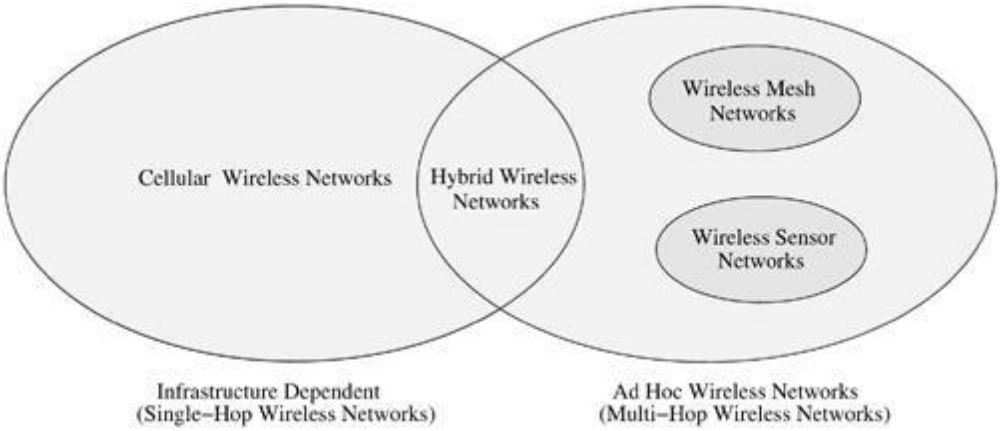
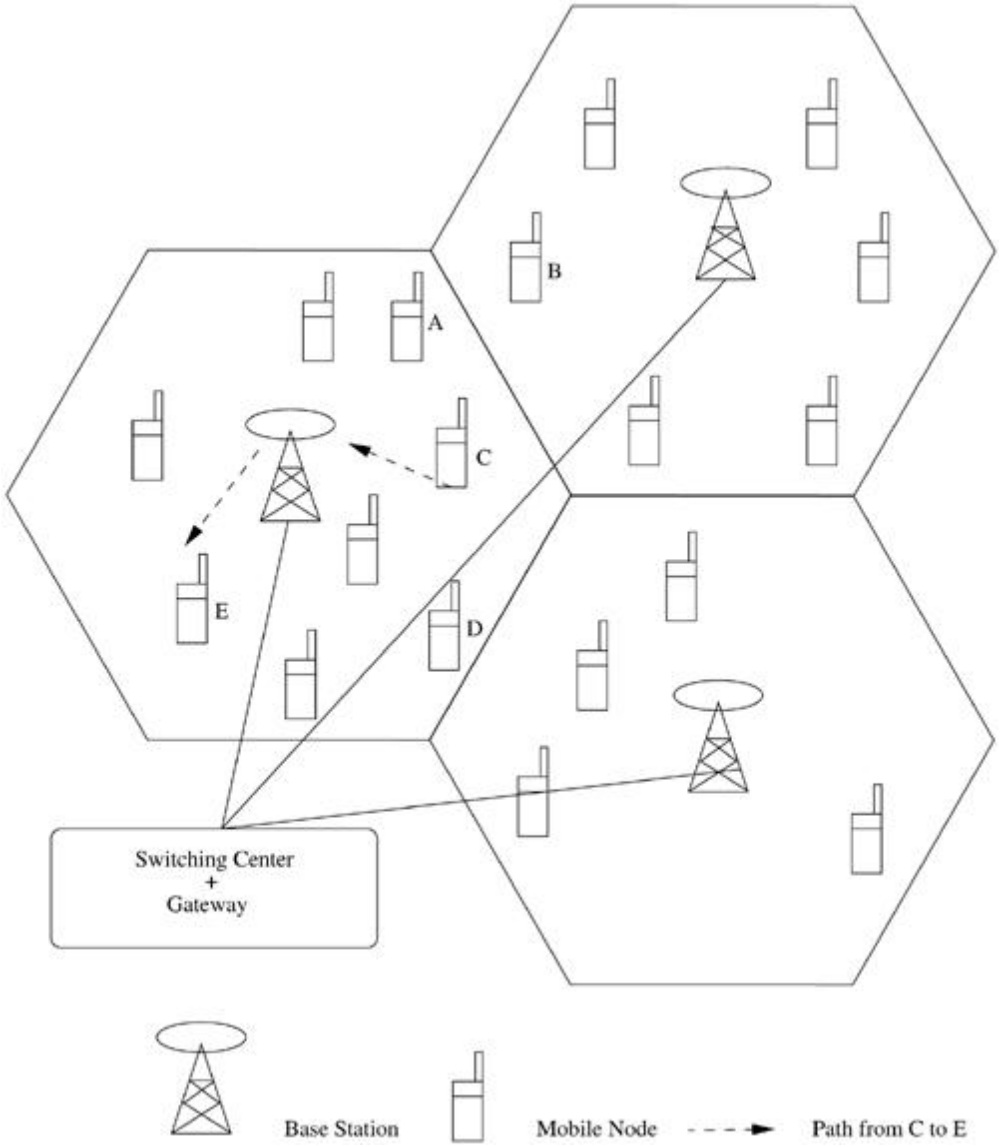
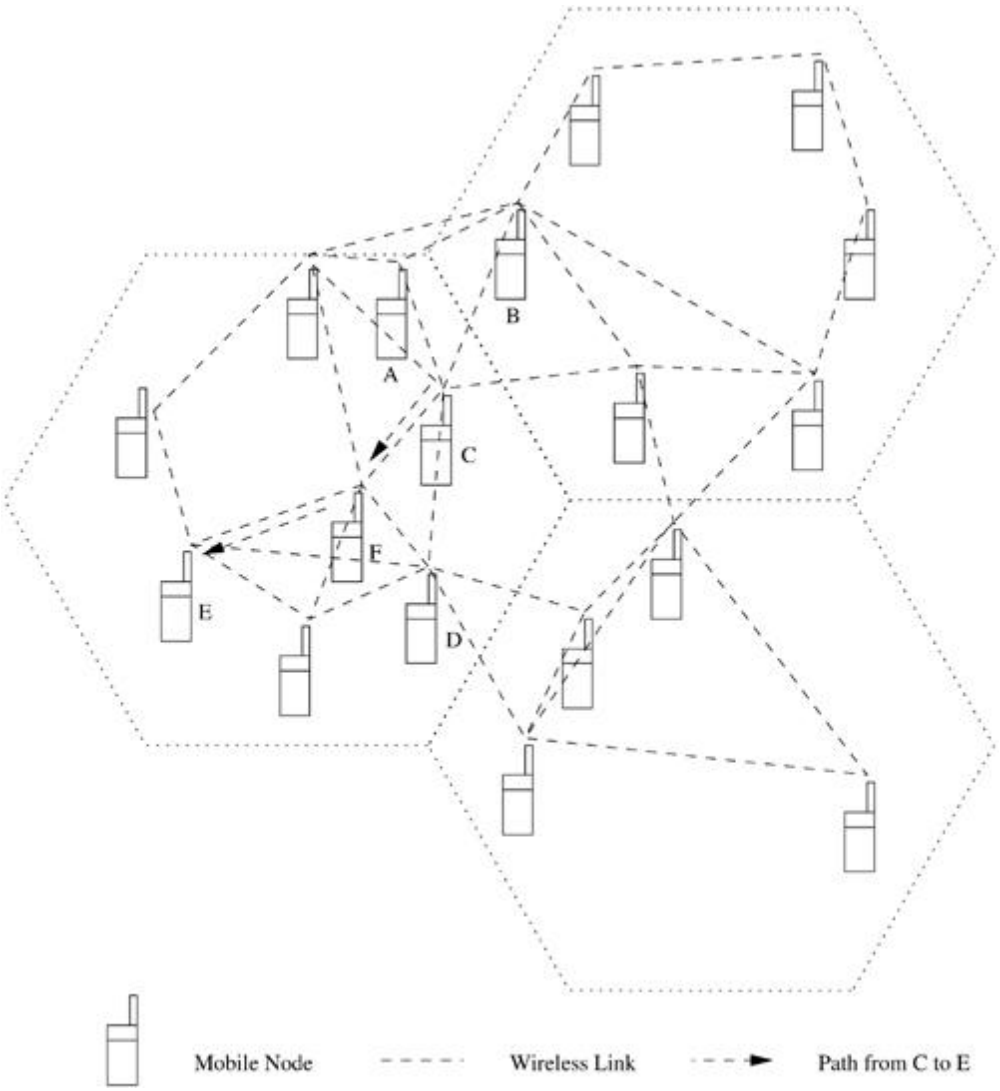


Figure 2.2. A cellular network.



Ad hoc wireless networks are defined as the category of wireless networks that utilize multi-hop radio relaying and are capable of operating without the support of any fixed infrastructure (hence they are also called infrastructureless networks). The absence of any central coordinator or base station makes the routing a complex one compared to cellular networks. Ad hoc wireless network topology for the cellular network shown in Figure 2.2 is illustrated in Figure 2.9. Note that in Figure 2.9 the cell boundaries are shown purely for comparison with the cellular network in Figure 2.2 and do not carry any special significance. The path setup for a call between two nodes, say, node *C* to node *E,* is completed through the intermediate mobile node *F,* as illustrated in Figure 2.9. Wireless mesh networks and wireless sensor networks are specific examples of ad hoc wireless networks.

Figure 2.3. An ad hoc wireless network.



The major differences between cellular networks and ad hoc wireless networks are summarized in Table 2.1. The presence of base stations simplifies routing and resource management in a cellular network as the routing decisions are made in a centralized manner with more information about the destination node. But in an ad hoc wireless network, the routing and resource management are done in a distributed manner in which all nodes coordinate to enable communication among themselves. This requires each node to be more intelligent so that it can function both as a network host for transmitting and receiving data and as a network router for routing packets from other nodes. Hence the mobile nodes in ad hoc wireless networks are more complex than their counterparts in cellular networks.

Table 2.1. Differences between cellular networks and ad hoc wireless networks

Cellular Networks	Ad Hoc Wireless Networks
Fixed infrastructure-based	Infrastructure-less
Single-hop wireless links	Multi-hop wireless links
Guaranteed bandwidth (designed for voice traffic)	Shared radio channel (more suitable for best-effort data traffic)
Centralized routing	Distributed routing
Circuit-switched (evolving toward packet switching)	Packet-switched (evolving toward emulation of circuit switching)
Seamless connectivity (low call drops during handoffs)	Frequent path breaks due to mobility
High cost and time of deployment	Quick and cost-effective deployment
Reuse of frequency spectrum through geographical channel reuse	Dynamic frequency reuse based on carrier sense mechanism
Easier to achieve time synchronization	Time synchronization is difficult and consumes bandwidth
Easier to employ bandwidth reservation	Bandwidth reservation requires complex medium access control protocols
Application domains include mainly civilian and commercial sectors	Application domains include battlefields, emergency search and rescue operations, and collaborative computing
High cost of network maintenance (backup power source, staffing, etc.)	Self-organization and maintenance properties are built into the network
Mobile hosts are of relatively low complexity	Mobile hosts require more intelligence (should have a transceiver as well as routing/switching capability)
Major goals of routing and call admission are to maximize the call acceptance ratio and minimize the call drop ratio	Main aim of routing is to find paths with minimum overhead and also quick reconfiguration of broken paths
Widely deployed and currently in the third generation of evolution	Several issues are to be addressed for successful commercial deployment even though widespread use exists in defense

2.1.2 Applications of Ad Hoc Wireless Networks Ad hoc wireless networks, due to their quick and economically less demanding deployment, find applications in several areas. Some of these include: military applications, collaborative and distributed computing, emergency operations, wireless mesh networks, wireless sensor networks, and hybrid wireless network architectures.

Military Applications

Ad hoc wireless networks can be very useful in establishing communication among a group of soldiers for tactical operations. Setting up a fixed infrastructure for communication among a group of soldiers in enemy territories or in inhospitable terrains may not be possible. In such environments, ad hoc wireless networks provide the required communication mechanism quickly. Another application in this area can be the coordination of military objects moving at

high speeds such as fleets of airplanes or warships. Such applications require quick and reliable communication. Secure communication is of prime importance as eavesdropping or other security threats can compromise the purpose of communication or the safety of personnel involved in these tactical operations. They also require the support of reliable and secure multimedia multicasting. For example, the leader of a group of soldiers may want to give an order to all the soldiers or to a set of selected personnel involved in the operation. Hence, the routing protocol in these applications should be able to provide quick, secure, and reliable multicast communication with support for real-time traffic.

As the military applications require very secure communication at any cost, the vehicle-mounted nodes can be assumed to be very sophisticated and powerful. They can have multiple high-power transceivers, each with the ability to hop between different frequencies for security reasons. Such communication systems can be assumed to be equipped with long-life batteries that might not be economically viable for normal usage. They can even use other services such as location tracking [using the global positioning system (GPS)] or other satellite-based services for efficient communication and coordination. Resource constraints such as battery life and transmitting power may not exist in certain types of applications of ad hoc wireless networks. For example, the ad hoc wireless network formed by a fleet of military tanks may not suffer from the power source constraints present in the ad hoc network formed by a set of wearable devices used by the foot soldiers. In short, the primary nature of the communication required in a military environment enforces certain important requirements on ad hoc wireless networks, namely, reliability, efficiency, secure communication, and support for multicast routing.

Collaborative and Distributed Computing

Another domain in which the ad hoc wireless networks find applications is collaborative computing. The requirement of a temporary communication infrastructure for quick communication with minimal configuration among a group of people in a conference or gathering necessitates the formation of an ad hoc wireless network. For example, consider a group of researchers who want to share their research findings or presentation materials during a conference, or a lecturer distributing notes to the class on the fly. In such cases, the formation of an ad hoc wireless network with the necessary support for reliable multicast routing can serve the purpose. The distributed file sharing applications utilized in such situations do not require the level of security expected in a military environment. But the reliability of data transfer is of high importance. Consider the example where a node that is part of an ad hoc wireless network has to distribute a file to other nodes in the network. Though this application does not demand the communication to be interruption-free, the goal of the transmission is that all the desired receivers must have the replica of the transmitted file. Other applications such as streaming of multimedia objects among the participating nodes in an ad hoc wireless network may require support for soft real-time communication. The users of such applications prefer economical and portable devices, usually powered by battery sources. Hence, a mobile node may drain its battery and can have varying transmission power, which may result in unidirectional links with its

neighbors. Devices used for such applications could typically be laptops with add-on wireless interface cards, enhanced personal digital assistants (PDAs), or mobile devices with high processing power. In the presence of such heterogeneity, interoperability is an important issue.

Emergency Operations

Ad hoc wireless networks are very useful in emergency operations such as search and rescue, crowd control, and commando operations. The major factors that favor ad hoc wireless networks for such tasks are self-configuration of the system with minimal overhead, independent of fixed or centralized infrastructure, the nature of the terrain of such applications, the freedom and flexibility of mobility, and the unavailability of conventional communication infrastructure. In environments where the conventional infrastructure-based communication facilities are destroyed due to a war or due to natural calamities such as earthquakes, immediate deployment of ad hoc wireless networks would be a good solution for coordinating rescue activities. Since the ad hoc wireless networks require minimum initial network configuration for their functioning, very little or no delay is involved in making the network fully operational. The above-mentioned scenarios are unexpected, in most cases unavoidable, and can affect a large number of people. Ad hoc wireless networks employed in such circumstances should be distributed and scalable to a large number of nodes. They should also be able to provide fault tolerant communication paths. Real-time communication capability is also important since voice communication predominates data communication in such situations.

Wireless Mesh Networks

Wireless mesh networks are ad hoc wireless networks that are formed to provide an alternate communication infrastructure for mobile or fixed nodes/users, without the spectrum reuse constraints and the requirements of network planning of cellular networks. The mesh topology of wireless mesh networks provides many alternate paths for a data transfer session between a source and destination, resulting in quick reconfiguration of the path when the existing path fails due to node failures. Wireless mesh networks provide the most economical data transfer capability coupled with the freedom of mobility. Since the infrastructure built is in the form of small radio relaying devices fixed on the rooftops of the houses in a residential zone as shown in Figure 2.10, or similar devices fitted on the lamp posts as depicted in Figure 2.11, the investment required in wireless mesh networks is much less than what is required for the cellular network counterparts. Such networks are formed by placing wireless relaying equipment spread across the area to be covered by the network. The possible deployment scenarios of wireless mesh networks include: residential zones (where broadband Internet connectivity is required), highways (where a communication facility for moving automobiles is required), business zones (where an alternate communication system to cellular networks is required), important civilian regions (where a high degree of service availability is required), and university campuses (where inexpensive campus-wide network coverage can be provided). Wireless mesh networks should be capable of self-organization and maintenance. The ability of the network to overcome single

or multiple node failures resulting from disasters makes it convenient for providing the communication infrastructure for strategic applications. The major advantages of wireless mesh networks are support for a high data rate, quick and low cost of deployment, enhanced services, high scalability, easy extendability, high availability, and low cost per bit. Wireless mesh networks operate at the license-free ISM bands around 2.4 GHz and 5 GHz. Depending on the technology used for the physical layer and MAC layer communication, data rates of 2 Mbps to 60 Mbps can be supported. For example, if IEEE 802.11a is used, a maximum data rate of 54 Mbps can be supported. The deployment time required for this network is much less than that provided by other infrastructure-based networks. Incremental deployment or partial batch deployment can also be done. Wireless mesh networks provide a very economical communication infrastructure in terms of both deployment and data transfer costs. Services such as smart environments that update information about the environment or locality to the visiting nodes are also possible in such an environment. A truck driver can utilize enhanced location discovery services, and hence spotting his location on an updated digital map is possible. Mesh networks scale well to provide support to a large number of nodes. Even at a very high density of mobile nodes, by employing power control at the mobile nodes and relay nodes, better system throughput and support for a large number of users can be achieved. But in the case of cellular networks, improving scalability requires additional infrastructural nodes, which in turn involves high cost. As mentioned earlier, mesh networks provide expandability of service in a cost effective manner. Partial roll out and commissioning of the network and extending the service in a seamless manner without affecting the existing installation are the benefits from the viewpoint of service providers. Wireless mesh networks provide very high availability compared to the existing cellular architecture, where the presence of a fixed base station that covers a much larger area involves the risk of a single point of failure.

Figure 2.10. Wireless mesh network operating in a residential zone.

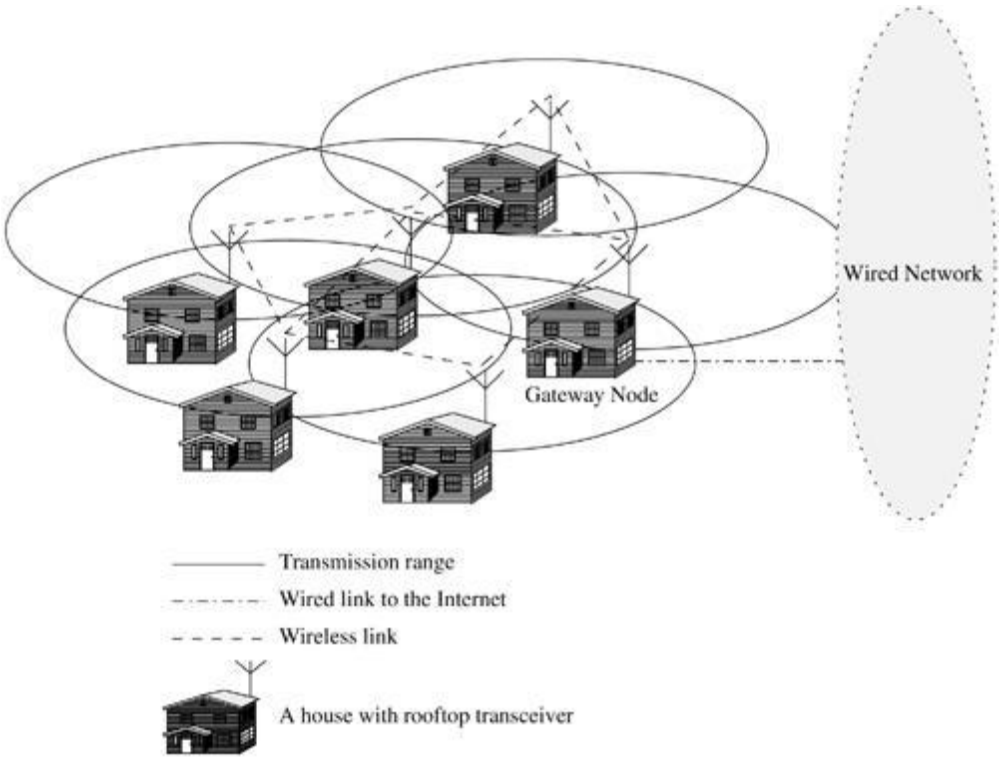
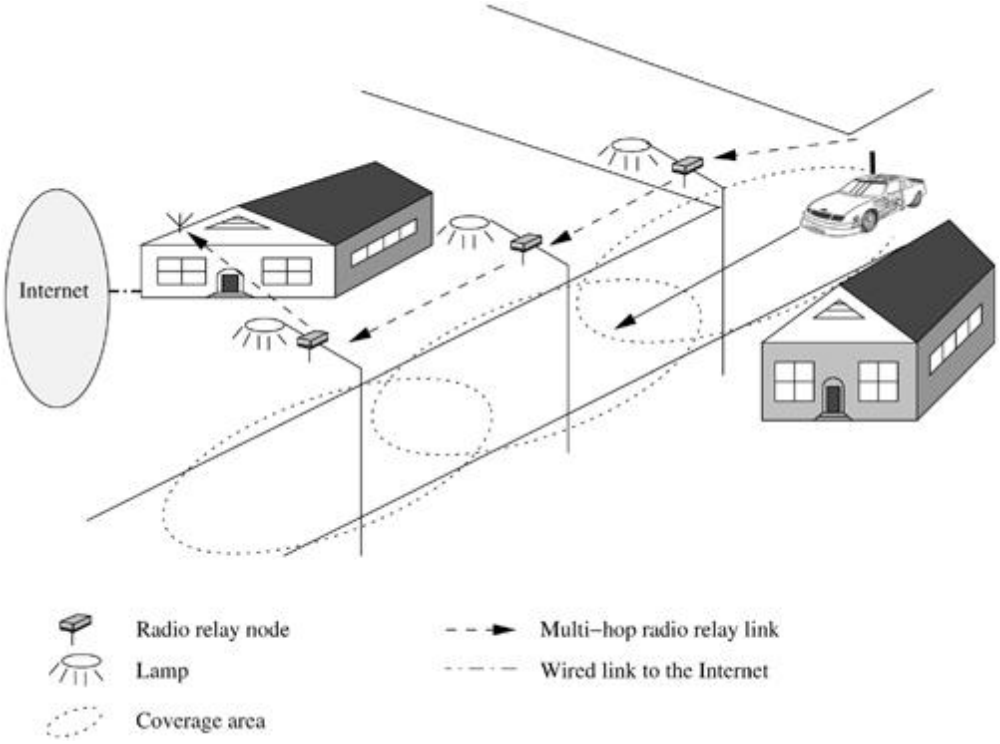


Figure 2.11. Wireless mesh network covering a highway.



Sensor networks are a special category of ad hoc wireless networks that are used to provide a wireless communication infrastructure among the sensors deployed in a specific application domain. Recent advances in wireless communication technology and research in ad hoc wireless networks have made smart sensing a reality. Sensor nodes are tiny devices that have the capability of sensing physical parameters, processing the data gathered, and communicating over the network to the monitoring station. A sensor network is a collection of a large number of sensor nodes that are deployed in a particular region. The activity of sensing can be periodic or sporadic. An example for the periodic type is the sensing of environmental factors for the measurement of parameters such as temperature, humidity, and nuclear radiation. Detecting border intrusion, sensing the temperature of a furnace to prevent it rising beyond a threshold, and measuring the stress on critical structures or machinery are examples of the sensing activities that belong to the sporadic type. Some of the domains of application for sensor networks are military, health care, home security, and environmental monitoring. The issues that make sensor networks a distinct category of ad hoc wireless networks are the following:

• **Mobility of nodes:** Mobility of nodes is not a mandatory requirement in sensor networks. For example, the nodes deployed for periodic monitoring of soil properties are not required to be mobile. However, the sensor nodes that are fitted on the bodies of patients in a post-surgery ward of a hospital may be designed to support limited or partial mobility. In general, sensor networks need not in all cases be designed to support mobility of sensor nodes.

• **Size of the network:** The number of nodes in the sensor network can be much larger than that in a typical ad hoc wireless network.

• **Density of deployment:** The density of nodes in a sensor network varies with the domain of application. For example, military applications require high availability of the network, making redundancy a high priority.

• **Power constraints:** The power constraints in sensor networks are much more stringent than those in ad hoc wireless networks. This is mainly because the sensor nodes are expected to operate in harsh environmental or geographical conditions, with minimum or no human supervision and maintenance. In certain cases, the recharging of the energy source is impossible. Running such a network, with nodes powered by a battery source with limited energy, demands very efficient protocols at network, data link, and physical layer. The power sources used in sensor networks can be classified into the following three categories:

– **Replenishable power source:** In certain applications of sensor networks, the power source can be replaced when the existing source is fully drained (*e.g.*, wearable sensors that are used to sense body parameters).

– **Non-replenishable power source:** In some specific applications of sensor networks, the power source cannot be replenished once the network has been deployed. The replacement of the

sensor node is the only solution to it (*e.g.*, deployment of sensor nodes in a remote, hazardous terrain).

– **Regenerative power source:** Power sources employed in sensor networks that belong to this category have the capability of regenerating power from the physical parameter under measurement. For example, the sensor employed for sensing temperature at a power plant can use power sources that can generate power by using appropriate transducers.

• **Data/information fusion:** The limited bandwidth and power constraints demand aggregation of bits and information at the intermediate relay nodes that are responsible for relaying. Data fusion refers to the aggregation of multiple packets into one before relaying it. This mainly aims at reducing the bandwidth consumed by redundant headers of the packets and reducing the media access delay involved in transmitting multiple packets. Information fusion aims at processing the sensed data at the intermediate nodes and relaying the outcome to the monitor node.

• **Traffic distribution:** The communication traffic pattern varies with the domain of application in sensor networks. For example, the environmental sensing application generates short periodic packets indicating the status of the environmental parameter under observation to a central monitoring station. This kind of traffic demands low bandwidth. The sensor network employed in detecting border intrusions in a military application generates traffic on detection of certain events; in most cases these events might have time constraints for delivery. In contrast, ad hoc wireless networks generally carry user traffic such as digitized and packetized voice stream or data traffic, which demands higher bandwidth.

Hybrid Wireless Networks

One of the major application areas of ad hoc wireless networks is in hybrid wireless architectures such as multi-hop cellular networks (MCNs) and and integrated cellular ad hoc relay (iCAR) networks . The tremendous growth in the subscriber base of existing cellular networks has shrunk the cell size up to the pico-cell level. The primary concept behind cellular networks is geographical channel reuse. Several techniques such as cell sectoring, cell resizing, and multitier cells have been proposed to increase the capacity of cellular networks. Most of these schemes also increase the equipment cost. The capacity (maximum throughput) of a cellular network can be increased if the network incorporates the properties of multi-hop relaying along with the support of existing fixed infrastructure. MCNs combine the reliability and support of fixed base stations of cellular networks with flexibility and multi-hop relaying of ad hoc wireless networks. The MCN architecture is depicted in Figure 2.6. In this architecture, when two nodes (which are not in direct transmission range) in the same cell want to communicate with each other, the connection is routed through multiple wireless hops over the intermediate nodes. The base station maintains the information about the topology of the network for efficient routing. The base station may or may not be involved in this multi-hop path. Suppose node A wants to communicate with node B. If all nodes are capable of operating in MCN mode, node A can reach

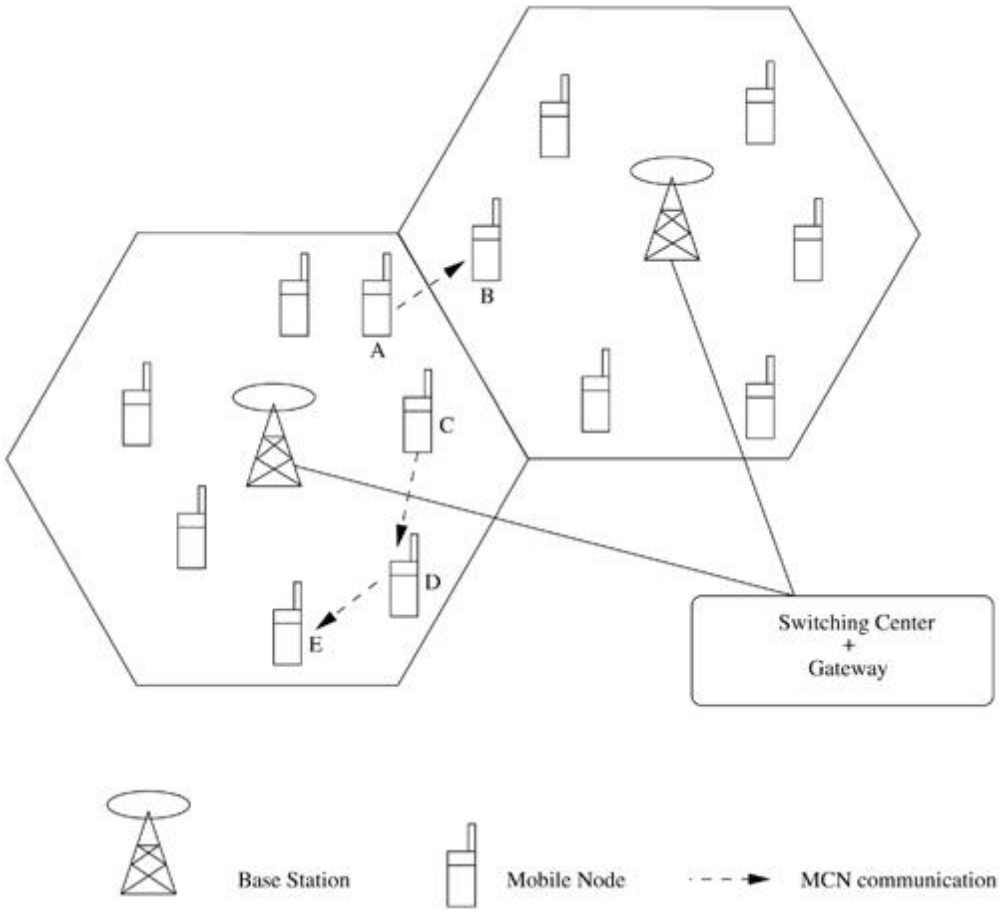
node B directly if the node B is within node A's transmission range. When node C wants to communicate with node E and both are in the same cell, node C can reach node E through node D, which acts as an intermediate relay node. Such hybrid wireless networks can provide high capacity resulting in lowering the cost of communication to less than that in single-hop cellular networks. The major advantages of hybrid wireless networks are as follows:

• Higher capacity than cellular networks obtained due to the better channel reuse provided by reduction of transmission power, as mobile nodes use a power range that is a fraction of the cell radius.

• Increased flexibility and reliability in routing. The flexibility is in terms of selecting the best suitable nodes for routing, which is done through multiple mobile nodes or through base stations, or by a combination of both. The increased reliability is in terms of resilience to failure of base stations, in which case a node can reach other nearby base stations using multi-hop paths.

• Better coverage and connectivity in holes (areas that are not covered due to transmission difficulties such as antenna coverage or the direction of antenna) of a cell can be provided by means of multiple hops through intermediate nodes in the cell.

Figure 2.6. MCN architecture.



2.2 ISSUES IN AD HOC WIRELESS NETWORKS

This section discusses the major issues and challenges that need to be considered when an ad hoc wireless system is to be designed. The deployment considerations for installation, operation, and maintenance of ad hoc wireless networks are also provided. The major issues that affect the design, deployment, and performance of an ad hoc wireless system are as follows:

• Medium access scheme

• Routing

• Multicasting

• Transport layer protocol

• Pricing scheme

• Quality of service provisioning

• Self-organization • Security

• Energy management

• Addressing and service discovery

• Scalability

• Deployment considerations

2.2.1 Medium Access Scheme The primary responsibility of a medium access control (MAC) protocol in ad hoc wireless networks is the distributed arbitration for the shared channel for transmission of packets. The performance of any wireless network hinges on the MAC protocol, more so for ad hoc wireless networks. The major issues to be considered in designing a MAC protocol for ad hoc wireless networks are as follows:

• **Distributed operation:** The ad hoc wireless networks need to operate in environments where no centralized coordination is possible. The MAC protocol design should be fully distributed involving minimum control overhead. In the case of polling-based MACprotocols, partial coordination is required.

• **Synchronization:** The MAC protocol design should take into account the requirement of time synchronization. Synchronization is mandatory for TDMA-based systems for management of transmission and reception slots. Synchronization involves usage of scarce resources such as bandwidth and battery power. The control packets used for synchronization can also increase collisions in the network.

• **Hidden terminals:** Hidden terminals are nodes that are hidden (or not reachable) from the sender of a data transmission session, but are reachable to the receiver of the session. In such cases, the hidden terminal can cause collisions at the receiver node. The presence of hidden terminals can significantly reduce the throughput of a MAC protocol used in ad hoc wireless networks. Hence the MAC protocol should be able to alleviate the effects of hidden terminals.

• **Exposed terminals:** Exposed terminals, the nodes that are in the transmission range of the sender of an on-going session, are prevented from making a transmission. In order to improve the efficiency of the MAC protocol, the exposed nodes should be allowed to transmit in a controlled fashion without causing collision to the on-going data transfer.

• **Throughput:** The MAC protocol employed in ad hoc wireless networks should attempt to maximize the throughput of the system. The important considerations for throughput enhancement are minimizing the occurrence of collisions, maximizing channel utilization, and minimizing control overhead.

• **Access delay:** The access delay refers to the average delay that any packet experiences to get transmitted. The MAC protocol should attempt to minimize the delay.

• **Fairness:** Fairness refers to the ability of the MAC protocol to provide an equal share or weighted share of the bandwidth to all competing nodes. Fairness can be either node-based or flow-based. The former attempts to provide an equal bandwidth share for competing nodes whereas the latter provides an equal share for competing data transfer sessions. In ad hoc wireless networks, fairness is important due to the multi-hop relaying done by the nodes. An unfair relaying load for a node results in draining the resources of that node much faster than that of other nodes.

• **Real-time traffic support:** In a contention-based channel access environment, without any central coordination, with limited bandwidth, and with location-dependent contention, supporting time-sensitive traffic such as voice, video, and real-time data requires explicit support from the MAC protocol.

• **Resource reservation:** The provisioning of QoS defined by parameters such as bandwidth, delay, and jitter requires reservation of resources such as bandwidth, buffer space, and processing power. The inherent mobility of nodes in ad hoc wireless networks makes such reservation of resources a difficult task. A MAC protocol should be able to provide mechanisms for supporting resource reservation and QoS provisioning.

• **Ability to measure resource availability:** In order to handle the resources such as bandwidth efficiently and perform call admission control based on their availability, the MACprotocol should be able to provide an estimation of resource availability at every node. This can also be used for making congestion-control decisions.

• **Capability for power control:** The transmission power control reduces the energy consumption at the nodes, causes a decrease in interference at neighboring nodes, and increases frequency reuse. Support for power control at the MAC layer is very important in the ad hoc wireless environment.

• **Adaptive rate control:** This refers to the variation in the data bit rate achieved over a channel. A MAC protocol that has adaptive rate control can make use of a high data rate when the sender and receiver are nearby and adaptively reduce the data rate as they move away from each other.

• **Use of directional antennas:** This has many advantages that include increased spectrum reuse, reduction in interference, and reduced power consumption. Most of the existing MAC protocols that use omni directional antennas do not work with directional antennas.

2.2.2 Routing The responsibilities of a routing protocol include exchanging the route information; finding a feasible path to a destination based on criteria such as hop length, minimum power required, and lifetime of the wireless link; gathering information about the path breaks; mending the broken paths expending minimum processing power and bandwidth; and utilizing minimum bandwidth. The major challenges that a routing protocol faces are as follows:

• **Mobility:** One of the most important properties of ad hoc wireless networks is the mobility associated with the nodes. The mobility of nodes results in frequent path breaks, packet collisions, transient loops, stale routing information, and difficulty in resource reservation. A good routing protocol should be able to efficiently solve all the above issues.

• **Bandwidth constraint:** Since the channel is shared by all nodes in the broadcast region (any region in which all nodes can hear all other nodes), the bandwidth available per wireless link depends on the number of nodes and the traffic they handle. Thus only a fraction of the total bandwidth is available for every node.

• **Error-prone and shared channel:** The bit error rate (BER) in a wireless channel is very high (of the order of 10^{-5} to 10^{-3}) compared to that in its wired counterparts (of the order of 10^{-12} to 10^{-9}). Routing protocols designed for ad hoc wireless networks should take this into account. Consideration of the state of the wireless link, signal-to-noise ratio, and path loss for routing in ad hoc wireless networks can improve the efficiency of the routing protocol.

• **Location-dependent contention:** The load on the wireless channel varies with the number of nodes present in a given geographical region. This makes the contention for the channel high when the number of nodes increases. The high contention for the channel results in a high number of collisions and a subsequent wastage of bandwidth. A good routing protocol should have built-in mechanisms for distributing the network load uniformly across the network so that the formation of regions where channel contention is high can be avoided.

• **Other resource constraints:** The constraints on resources such as computing power, battery power, and buffer storage also limit the capability of a routing protocol. The major requirements of a routing protocol in ad hoc wireless networks are the following:

• **Minimum route acquisition delay:** The route acquisition delay for a node that does not have a route to a particular destination node should be as minimal as possible. This delay may vary with the size of the network and the network load.

• **Quick route reconfiguration:** The unpredictable changes in the topology of the network require that the routing protocol be able to quickly perform route reconfiguration in order to handle path breaks and subsequent packet losses.

• **Loop-free routing:** This is a fundamental requirement of any routing protocol to avoid unnecessary wastage of network bandwidth. In ad hoc wireless networks, due to the random movement of nodes, transient loops may form in the route thus established. A routing protocol should detect such transient routing loops and take corrective actions.

• **Distributed routing approach:** An ad hoc wireless network is a fully distributed wireless network and the use of centralized routing approaches in such a network may consume a large amount of bandwidth.

• **Minimum control overhead:** The control packets exchanged for finding a new route and maintaining existing routes should be kept as minimal as possible. The control packets consume precious bandwidth and can cause collisions with data packets, thereby reducing network throughput.

• **Scalability:** Scalability is the ability of the routing protocol to scale well (*i.e.*, perform efficiently) in a network with a large number of nodes. This requires minimization of control overhead and adaptation of the routing protocol to the network size.

• **Provisioning of QoS:** The routing protocol should be able to provide a certain level of QoS as demanded by the nodes or the category of calls. The QoS parameters can be bandwidth, delay, jitter, packet delivery ratio, and throughput. Supporting differentiated classes of service may be of importance in tactical operations.

• **Support for time-sensitive traffic:** Tactical communications and similar applications require support for time-sensitive traffic. The routing protocol should be able to support both hard realtime and soft real-time traffic.

• **Security and privacy:** The routing protocol in ad hoc wireless networks must be resilient to threats and vulnerabilities. It must have inbuilt capability to avoid resource consumption, denial-of- service, impersonation, and similar attacks possible against an ad hoc wireless network.

2.2.3 Multicasting

Multicasting plays an important role in the typical applications of ad hoc wireless networks, namely, emergency search-and-rescue operations and military communication. In such an environment, nodes form groups to carry out certain tasks that require point-to-multipoint and multipoint-to-multipoint voice and data communication. The arbitrary movement of nodes changes the topology dynamically in an unpredictable manner. The mobility of nodes, with the constraints of power source and bandwidth, makes multicast routing very challenging. Traditional wired network multicast protocols such as core based trees (CBT), protocol independent multicast (PIM), and distance vector multicast routing protocol (DVMRP) do not perform well in ad hoc wireless networks because a tree-based multicast structure is highly unstable and needs to be frequently readjusted to include broken links. Use of any global routing structure such as the link-state table results in high control overhead. The use of single-link connectivity among the nodes in a multicast group results in a tree-shaped multicast routing topology. Such a tree-shaped topology provides high multicast efficiency, with low packet delivery ratio due to the frequent tree breaks. Provisioning of multiple links among the nodes in an ad hoc wireless network results in a mesh-shaped structure. The mesh-based multicast routing structure may work well in a high-mobility environment. The major issues in designing multicast routing protocols are as follows:

• **Robustness:** The multicast routing protocol must be able to recover and reconfigure quickly from potential mobility-induced link breaks thus making it suitable for use in highly dynamic environments.

• **Efficiency:** A multicast protocol should make a minimum number of transmissions to deliver a data packet to all the group members.

• **Control overhead:** The scarce bandwidth availability in ad hoc wireless networks demands minimal control overhead for the multicast session.

• **Quality of service:** QoS support is essential in multicast routing because, in most cases, the data transferred in a multicast session is time-sensitive.

• **Efficient group management:** Group management refers to the process of accepting multicast session members and maintaining the connectivity among them until the session expires. This process of group management needs to be performed with minimal exchange of control messages.

• **Scalability:** The multicast routing protocol should be able to scale for a network with a large number of nodes.

• **Security:** Authentication of session members and prevention of non-members from gaining unauthorized information play a major role in military communications.

2.2.4 Transport Layer Protocols The main objectives of the transport layer protocols include setting up and maintaining end-to-end connections, reliable end-to-end delivery of data packets, flow control, and congestion control. There exist simple connectionless transport layer protocols (*e.g.,* UDP) which neither perform flow control and congestion control nor provide reliable data transfer. Such unreliable connectionless transport layer protocols do not take into account the current network status such as congestion at the intermediate links, the rate of collision, or other similar factors affecting the network throughput. This behaviour of the transport layer protocols increases the contention of the already-choked wireless links. For example, in an ad hoc wireless network that employs a contention-based MAC protocol, nodes in a high-contention region experience several backoff states, resulting in an increased number of collisions and a high latency. Connectionless transport layer protocols, unaware of this situation, increase the load in the network, degrading the network performance. The major performance degradation faced by a reliable connection-oriented transport layer protocol such as transmission control protocol (TCP) in an ad hoc wireless network arises due to frequent path breaks, presence of stale routing information, high channel error rate, and frequent network partitions. Further discussion of each of the above properties and their effect on the performance of the transport layer protocol assumes TCP as the transport layer protocol. Due to the mobility of nodes and limited transmission range, an existing path to a destination node experiences frequent path breaks. Each path break results in route reconfiguration that depends on the routing protocol employed. The process of finding an alternate path or reconfiguring the broken path might take longer than the retransmission timeout of the transport layer at the sender, resulting in retransmission of packets and execution of the congestion control algorithm. The congestion control algorithm decreases the size of the congestion window, resulting in low throughput. In an environment where path breaks are frequent, the execution of congestion control algorithms on every path break affects the throughput drastically. The latency associated with the reconfiguration of a broken path and the use of route caches result in stale route information at the nodes. Hence the packets will be forwarded through multiple paths to a destination, causing an increase in the number of out-of-order packets. Also, multipath routing protocols such as temporally-ordered routing algorithm (TORA) and split multipath routing (SMR) protocols (and) employ multiple paths between a sourcedestination pair. Out-of-order packet arrivals force the receiver of the TCPconnection to generate duplicate acknowledgments (ACKs). On receiving duplicate ACKs, the sender invokes the congestion control algorithm. Wireless channels are inherently unreliable due to the high probability of errors caused by interference. In addition to the error due to the channel noise, the presence of hidden terminals also contributes to the increased loss of TCP data packets or ACKs. When the TCPACK is delayed more than the round-trip timeout, the congestion control algorithm is invoked. Due to the mobility of the nodes, ad hoc wireless networks frequently experience isolation of nodes from the rest of the network or occurrence of partitions in the network. If a TCP connection spans across multiple partitions, that is, the sender and receiver of the connection are in two different partitions, all the packets get dropped. This tends to be more serious when the partitions exist for a long duration, resulting in

multiple retransmissions of the TCP packets and subsequent increase in the retransmission timers. Such a behavior causes long periods of inactivity even when a transient partition in the network lasts for a short while. Adaptation of the existing transport layer protocols should attempt to handle the above issues for performing efficiently in ad hoc wireless networks.

2.2.5 Pricing Scheme

An ad hoc wireless network's functioning depends on the presence of relaying nodes and their willingness to relay other nodes' traffic. Even if the node density is sufficient enough to ensure a fully connected network, a relaying neighbor node may not be interested in relaying a call and may just decide to power down. Assume that an optimal route from node A to node B passes through node C, and node C is not powered on. Then node A will have to set up a costlier and non-optimal route to B. The non-optimal path consumes more resources and affects the throughput of the system. As the intermediate nodes in a path that relay the data packets expend their resources such as battery charge and computing power, they should be properly compensated. Hence pricing schemes that incorporate service compensation or service reimbursement are required. Ad hoc wireless networks employed for special tasks such as military missions, rescue operations, and law enforcement do not require such pricing schemes, whereas the successful commercial deployment of ad hoc wireless networks requires billing and pricing. The obvious solution to provide participation guarantee is to provide incentives to forwarding nodes.

2.2.6 Quality of Service Provisioning

Quality of service (QoS) is the performance level of services offered by a service provider or a network to the user. QoS provisioning often requires negotiation between the host and the network, resource reservation schemes, priority scheduling, and call admission control. Rendering QoS in ad hoc wireless networks can be on a per flow, per link, or per node basis. In ad hoc wireless networks, the boundary between the service provider (network) and the user (host) is blurred, thus making it essential to have better coordination among the hosts to achieve QoS. The lack of central coordination and limited resources exacerbate the problem. In this section, a brief discussion of QoS parameters, QoS-aware routing, and QoS frameworks in ad hoc wireless networks is provided.

• **QoS parameters:** As different applications have different requirements, their level of QoS and the associated QoS parameters also differ from application to application. For example, for multimedia applications, the bandwidth and delay are the key parameters, whereas military applications have the additional requirements of security and reliability. For defense applications, finding trustworthy intermediate hosts and routing through them can be a QoS parameter. For applications such as emergency search-and-rescue operations, availability is the key QoS parameter. Multiple link disjoint paths can be the major requirement for such applications. Applications for hybrid wireless networks can have maximum available link life,

delay, channel utilization, and bandwidth as the key parameters for QoS. Finally, applications such as communication among the nodes in a sensor network require that the transmission among them results in minimum energy consumption, hence battery life and energy conservation can be the prime QoS parameters here.

• **QoS-aware routing:** The first step toward a QoS-aware routing protocol is to have the routing use QoS parameters for finding a path. The parameters that can be considered for routing decisions are network throughput, packet delivery ratio, reliability, delay, delay jitter, packet loss rate, bit error rate, and path loss. Decisions on the level of QoS and the related parameters for such services in ad hoc wireless networks are application-specific and are to be met by the underlying network. For example, in the case where the QoS parameter is bandwidth, the routing protocol utilizes the available bandwidth at every link to select a path with necessary bandwidth. This also demands the capability to reserve the required amount of bandwidth for that particular connection.

• **QoS framework:** A framework for QoS is a complete system that attempts to provide the promised services to each user or application. All the components within this subsystem should cooperate in providing the required services. The key component of QoS framework is a QoS service model which defines the way user requirements are served. The key design issue is whether to serve the user on a per-session basis or a per-class basis. Each class represents an aggregation of users based on certain criteria. The other key components of this framework are QoS routing to find all or some feasible paths in the network that can satisfy user requirements, QoS signaling for resource reservation required by the user or application, QoS medium access control, connection admission control, and scheduling schemes pertaining to that service model. The QoS modules such as routing protocol, signaling protocol, and resource management should react promptly according to changes in the network state (topology change in ad hoc wireless networks) and flow state (change in end-to-end view of service delivered).

2.2.7 Self-Organization

One very important property that an ad hoc wireless network should exhibit is organizing and maintaining the network by itself. The major activities that an ad hoc wireless network is required to perform for self-organization are neighbor discovery, topology organization, and topology reorganization. During the neighbor discovery phase, every node in the network gathers information about its neighbors and maintains that information in appropriate data structures. This may require periodic transmission of short packets named *beacons,* or promiscuous snooping on the channel for detecting activities of neighbors. Certain MAC protocols permit varying the transmission power to improve upon spectrum reusability. In the topology organization phase, every node in the network gathers information about the entire network or a part of the network in order to maintain topological information. During the topology reorganization phase, the ad hoc wireless networks require updating the topology information by incorporating the topological changes occurred in the network due to the mobility of nodes, failure of nodes, or complete depletion of power sources of the nodes. The

reorganization consists of two major activities. First is the periodic or a periodic exchange of topological information. Second is the adaptability (recovery from major topological changes in the network). Similarly, network partitioning and merging of two existing partitions require major topological reorganization. Ad hoc wireless networks should be able to perform self-organization quickly and efficiently in a way transparent to the user and the application.

2.2.8 Security The security of communication in ad hoc wireless networks is very important, especially in military applications. The lack of any central coordination and shared wireless medium makes them more vulnerable to attacks than wired networks. The attacks against ad hoc wireless networks are generally classified into two types: passive and active attacks. Passive attacks refer to the attempts made by malicious nodes to perceive the nature of activities and to obtain information transacted in the network without disrupting the operation. Active attacks disrupt the operation of the network. Those active attacks that are executed by nodes outside the network are called external attacks, and those that are performed by nodes belonging to the same network are called internal attacks. Nodes that perform internal attacks are compromised nodes. The major security threats that exist in ad hoc wireless networks are as follows:

• **Denial of service:** The attack effected by making the network resource unavailable for service to other nodes, either by consuming the bandwidth or by overloading the system, is known as denial of service (DoS). A simple scenario in which a DoS attack interrupts the operation of ad hoc wireless networks is by keeping a target node busy by making it process unnecessary packets.

• **Resource consumption:** The scarce availability of resources in ad hoc wireless network makes it an easy target for internal attacks, particularly aiming at consuming resources available in the network. The major types of resource-consumption attacks are the following: – **Energy depletion:** Since the nodes in ad hoc wireless networks are highly constrained by the energy source, this type of attack is basically aimed at depleting the battery power of critical nodes by directing unnecessary traffic through them.

– **Buffer overflow:** The buffer overflow attack is carried out either by filling the routing table with unwanted routing entries or by consuming the data packet buffer space with unwanted data. Such attacks can lead to a large number of data packets being dropped, leading to the loss of critical information. Routing table attacks can lead to many problems, such as preventing a node from updating route information for important destinations and filling the routing table with routes for nonexistent destinations.

• **Host impersonation:** A compromised internal node can act as another node and respond with appropriate control packets to create wrong route entries, and can terminate the traffic meant for the intended destination node.

• **Information disclosure:** A compromised node can act as an informer by deliberate disclosure of confidential information to unauthorized nodes. Information such as the amount and the periodicity of traffic between a selected pair of nodes and pattern of traffic changes can be very valuable for military applications. The use of filler traffic (traffic generated for the sole purpose of changing the traffic pattern) may not be suitable in resource-constrained ad hoc wireless networks.

• **Interference:** A common attack in defense applications is to jam the wireless communication by creating a wide-spectrum noise. This can be done by using a single wide-band jammer, sweeping across the spectrum. The MAC and the physical layer technologies should be able to handle such external threats.

2.2.9 Addressing and Service Discovery

Addressing and service discovery assume significance in ad hoc wireless networks due to the absence of any centralized coordinator. An address that is globally unique in the connected part of the ad hoc wireless network is required for a node in order to participate in communication. Auto-configuration of addresses is required to allocate non-duplicate addresses to the nodes. In networks where the topology is highly dynamic, frequent partitioning and merging of network components require duplicate address-detection mechanisms in order to maintain unique addressing throughout the connected parts of the network. Nodes in the network should be able to locate services that other nodes provide. Hence efficient service advertisement mechanisms are necessary. Topological changes force a change in the location of the service provider as well, hence fixed positioning of a server providing a particular service is ruled out. Rather, identifying the current location of the service provider gathers importance. The integration of service discovery with the route-acquisition mechanism, though it violates the traditional design objectives of the routing protocol, is a viable alternative. However, provisioning of certain kinds of services demands authentication, billing, and privacy that in turn require the service discovery protocols to be separated from the network layer protocols.

2.2.10 Energy Management Energy management is defined as the process of managing the sources and consumers of energy in a node or in the network as a whole for enhancing the lifetime of the network. Shaping the energy discharge pattern of a node's battery to enhance the battery life; finding routes that result in minimum total energy consumption in the network; using distributed scheduling schemes to improve battery life; and handling the processor and interface devices to minimize power consumption are some of the functions of energy management. Energy management can be classified into the following categories:

• **Transmission power management:** The power consumed by the radio frequency (RF) module of a mobile node is determined by several factors such as the state of operation, the transmission power, and the technology used for the RF circuitry. The state of operation refers to the transmit, receive, and sleep modes of the operation. The transmission power is determined by the

reachability requirement of the network, the routing protocol, and the MAC protocol employed. The RF hardware design should ensure minimum power consumption in all the three states of operation. Going to the sleep mode when not transmitting or receiving can be done by additional hardware that can wake up on reception of a control signal. Power conservation responsibility lies across the data link, network, transport, and application layers. By designing a data link layer protocol that reduces unnecessary retransmissions, by preventing collisions, by switching to standby mode or sleep mode whenever possible, and by reducing the transmit/receive switching, power management can be performed at the data link layer. The use of a variable power MAC protocol can lead to several advantages that include energy saving at the nodes, increase in bandwidth reuse, and reduction in interference. Also, MAC protocols for directional antennas are at their infancy. The network layer routing protocols can consider battery life and relaying load of the intermediate nodes while selecting a path so that the load can be balanced across the network, in addition to optimizing and reducing the size and frequency of control packets. At the transport layer, reducing the number of retransmissions, and recognizing and handling the reason behind the packet losses locally, can be incorporated into the protocols. At the application layer, the power consumption varies with applications. In a mobile computer, the image/video processing/playback software and 3D gaming software consume higher power than other applications. Hence application software developed for mobile computers should take into account the aspect of power consumption as well.

• **Battery energy management:** The battery management is aimed at extending the battery life of a node by taking advantage of its chemical properties, discharge patterns, and by the selection of a battery from a set of batteries that is available for redundancy. Recent studies showed that pulsed discharge of a battery gives longer life than continuous discharge. Controlling the charging rate and discharging rate of the battery is important in avoiding early charging to the maximum charge or full discharge below the minimum threshold. This can be achieved by means of embedded charge controllers in the battery pack. Also, the protocols at the data link layer and network layer can be designed to make use of the discharge models. Monitoring of the battery for voltage levels, remaining capacity, and temperature so that proactive actions (such as incremental powering off of certain devices, or shutting down of the mobile node when the voltage crosses a threshold) can be taken is required.

• **Processor power management:** The clock speed and the number of instructions executed per unit time are some of the processor parameters that affect power consumption. The CPU can be put into different power saving modes during low processing load conditions. The CPU power can be completely turned off if the machine is idle for a long time. In such cases, interrupts can be used to turn on the CPU upon detection of user interaction or other events.

• **Devices power management:** Intelligent device management can reduce power consumption of a mobile node significantly. This can be done by the operating system (OS) by selectively powering down interface devices that are not used or by putting devices into different power

saving modes, depending on their usage. Advanced power management features built into the operating system and application softwares for managing devices effectively are required.

2.2.11 Scalability

Even though the number of nodes in an ad hoc wireless network does not grow in the same magnitude as today's Internet, the operation of a large number of nodes in the ad hoc mode is not far away. Traditional applications such as military, emergency operations, and crowd control may not lead to such a big ad hoc wireless network. Commercial deployments of ad hoc wireless networks that include wireless mesh networks show early trends for a widespread installation of ad hoc wireless networks for mainstream wireless communication. For example, the latency of path-finding involved with an on-demand routing protocol in a large ad hoc wireless network may be unacceptably high. Similarly, the periodic routing overhead involved in a table-driven routing protocol may consume a significant amount of bandwidth in such large networks. Also a large ad hoc wireless network cannot be expected to be formed by homogeneous nodes, raising issues such as widely varying resource capabilities across the nodes. A hierarchical topology-based system and addressing may be more suitable for large ad hoc wireless networks. Hybrid architectures that combine the multi-hop radio relaying in the presence of infrastructure may improve scalability.

2.2.12 Deployment Considerations

The deployment of ad hoc wireless networks involves actions different from those of wired networks. It requires a good amount of planning and estimation of future traffic growth over any link in the network. The time-consuming planning stage is followed by the actual deployment of the network. The cost and time required for laying copper cables or fiber cables make it difficult to reconfigure any partial deployment that has already been done. The deployment of a commercial ad hoc wireless network has the following benefits when compared to wired networks:

• **Low cost of deployment:** The use of multi-hop wireless relaying essentially eliminates the requirement of laying cables and maintenance in a commercial deployment of communication infrastructure. Hence the cost involved is much lower than that of wired networks.

• **Incremental deployment:** In commercial wireless WANs based on ad hoc wireless networks, deployment can be performed incrementally over geographical regions of the city. The deployed part of the network starts functioning immediately after the minimum configuration is done. For example, during the deployment process for covering a highway, whenever each radio relaying equipment is installed on the highway side, it can be commissioned.

• **Short deployment time:** Compared to wired networks, the deployment time is considerably less due to the absence of any wired links. Also, wiring a dense urban region is extremely difficult and time-consuming in addition to the inconvenience caused.

• **Reconfigurability:** The cost involved in reconfiguring a wired network covering a metropolitan area network (MAN) is very high compared to that of an ad hoc wireless network covering the same service area. Also, the incremental deployment of ad hoc wireless networks might demand changes in the topology of the fixed part (*e.g.*, the relaying devices fixed on lamp posts or rooftops) of the network at a later stage. The issues and solutions for deployment of ad hoc wireless networks vary with the type of applications and the environment in which the networks are to be deployed. The following are the major issues to be considered in deploying an ad hoc wireless network:

• **Scenario of deployment:** The scenario of deployment assumes significance because the capability required for a mobile node varies with the environment in which it is used. The capabilities required for the mobile nodes that form an ad hoc wireless network among a fleet of ships are not the same as those required for forming an ad hoc wireless network among a set of notebook computers at a conference. The following are some of the different scenarios in which the deployment issues vary widely.

– **Military deployment:** The military deployment of an ad hoc wireless network may be datacentric (*e.g.*, a wireless sensor network) or user-centric (*e.g.*, soldiers or armored vehicles carrying soldiers equipped with wireless communication devices). The data-centric networks handle a different pattern of data traffic and can be partially comprised of static nodes, whereas the user-centric network consists of highly mobile nodes with or without any support from any infrastructure (*e.g.*, military satellite constellations). The vehicle-mounted nodes have at their disposal better power sources and computational resources, whereas the hand-held devices are constrained by energy and computational resources. Thus the resource availability demands appropriate changes in the protocols employed. Also, the military environment requires secure communication. Routing should involve as few nodes as possible to avoid possible leakage of information. Flat addressing schemes are preferred to hierarchical addressing since the latter addressing requires paths to be set up through the hierarchy, and hence the chances of unreliable nodes forwarding the packets are high.

– **Emergency operations deployment:** This kind of application scenario demands a quick deployment of rescue personnel equipped with hand-held communication equipment. Essentially, the network should provide support for time-sensitive traffic such as voice and video. Short data messaging can be used in case the resource constraints do not permit voice communication. Also in this scenario, a flat fixed addressing scheme with a static configuration is preferred. Typically, the size of the network for such applications is not more than 100 nodes. The nodes are fully mobile without expecting support from any fixed infrastructure.

– **Commercial wide-area deployment:** One example of this deployment scenario is the wireless mesh networks. The aim of the deployment is to provide an alternate communication infrastructure for wireless communication in urban areas and areas where a traditional cellular base station cannot handle the traffic volume. This scenario assumes significance as it provides

very low cost per bit transferred compared to the wide-area cellular network infrastructure. Another major advantage of this application is the resilience to failure of a certain number of nodes. Addressing, configuration, positioning of relaying nodes, redundancy of nodes, and power sources are the major issues in deployment. Billing, provisioning of QoS, security, and handling mobility are major issues that the service providers need to address.

– **Home network deployment:** The deployment of a home area network needs to consider the limited range of the devices that are to be connected by the network. Given the short transmission ranges of a few meters, it is essential to avoid network partitions. Positioning of relay nodes at certain key locations of a home area network can solve this. Also, network topology should be decided so that every node is connected through multiple neighbors for availability.

• **Required longevity of network:** The deployment of ad hoc wireless networks should also consider the required longevity of the network. If the network is required for a short while (*e.g.*, the connectivity among a set of researchers at a conference and the connectivity required for coordination of a crowd control team), battery-powered mobile nodes can be used. When the connectivity is required for a longer duration of time, fixed radio relaying equipment with regenerative power sources can be deployed. A wireless mesh network with roof-top antennas deployed at a residential zone requires weather-proof packages so that the internal circuitry remains unaffected by the environmental conditions. In such an environment, the mesh connectivity is planned in such a way that the harsh atmospheric factors do not create network partitions.

• **Area of coverage:** In most cases, the area of coverage of an ad hoc wireless network is determined by the nature of application for which the network is set up. For example, the home area network is limited to the surroundings of a home. The mobile nodes' capabilities such as the transmission range and associated hardware, software, and power source should match the area of coverage required. In some cases where some nodes can be fixed and the network topology is partially or fully fixed, the coverage can be enhanced by means of directional antennas.

• **Service availability:** The availability of network service is defined as the ability of an ad hoc wireless network to provide service even with the failure of certain nodes. Availability assumes significance both in a fully mobile ad hoc wireless network used for tactical communication and in partially fixed ad hoc wireless networks used in commercial communication infrastructure such as wireless mesh networks. In the case of wireless mesh networks, the fixed nodes need to be placed in such a way that the failure of multiple nodes does not lead to lack of service in that area. In such cases, redundant inactive radio relaying devices can be placed in such a way that on the event of failure of an active relaying node, the redundant relaying device can take over its responsibilities.

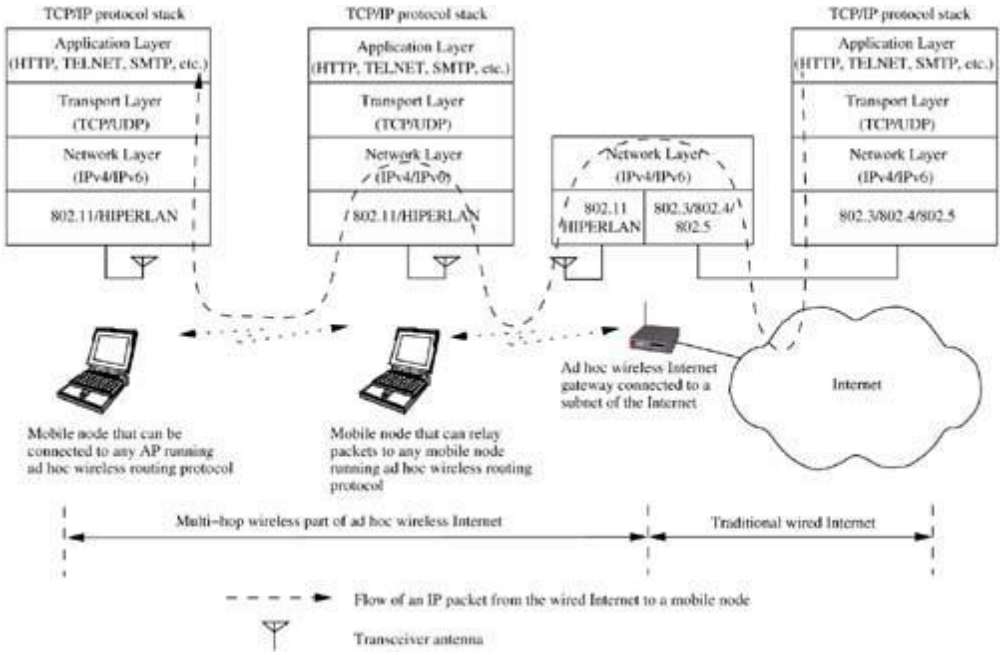
• **Operational integration with other infrastructure:** Operational integration of ad hoc wireless networks with other infrastructures can be considered for improving the performance or gathering additional information, or for providing better QoS. In the military environment, integration of ad hoc wireless networks with satellite networks or unmanned aerial vehicles (UAVs) improves the capability of the ad hoc wireless networks. Several routing protocols assume the availability of the global positioning system (GPS), which is a satellite-based infrastructure by which the geographical location information can be obtained as a resource for network synchronization and geographical positioning. In the commercial world, the wireless mesh networks that service a given urban region can interoperate with the wide-area cellular infrastructure in order to provide better QoS and smooth handoffs across the networks. Handoff to a different network can be done in order to avoid call drops when a mobile node with an active call moves into a region where service is not provided by the current network.

• **Choice of protocols:** The choice of protocols at different layers of the protocol stack is to be done taking into consideration the deployment scenario. A TDMA-based and in secure MAC protocol may not be the best suited compared to a CDMA-based MAC protocol for a military application. The MAC protocol should ensure provisioning of security at the link level. At the network layer, the routing protocol has to be selected with care. A routing protocol that uses geographical information (GPS information) may not work well in situations where such information is not available. For example, the search-and-rescue operation teams that work in extreme terrains or underground or inside a building may not be able to use such a routing protocol. An ad hoc wireless network with nodes that cannot have their power sources replenished should use a routing protocol that does not employ periodic *bea-cons* for routing. The periodic beacons, or routing updates, drain the battery with time. In situations of high mobility, for example, an ad hoc wireless network formed by devices connected to military vehicles, the power consumption may not be very important and hence one can employ beacon based routing protocols for them. The updated information about connectivity leads to improved performance. In the case of deployment of wireless mesh networks, the protocols should make use of the fixed nodes to avoid unstable paths due to the mobility of the relaying nodes. At the transport layer, the connection-oriented or connectionless protocol should be adapted to work in the environment in which the ad hoc wireless network is deployed. In a high-mobility environment, path breaks, network partitions, and remerging of partitions are to be considered, and appropriate actions should be taken at the higher layers. This can be extended to connectionless transport protocols to avoid congestion. Also, packet loss arising out of congestion, channel error, link break, and network partition is to be handled differently in different applications. The timer values at different layers of the protocol stack should be adapted to the deployment scenarios.

2.3 AD HOC WIRELESS INTERNET

Similar to the wireless Internet discussed in Chapter 4, the ad hoc wireless Internet extends the services of the Internet to the end users over an ad hoc wireless network. Some of the applications of the ad hoc wireless Internet are wireless mesh networks, provisioning of temporary Internet services to major conference venues, sports venues, temporary military settlements, battlefields, and broadband Internet services in rural regions. A schematic diagram of the ad hoc wireless Internet is shown in Figure 2.7.

Figure 2.7. A schematic diagram of the ad hoc wireless Internet.



The major issues to be considered for a successful ad hoc wireless Internet are the following: • **Gateways:** Gateway nodes in the ad hoc wireless Internet are the entry points to the wired Internet. The major part of the service provisioning lies with the gateway nodes. Generally owned and operated by a service provider, gateways perform the following tasks: keeping track of the end users, bandwidth management, load balancing, traffic shaping, packet filtering, bandwidth fairness, and address, service, and location discovery.

• **Address mobility:** Similar to the Mobile IP, the ad hoc wireless Internet also faces the challenge of address mobility. This problem is worse here as the nodes operate over multiple wireless hops. Solutions such as Mobile IP can provide temporary alternatives for this.

• **Routing:** Routing is a major problem in the ad hoc wireless Internet, due to the dynamic topological changes, the presence of gateways, multi-hop relaying, and the hybrid character of the network. The possible solution for this is the use of a separate routing protocol, for the wireless part of the ad hoc wireless Internet. Routing protocols are more suitable as they exploit the presence of gateway nodes.

• **Transport layer protocol:** Even though several solutions for transport layer protocols exist for ad hoc wireless networks, unlike other layers, the choice lies in favor of TCP's extensions proposed for ad hoc wireless networks. Split approaches that use traditional wired TCP for the wired part and a specialized transport layer protocol for the ad hoc wireless network part can also be considered where the gateways act as the intermediate nodes at which the connections are split. Several factors are to be considered here, the major one being the state maintenance overhead at the gateway nodes.

• **Load balancing:** It is likely that the ad hoc wireless Internet gateways experience heavy traffic. Hence the gateways can be saturated much earlier than other nodes in the network. Load balancing techniques are essential to distribute the load so as to avoid the situation where the gateway nodes become bottleneck nodes. Gateway selection strategies and load balancing schemes can be used for this purpose.

• **Pricing/billing:** Since Internet bandwidth is expensive, it becomes very important to introduce pricing/billing strategies for the ad hoc wireless Internet. Gateway is the preferred choice for charging the traffic to and from the Internet. Pricing schemes can be used for this purpose. A much more complex case is pricing the local traffic (traffic within the wireless part, that is, it originated and terminated within the wireless part without passing through the gateway nodes), where it becomes necessary to have a dedicated, secure, and lightweight pricing/billing infrastructure installed at every node.

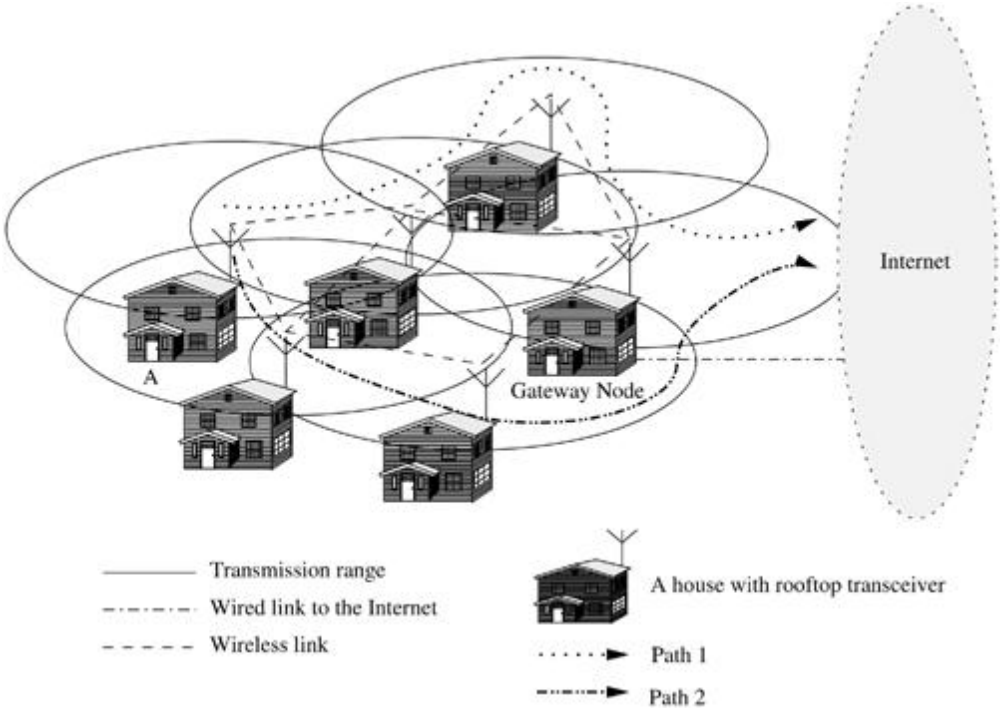
• **Provisioning of security:** The inherent broadcast nature of the wireless medium attracts not just the mobility seekers but also potential hackers who love to snoop on important information sent unprotected over the air. Hence security is a prime concern in the ad hoc wireless Internet. Since the end users can utilize the ad hoc wireless Internet infrastructure to make e-commerce transactions, it is important to include security mechanisms in the ad hoc wireless Internet.

• **QoS support:** With the widespread use of voice over IP (VoIP) and growing multimedia applications over the Internet, provisioning of QoS support in the ad hoc wireless Internet becomes a very important issue. As discussed in Chapter 10, this is a challenging problem in the wired part as well as in the wireless part.

• **Service, address, and location discovery:** Service discovery in any network refers to the activity of discovering or identifying the party which provides a particular service or resource. In wired networks, service location protocols exist to do the same, and similar systems need to be extended to operate in the ad hoc wireless Internet as well. Address discovery refers to the services such as those provided by address resolution protocol (ARP) or domain name service (DNS) operating within the wireless domain. Location discovery refers to different activities such as detecting the location of a particular mobile node in the network or detecting the geographical location of nodes. Location discovery services can provide enhanced services such as routing of packets, location-based services, and selective region-wide broadcasts. Figure 2.8

shows a wireless mesh network that connects several houses to the Internet through a gateway node. Such networks can provide highly reliable broadband wireless networks for the urban as well as the rural population in a cost-effective manner with fast deployment and reconfiguration. This wireless mesh network is a special case of the ad hoc wireless Internet where mobility of nodes is not a major concern as most relay stations and end users use fixed transceivers. Figure 2.8 shows that house A is connected to the Internet over multiple paths (path 1 and path 2).

Figure 2.8. An illustration of the ad hoc wireless Internet implemented by a wireless mesh network.



MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS

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

2.4 ISSUES IN DESIGNING A MAC PROTOCOL FOR AD HOC WIRELESS NETWORKS

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

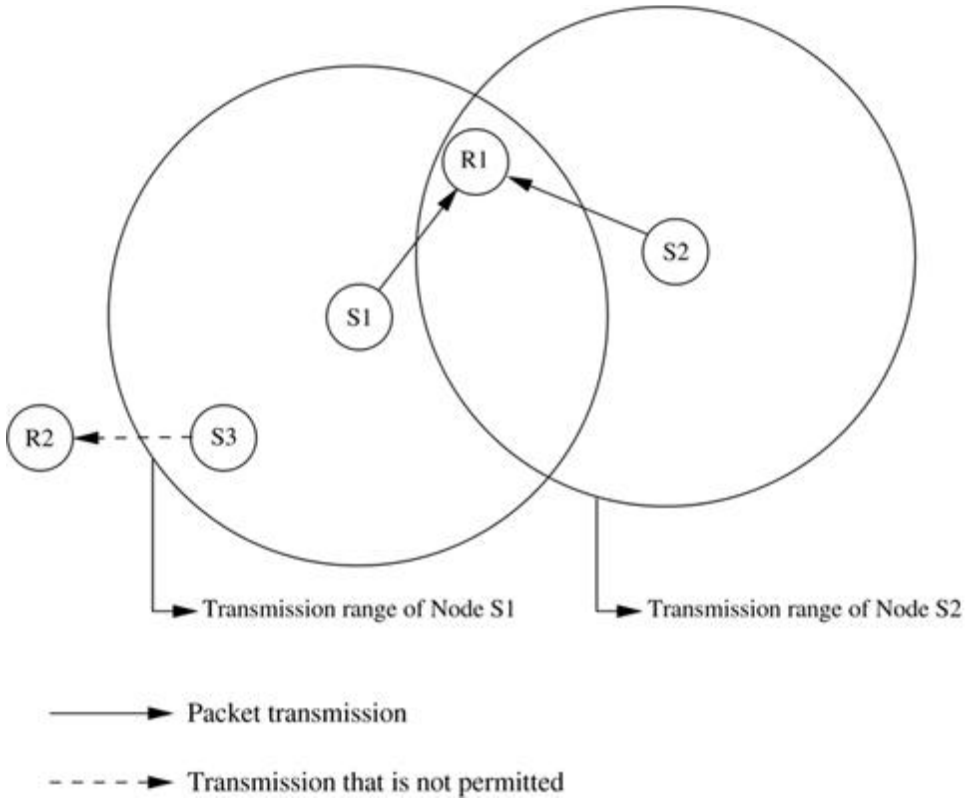
2.4.1 Bandwidth Efficiency As mentioned earlier, since the radio spectrum is limited, the bandwidth available for communication is also very limited. The MAC protocol must be designed in such a way that the scarce bandwidth is utilized in an efficient manner. The control overhead involved must be kept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

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

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

2.4.4 Hidden and Exposed Terminal Problems The hidden and exposed terminal problems are unique to wireless networks. The hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver. Collision occurs when both nodes transmit packets at the same time without knowing about the transmission of each other. For example, consider Figure 2.9. Here, if both node S1 and node S2 transmit to node R1 at the same time, their packets collide at node R1. This is because both nodes S1 and S2 are hidden from each other as they are not within the direct transmission range of each other and hence do not know about the presence of each other.

Figure 2.9. Hidden and exposed terminal problems.



The exposed terminal problem refers to the inability of a node, which is blocked due to transmission by a nearby transmitting node, to transmit to another node. Consider the example in Figure 2.9. Here, if a transmission from node S1 to another node R1 is already in progress, node S3 cannot transmit to node R2, as it concludes that its neighbor node S1 is in transmitting mode and hence it should not interfere with the on-going transmission. The hidden and exposed terminal problems significantly reduce the throughput of a network when the traffic load is high. It is therefore desirable that the MAC protocol be free from the hidden and exposed terminal problems.

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

2.4.6 Distributed Nature/Lack of Central Coordination Ad hoc wireless networks do not have centralized coordinators. In cellular networks, for example, the base stations act as

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

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

2.5 DESIGN GOALS OF A MAC PROTOCOL FOR AD HOC WIRELESS NETWORKS

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

• The operation of the protocol should be distributed.

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

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

• The available bandwidth must be utilized efficiently.

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

• Control overhead must be kept as low as possible.

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

• The protocol must be scalable to large networks.

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

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

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

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

2.6 CLASSIFICATIONS OF MAC PROTOCOLS

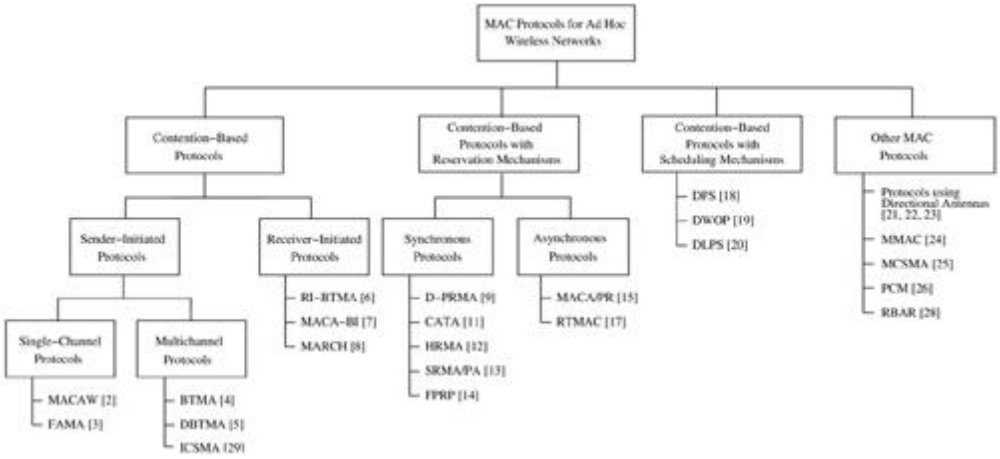
MAC protocols for ad hoc wireless networks can be classified into several categories based on various criteria such as initiation approach, time synchronization, and reservation approaches. Figure 2.10 provides a detailed classification tree. In this section, some of the classifications of MAC protocols are briefly discussed. Ad hoc network MAC protocols can be classified into three basic types:

• Contention-based protocols

• Contention-based protocols with reservation mechanisms

• Contention-based protocols with scheduling mechanisms

Figure 2.10. Classifications of MAC protocols.



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

2.6.1 Contention-Based Protocols

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

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

• Receiver-initiated protocols: The receiver node initiates the contention resolution protocol. Sender-initiated protocols can be further divided into two types:

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

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

2.6.2 Contention-Based Protocols with Reservation Mechanisms

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

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

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

2.6.3 Contention-Based Protocols with Scheduling Mechanisms

As mentioned earlier, these protocols focus on packet scheduling at nodes, and also scheduling nodes for access to the channel. Node scheduling is done in a manner so that all nodes are treated fairly and no node is starved of bandwidth. Scheduling-based schemes are also used for enforcing priorities among flows whose packets are queued at nodes. Some scheduling schemes also take into consideration battery characteristics, such as remaining battery power, while scheduling nodes for access to the channel.

2.6.4 Other Protocols There are several other MAC protocols that do not strictly fall under the above categories.

2.7 CONTENTION-BASED PROTOCOLS

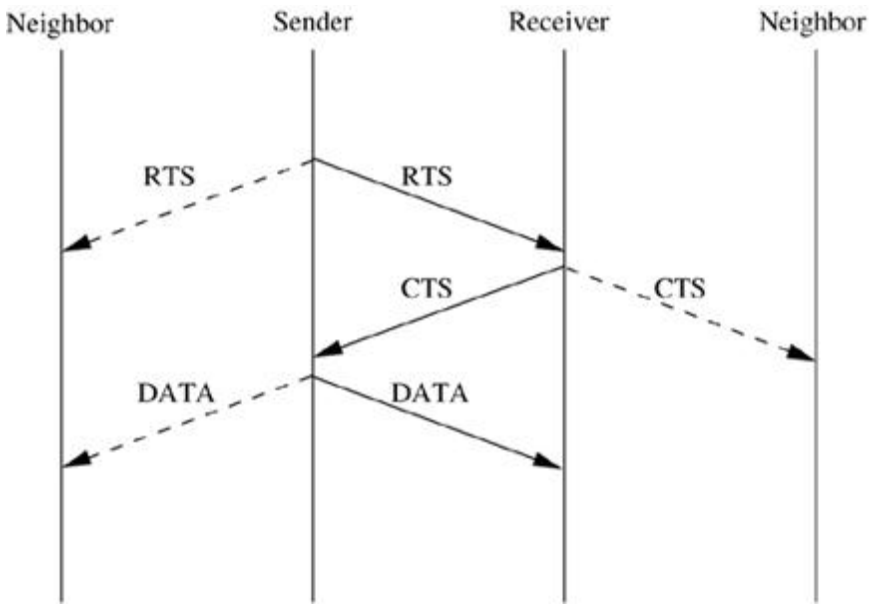
As mentioned earlier, contention-based protocols do not have any bandwidth reservation mechanisms. All ready nodes contend for the channel simultaneously, and the winning node gains access to the channel. Since nodes are

not guaranteed bandwidth, these protocols cannot be used for transmitting real-time traffic, which requires QoS guarantees from the system. In this section, several contention-based MAC protocols are described in detail.

2.7.1 MACAW: A Media Access Protocol for Wireless LANs

This protocol is based on the multiple access collision avoidance protocol (MACA) proposed by Karn. MACA was proposed due to the shortcomings of CSMA protocols when used for wireless networks. In what follows, a brief description on why CSMA protocols fail in wireless networks is given. This is followed by detailed descriptions of the MACA protocol and the MACAW protocol. MACA Protocol The MACA protocol was proposed as an alternative to the traditional carrier sense multiple access (CSMA) protocols used in wired networks. In CSMA protocols, the sender first senses the channel for the carrier signal. If the carrier is present, it retries after a random period of time. Otherwise, it transmits the packet. CSMA senses the state of the channel only at the transmitter. This protocol does not overcome the hidden terminal problem. In a typical ad hoc wireless network, the transmitter and receiver may not be near each other at all times. In such situations, the packets transmitted by a node are prone to collisions at the receiver due to simultaneous transmissions by the hidden terminals. Also, the bandwidth utilization in CSMA protocols is less because of the exposed terminal problem. MACA does not make use of carrier-sensing for channel access. It uses two additional signaling packets: the request-to-send (RTS) packet and the clear-to-send (CTS) packet. When a node wants to transmit a data packet, it first transmits an RTS packet. The receiver node, on receiving the RTS packet, if it is ready to receive the data packet, transmits a CTS packet. Once the sender receives the CTS packet without any error, it starts transmitting the data packet. This data transmission mechanism is depicted in Figure 2.11. If a packet transmitted by a node is lost, the node uses the binary exponential back-off (BEB) algorithm to back off for a random interval of time before retrying. In the binary exponential backoff mechanism, each time a collision is detected, the node doubles its maximum back-off window. Neighbor nodes near the sender that hear the RTS packet do not transmit for a long enough period of time so that the sender could receive the CTS packet. Both the RTS and the CTS packets carry the expected duration of the data packet transmission. A node near the receiver, upon hearing the CTS packet, defers its transmission till the receiver receives the data packet. Thus, MACA overcomes the hidden node problem. Similarly, a node receiving an RTS defers only for a short period of time till the sender could receive the CTS. If no CTS is heard by the node during its waiting period, it is free to transmit packets once the waiting interval is over. Thus, a node that hears only the RTS packet is free to transmit simultaneously when the sender of the RTS is transmitting data packets. Hence, the exposed terminal problem is also overcome in MACA. But MACA still has certain problems, which was why MACAW, described below, was proposed.

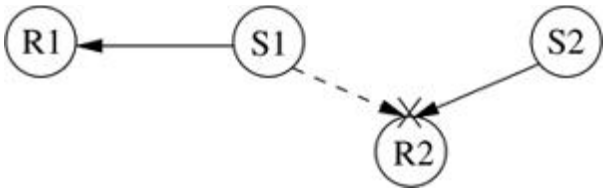
Figure 2.11. Packet transmission in MACA.



MACAW Protocol

The binary exponential back-off mechanism used in MACA at times starves flows. For example, consider Figure 2.6. Here both nodes S1 and S2 keep generating a high volume of traffic. The node that first captures the channel (say, node S1) starts transmitting packets. The packets transmitted by the other node S2 get collided, and the node keeps incrementing its back-off window according to the BEB algorithm. As a result, the probability of node S2 acquiring the channel keeps decreasing, and over a period of time it gets completely blocked. To overcome this problem, the back-off algorithm has been modified in MACAW . The packet header now has an additional field carrying the current back-off counter value of the transmitting node. A node receiving the packet copies this value into its own back-off counter. This mechanism allocates bandwidth in a fair manner. Another problem with BEB algorithm is that it adjusts the back-off counter value very rapidly, both when a node successfully transmits a packet and when a collision is detected by the node. The back-off counter is reset to the minimum value after every successful transmission. In the modified back-off process, this would require a period of contention to be repeated after each successful transmission in order to build up the back-off timer values. To prevent such large variations in the back-off values, a multiplicative increase and linear decrease (MILD) back-off mechanism is used in MACAW. Here, upon a collision, the back-off is increased by a multiplicative factor (1.5), and upon a successful transmission, it is decremented by one. This eliminates contention and hence long contention periods after every successful transmission, at the same time providing a reasonably quick escalation in the back-off values when the contention is high.

Figure 2.6. Example topology.



In MACAW another modification related to the back-off mechanism has been made. MACAW implements per flow fairness as opposed to the per node fairness in MACA. This is done by maintaining multiple queues at every node, one each for each data stream, and running the backoff algorithm independently for each queue. A node that is ready to transmit packets first determines how long it needs to wait before it could transmit an RTS packet to each of the destination nodes corresponding to the top-most packets in the node's queues. It then selects the packet for which the waiting time is minimal. In addition to the RTS and CTS control packets used in MACA, MACAW uses another new control packet called acknowledgment (ACK) packet. The need for using this additional packet arises because of the following reason. In MACA, the responsibility of recovering from transmission errors lies with the transport layer. As many TCP implementations have a minimum timeout period of about 0.5 sec, significant delay is involved while recovering from errors. But in MACAW, the error recovery responsibility is given to the data link layer (DLL). In DLL, the recovery process can be made quicker as the timeout periods can be modified in order to suit the physical media being employed. In MACAW, after successful reception of each data packet, the receiver node transmits an ACK packet. If the sender does not receive the ACK packet, it reschedules the same data packet for transmission. The back-off counter is incremented if the ACK packet is not received by the sender. If the ACK packet got lost in transmission, the sender would retry by transmitting an RTS for the same packet. But now the receiver, instead of sending back a CTS, sends an ACK for the packet received, and the sender moves on to transmit the next data packet. In MACA, an exposed node (which received only the RTS and not the CTS packet) is free to transmit simultaneously when the source node is transmitting packets. For example, in Figure 2.7, when a transmission is going on between nodes S1 and R1, node S2 is free to transmit. RTS transmissions by node S2 are of no use, as it can proceed further only if it can receive a CTS from R2. But this is not possible as CTS packets from R2 get collided at node S2 with packets transmitted by node S1. As a result, the back-off counter at node S2 builds up unnecessarily. So an exposed node should not be allowed to transmit. But an exposed node, since it can hear only the RTS sent by the source node and not the CTS sent by the receiver, does not know for sure whether the RTS-CTS exchange was successful. To overcome this problem, MACAW uses another small (30-bytes) control packet called the data-sending (DS) packet. Before transmitting the actual data packet, the source node transmits this DS packet. The DS packet carries information such as the duration of the data packet transmission, which could be used by the exposed nodes for updating information they hold regarding the duration of the data packet transmission. An exposed node, overhearing the DS packet, understands that the previous RTS-CTS exchange was successful, and so defers its transmissions until the expected duration of the

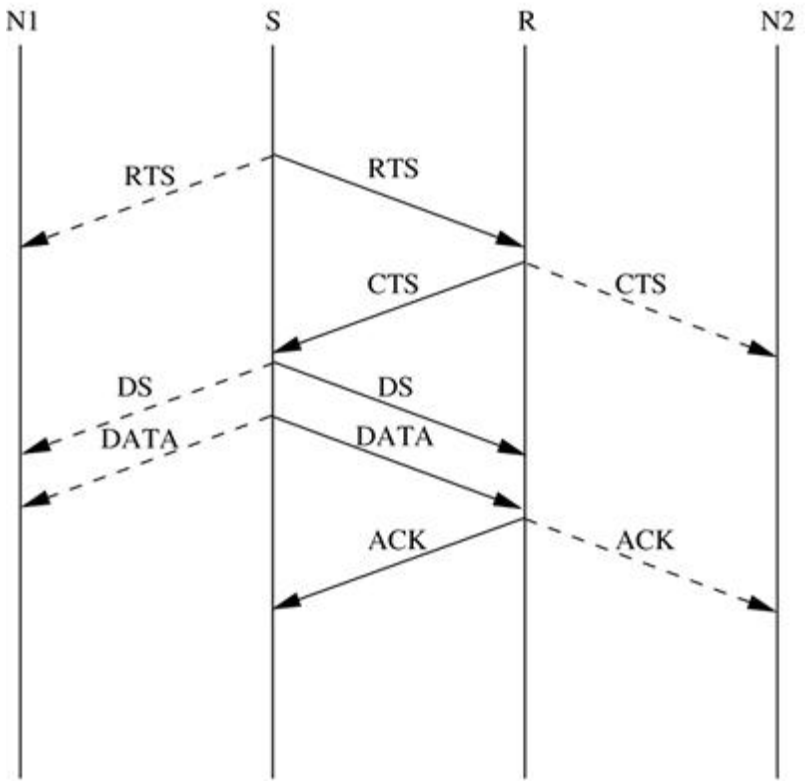
DATA-ACK exchange. If the DS packet was not used, the exposed node (node S2) would retransmit after waiting for random intervals of time, and with a high probability the data transmission (between nodes S1 and R1) would be still going on when the exposed node retransmits. This would result in a collision and the back-off period being further incremented, which affects the node even more.

Figure 2.7. Example topology.

The MACAW protocol uses one more control packet called the request-for-request-to-send (RRTS) packet. The following example shows how this RRTS packet proves to be useful. Consider Figure 2.8. Here assume transmission is going on between nodes S1 and R1. Now node S2 wants to transmit to node R2. But since R2 is a neighbor of R1, it receives CTSpackets from node R1, and therefore it defers its own transmissions. Node S2 has no way to learn about the contention periods during which it can contend for the channel, and so it keeps on trying, incrementing its back-off counter after each failed attempt. Hence the main reason for this problem is the lack of synchronization information at source S2. MACAW overcomes this problem by using the RRTS packet. In the same example shown in Figure 2.8, receiver node R2 contends for the channel on behalf of source S2. If node R2 had received anRTS previously for which it was not able to respond immediately because of the on-going transmission between nodes S1 and R1, then node R2 waits for the next contention period and transmits the RRTS packet. Neighbor nodes that hear the RRTS packet (including node R1) are made to wait for two successive slots (for the RTS-CTS exchange to take place). The source node S2, on receiving the RRTS from node R2, transmits the regular RTS packet to node R2, and the normal packet exchange (RTS-CTS-Data-ACK) continues from here. Figure 2.9 shows the operation of the MACAW protocol. In the figure, S is the source node and R denotes the receiver node. N1 and N2 are neighbor nodes. When RTS transmitted by node S is overheard by node N1, it refrains from transmitting until node S receives the CTS. Similarly, when the CTS transmitted by node R is heard by neighbor node N2, it defers its transmissions until the data packet is received by receiver R. On receiving this CTS packet, node S immediately transmits the DS message carrying the expected duration of the data packet transmission. On hearing this packet, node N1 backs off until the data packet is transmitted. Finally, after receiving the data packet, node R acknowledges the reception by sending node S an ACK packet.

Figure 2.8. Example topology.

Figure 2.9. Packet exchange in MACAW.



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

2.7.2 Floor Acquisition Multiple Access Protocols

The floor acquisition multiple access (FAMA) protocols are based on a channel access discipline which consists of a carrier-sensing operation and a collision-avoidance dialog between the sender and the intended receiver of a packet. Floor acquisition refers to the process of gaining control of the channel. At any given point of time, the control of the channel is assigned to only one node, and this node is guaranteed to transmit one or more data packets to different

destinations without suffering from packet collisions. Carrier-sensing by the sender, followed by the RTS-CTS control packet exchange, enables the protocol to perform as efficiently as MACA in the presence of hidden terminals, and as efficiently as CSMA otherwise. FAMA requires a node that wishes to transmit packets to first acquire the floor (channel) before starting to transmit the packets. The floor is acquired by means of exchanging control packets. Though the control packets themselves may collide with other control packets, it is ensured that data packets sent by the node that has acquired the channel are always transmitted without any collisions. Any single-channel MAC protocol that does not require a transmitting node to sense the channel can be adapted for performing floor acquisition tasks. Floor acquisition using the RTS-CTS exchange is advantageous as the mechanism also tries to provide a solution for the hidden terminal problem. Two FAMA protocol variants are discussed in this section: RTS-CTS exchange with no carrier sensing, and RTS-CTS exchange with non-persistent carrier-sensing. The first variant uses the ALOHA protocol for transmitting RTS packets, while the second variant uses non persistent CSMA for the same purpose.

Multiple Access Collision Avoidance

Multiple access collision avoidance (MACA) , which was discussed earlier in this chapter, belongs to the category of FAMA protocols. In MACA, a ready node transmits an RTS packet. A neighbor node receiving the RTS defers its transmissions for the period specified in the RTS. On receiving the RTS, the receiver node responds by sending back a CTS packet, and waits for a long enough period of time in order to receive a data packet. Neighbor nodes of the receiver which hear this CTS packet defer their transmissions for the time duration of the impending data transfer. In MACA, nodes do not sense the channel. A node defers its transmissions only if it receives an RTS or CTS packet. In MACA, data packets are prone to collisions with RTS packets.

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

FAMA – Non-Persistent Transmit Request

This variant of FAMA, called FAMA – non-persistent transmit request (FAMA-NTR), combines non-persistent carrier-sensing along with the RTS-CTS control packet exchange mechanism. Before sending a packet, the sender node senses the channel. If the channel is found to be busy, then the node backs off for a random time period and retries later. If the channel is found to be free, it transmits the RTS packet. After transmitting the RTS, the sender listens to the channel for one round-trip time in addition to the time required by the receiver node to transmit a CTS. If it does not receive the CTS within this time period or if the CTS received is found to be corrupted,

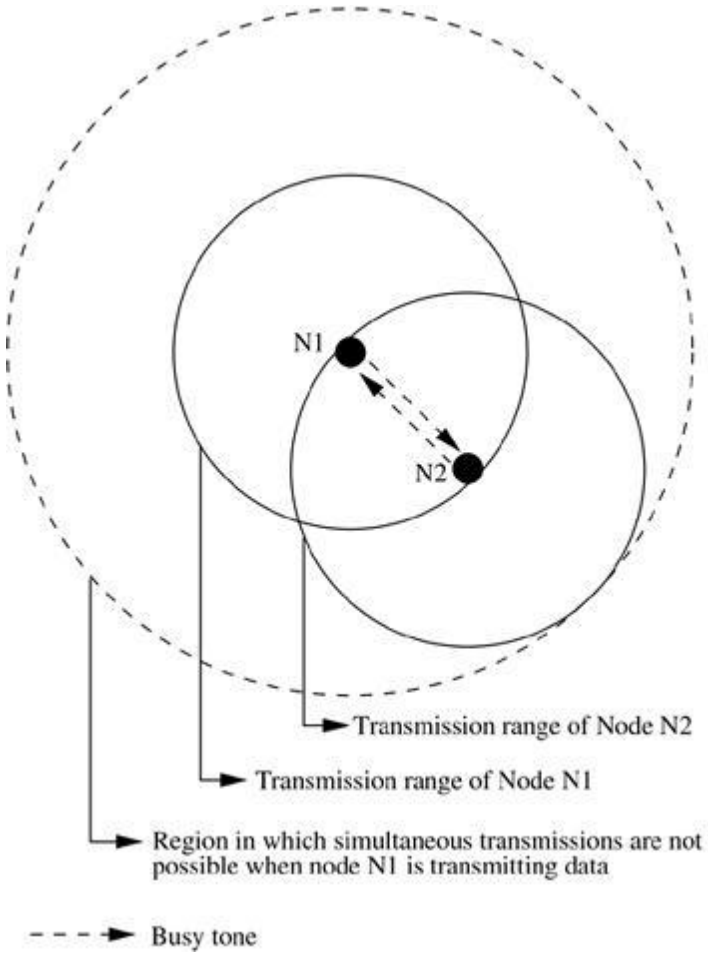
then the node takes a random back-off and retries later. Once the sender node receives the CTS packet without any error, it can start transmitting its data packet burst. The burst is limited to a maximum number of data packets, after which the node releases the channel, and contends with other nodes to again acquire the channel. In order to allow the sender node to send a burst of packets once it acquires the floor, the receiver node is made to wait for a time duration of τ seconds after processing each data packet received. Here, τ denotes the maximum channel propagation time. A waiting period of 2τ seconds is enforced on a transmitting node after transmitting any control packet. This is done to allow the RTS-CTS exchange to take place without any error. A node transmitting an RTS is required to wait for 2τ seconds after transmitting the RTS in order to enable the receiver node to receive the RTS and transmit the corresponding CTS packet. After sending the final data packet, a sender node is made to wait for τ seconds in order to allow the destination to receive the data packet and to account for the enforced waiting time at the destination node.

2.7.3 Busy Tone Multiple Access Protocols

Busy Tone Multiple Access

The busy tone multiple access (BTMA) protocol is one of the earliest protocols proposed for overcoming the hidden terminal problem faced in wireless environments. The transmission channel is split into two: a data channel and a control channel. The data channel is used for data packet transmissions, while the control channel is used to transmit the busy tone signal. When a node is ready for transmission, it senses the channel to check whether the busy tone is active. If not, it turns on the busy tone signal and starts data transmission; otherwise, it reschedules the packet for transmission after some random rescheduling delay. Any other node which senses the carrier on the incoming data channel also transmits the busy tone signal on the control channel. Thus, when a node is transmitting, no other node in the two-hop neighborhood of the transmitting node is permitted to simultaneously transmit. Though the probability of collisions is very low in BTMA, the bandwidth utilization is very poor.Figure 2.10 shows the worst-case scenario where the node density is very high; the dotted circle shows the region in which nodes are blocked from simultaneously transmitting when node N1 is transmitting packets.

Figure 2.10. Transmission in BTMA.

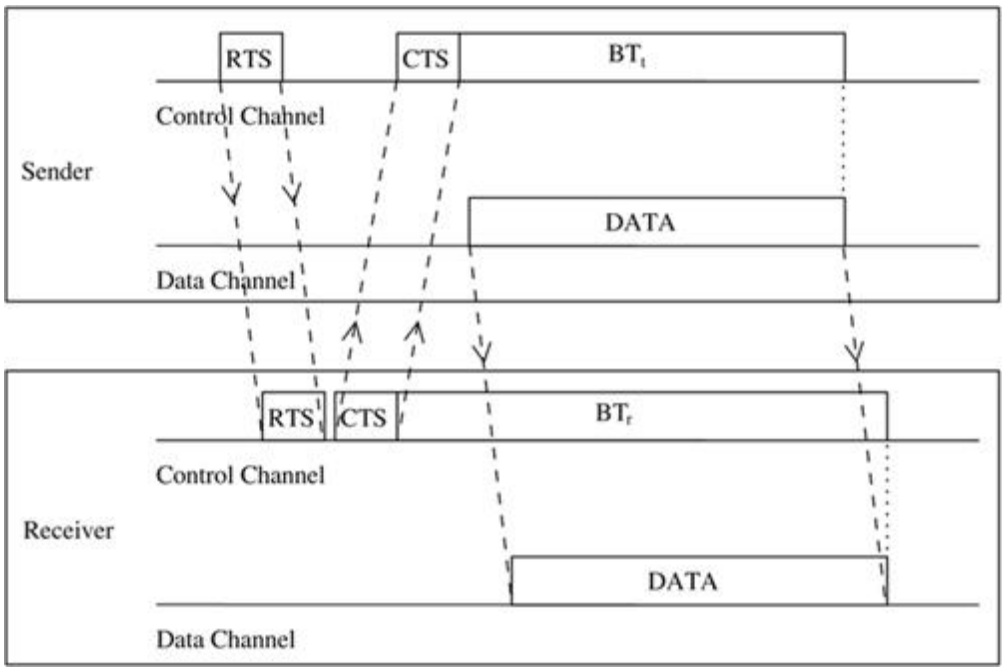


Dual Busy Tone Multiple Access Protocol

The dual busy tone multiple access protocol (DBTMA) is an extension of the BTMA scheme. Here again, the transmission channel is divided into two: the data channel and the control channel. As in BTMA, the data channel is used for data packet transmissions. The control channel is used for control packet transmissions (RTS and CTS packets) and also for transmitting the busy tones. DBTMA uses two busy tones on the control channel, BT_t and BT_r. The BT_t tone is used by the node to indicate that it is transmitting on the data channel. The BT_r tone is turned on by a node when it is receiving data on the data channel. The two busy tone signals are two sine waves at different well-separated frequencies. When a node is ready to transmit a data packet, it first senses the channel to determine whether the BT_r signal is active. An active BT_r signal indicates that a node in the neighborhood of the ready node is currently receiving packets. If the ready node finds that there is no BT_r signal, it transmits the RTS packet on the control channel. On receiving theRTS packets, the node to which the RTS was destined checks whether the BT_t tone is active in its neighborhood. An active BT_t implies that some other node in its neighborhood is transmitting packets and so it cannot receive packets for the moment. If the node finds no BT_t signal, it responds by sending a CTS packet and then turns on the BT_r signal (which informs other nodes in its neighborhood that it is receiving). The sender node, on receiving this CTS packet,

turns on the BT_t signal (to inform nodes in its neighborhood that it is transmitting) and starts transmitting data packets. After completing transmission, the sender node turns off the BT_t signal. The receiver node, after receiving all data packets, turns off the BT_r signal. The above process is depicted in Figure 2.11.

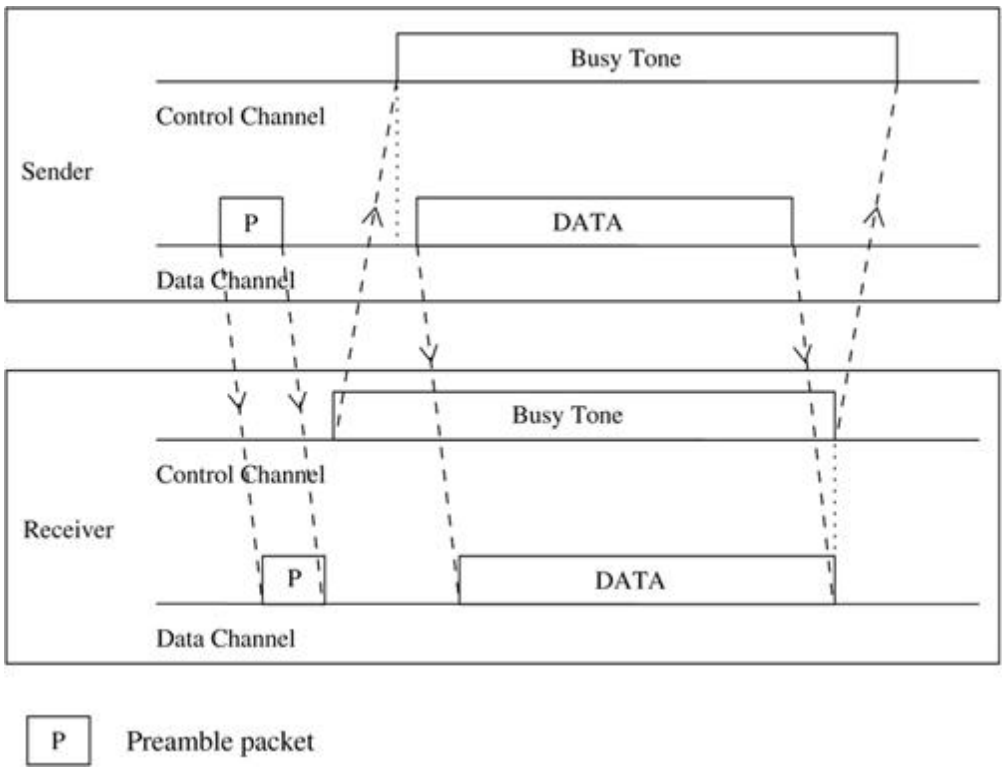
Figure 2.11. Packet transmission in DBTMA.



When compared to other RTS/CTS-based medium access control schemes (such as MACA and MACAW), DBTMA exhibits better network utilization. This is because the other schemes block both the forward and reverse transmissions on the data channel when they reserve the channel through their RTS or CTS packets. But in DBTMA, when a node is transmitting or receiving, only the reverse (receive) or forward (transmit) channels, respectively, are blocked. Hence the bandwidth utilization of DBTMA is nearly twice that of other RTS/CTS-based schemes. Receiver-Initiated Busy Tone Multiple Access Protocol In the receiver-initiated busy tone multiple access protocol (RI-BTMA), similar to BTMA, the available bandwidth is divided into two channels: a data channel for transmitting data packets and a control channel. The control channel is used by a node to transmit the busy tone signal. A node can transmit on the data channel only if it finds the busy tone to be absent on the control channel. The data packet is divided into two portions: a preamble and the actual data packet. The preamble carries the identification of the intended destination node. Both the data channel and the control channel are slotted, with each slot equal to the length of the preamble. Data transmission consists of two steps. First, the preamble needs to be transmitted by the sender. Once the receiver node acknowledges the reception of this preamble by transmitting the busy tone signal on the control channel, the actual data packet is transmitted. A sender node that needs to transmit a data packet first waits for a free slot, that is, a slot in which the busy tone signal is absent on the control channel. Once it finds

such a slot, it transmits the preamble packet on the data channel. If the destination node receives this preamble packet correctly without any error, it transmits the busy tone on the control channel. It continues transmitting the busy tone signal as long as it is receiving data from the sender. If preamble transmission fails, the receiver does not acknowledge with the busy tone, and the sender node waits for the next free slot and tries again. The operation of the RI-BTMA protocol is shown in Figure 2.12. The busy tone serves two purposes. First, it acknowledges the sender about the successful reception of the preamble. Second, it informs the nearby hidden nodes about the impending transmission so that they do not transmit at the same time.

Figure 2.12. Packet transmission in RI-BTMA.



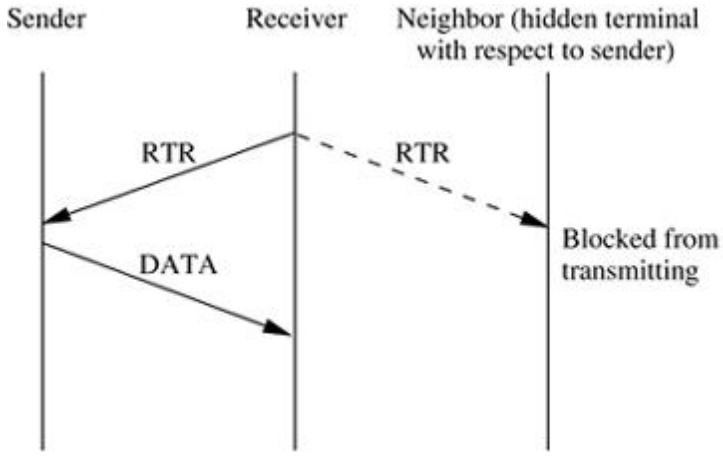
There are two types of RI-BTMA protocols: the basic protocol and the controlled protocol. The basic packet transmission mechanism is the same in both protocols. In the basic protocol, nodes do not have backlog buffers to store data packets. Hence packets that suffer collisions cannot be retransmitted. Also, when the network load increases, packets cannot be queued at the nodes. This protocol would work only when the network load is not high; when network load starts increasing, the protocol becomes unstable. The controlled protocol overcomes this problem. This protocol is the same as the basic protocol, the only difference being the availability of backlog buffers at nodes. Therefore, packets that suffer collisions, and those that are generated during busy slots, can be queued at nodes. A node is said to be in the backlogged mode if its backlog buffer is non-empty. When a node in the backlogged mode receives a packet from its higher layers, the packet is put into the buffer and transmitted later. Suppose the packet arrives at a node

when it is not in the backlogged mode, then if the current slot is free, the preamble for the packet is transmitted with probability *p* in the current slot itself (not transmitted in the same slot with probability (1 - *p*)). If the packet was received during a busy slot, the packet is just put into the backlog buffer, where it waits until the next free slot. A backlogged node transmits a backlogged packet in the next idle slot with a probability *q.* All other packets in the backlog buffer just keep waiting until this transmission succeeds. This protocol can work for multi-hop radio networks as well as for single-hop fully connected networks.

2.7.4MACA-By Invitation

MACA-by invitation (MACA-BI) is a receiver-initiated MAC protocol. It reduces the number of control packets used in the MACA protocol. MACA, which is a sender-initiated protocol, uses the three-way handshake mechanism (which was shown in Figure 2.11), where first the RTS and CTS control packets are exchanged, followed by the actual DATA packet transmission. MACA-BI eliminates the need for the RTS packet. In MACA-BI the receiver node initiates data transmission by transmitting a ready to receive (RTR) control packet to the sender (Figure 2.13). If it is ready to transmit, the sender node responds by sending a DATA packet. Thus data transmission in MACA-BI occurs through a two-way handshake mechanism.

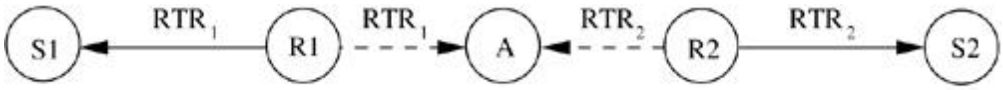
Figure 2.13. Packet transmission in MACA-BI.



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

packet was used to inform the hidden terminals (nodes) about the impending DATA packet transmission, so that they do not transmit at the same time and disrupt the session. This role is played in MACA-BI by the RTR packets. An RTR packet carries information about the time interval during which the DATA packet would be transmitted. When a node hears RTR packets transmitted by its neighbors, it can obtain information about the duration of DATA packet transmissions by nodes that may be either its direct one-hop neighbors or its two hop neighbors, that is, hidden terminals. Since it has information about transmissions by the hidden terminals, it refrains from transmitting during those periods (Figure 2.13). Hence the hidden terminal problem is overcome in MACA-BI. Collision among DATA packets is impossible. However, the hidden terminal problem still affects the control packet transmissions. This leads to protocol failure, as in certain cases the RTR packets can collide with DATA packets. One such scenario is depicted in Figure 2.92. Here, RTR packets transmitted by receiver nodes R1 and R2 collide at node A. So node A is not aware of the transmissions from nodes S1 and S2. When node A transmits RTR packets, they collide with DATA packets at receiver nodes R1 and R2.

Figure 2.14. Hidden terminal problem in MACA-BI.



S1, S2 – Sender nodes
R1, R2 – Receiver nodes
A – Neighbor node

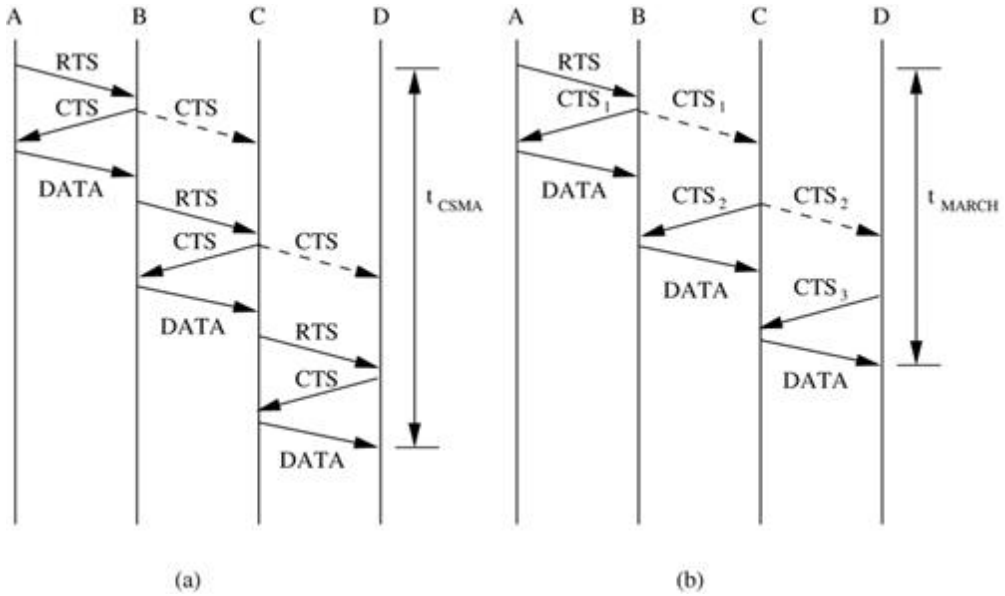
The efficiency of the MACA-BI scheme is mainly dependent on the ability of the receiver node to predict accurately the arrival rates of traffic at the sender nodes.

2.7.5 Media Access with Reduced Handshake

The media access with reduced handshake protocol (MARCH) is a receiver-initiated protocol. MARCH, unlike MACA-BI, does not require any traffic prediction mechanism. The protocol exploits the broadcast nature of traffic from omnidirectional antennas to reduce the number of handshakes involved in data transmission. In MACA , the RTS-CTS control packets exchange takes place before the transmission of every data packet. But in MARCH, the RTS packet is used only for the first packet of the stream. From the second packet onward, only the CTS packet is used. A node obtains information about data packet arrivals at its neighboring nodes by overhearing the CTS packets transmitted by them. It then sends a CTS packet to the concerned neighbor node for relaying data from that node. This mechanism is illustrated inFigure 2.15. Figure 2.15 (a) depicts the packet exchange mechanism of MACA. Here two control packets RTS and CTS need to be exchanged before each data packet is transmitted. It can be seen from this figure that node C, for example, can hear both the CTS and the RTSpackets transmitted by node B. MARCH uses this property of the broadcast channel to reduce the twoway handshake

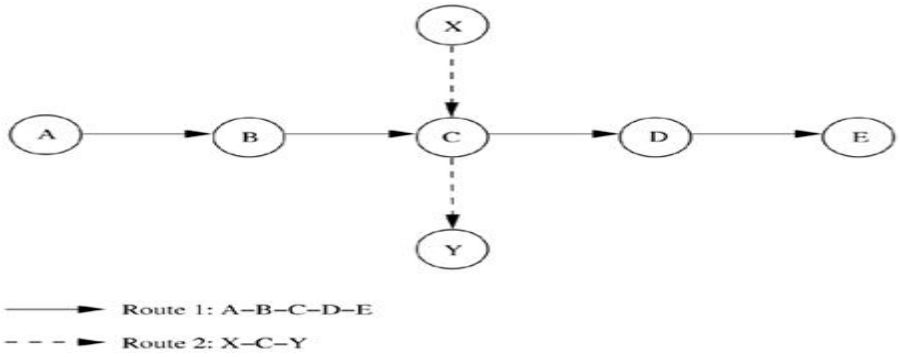
into a single CTS-only handshake. Figure 2.15 (b) shows the handshake mechanism of MARCH. Here, when node B transmits the CTS_1 packet, this packet is also heard by node C. A CTS packet carries information regarding the duration of the next data packet. Node C therefore determines the time at which the next data packet would be available at node B. It sends the CTS_2 packet at that point of time. On receiving the CTS_2packet, node B sends the data packet directly to node C. It can be observed from the figure that the time taken for a packet transmitted by node A to reach node D in MARCH, that is,t_{MARCH}, is less compared to the time taken in MACA, t_{MACA}.

Figure 2.15. Handshake mechanism in (a) MACA and (b) MARCH.



The CTS packet carries the MAC addresses of the sender and the receiver node, and the route identification number (RT_{id}) for that flow. The RT_{id} is used by nodes in order to avoid misinterpretation of CTS packets and initiation of false CTS-only handshakes. Consider Figure 2.16. Here there are two routes – Route 1: A-B-C-D-E and Route 2: X-C-Y. When node C hears a CTS packet transmitted by node B, by means of the RT_{id} field on the packet, it understands that the CTS was transmitted by its upstream node (upstream node refers to the next hop neighbor node on the path from the current node to the source node of the data session) on Route 1. It invokes a timer T which is set to expire after a certain period of time, long enough for node B to receive a packet from node A. A CTS packet is transmitted by node C once the timer expires. This CTS is overheard by node Y also, but since the RT_{id} carried on the CTS is different from the RT_{id} corresponding to Route 2, node Y does not respond. In MARCH, the MAC layer has access to tables that maintain routing information (such asRT_{id}), but the protocol as such does not get involved in routing.

Figure 2.16. Example topology.



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

2.8 CONTENTION-BASED PROTOCOLS WITH RESERVATION MECHANISMS

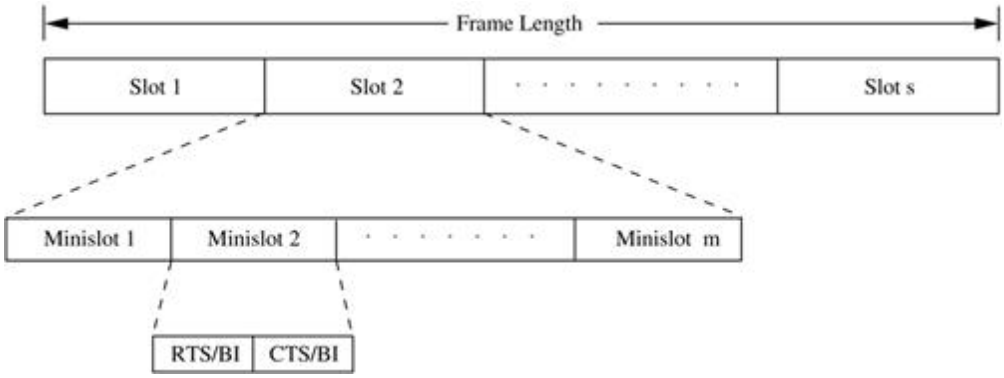
Protocols described in this section have certain mechanisms that aid the nodes in effecting bandwidth reservations. Though these protocols are contention-based, contention occurs only during the resource (bandwidth) reservation phase. Once the bandwidth is reserved, the node gets exclusive access to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

2.8.1 Distributed Packet Reservation Multiple Access Protocol

The distributed packet reservation multiple access protocol (D-PRMA) [9] extends the earlier centralized packet reservation multiple access (PRMA) [10] scheme into a distributed scheme that can be used in ad hoc wireless networks. PRMA was proposed for voice support in a wireless LAN with a base station, where the base station serves as the fixed entity for theMAC operation. D-PRMA extends this protocol for providing voice support in ad hoc wireless networks. D-PRMA is a TDMA-based scheme. The channel is divided into fixed- and equal-sized frames along the time axis (Figure 2.17). Each frame is composed of *s* slots, and each slot consists of *m* minislots. Each minislot can be further divided into two control fields, RTS/BI and CTS/BI(BI stands for busy indication), as shown in the figure. These control fields are used for slot reservation and for overcoming the hidden terminal problem. Details on how this is done will be explained later in this section. All nodes having packets ready for transmission contend for the first minislot of each slot. The remaining (*m* - 1) minislots are granted to the node that wins the contention. Also, the same slot in each subsequent frame can be reserved for this

winning terminal until it completes its packet transmission session. If no node wins the first minislot, then the remaining minislots are continuously used for contention, until a contending node wins any minislot. Within a reserved slot, communication between the source and receiver nodes takes place by means of either time division duplexing (TDD) or frequency division duplexing (FDD). Any node that wants to transmit packets has to first reserve slots, if they have not been reserved already. A certain period at the beginning of each minislot is reserved for carrier-sensing. If a sender node detects the channel to be idle at the beginning of a slot (minislot 1), it transmits an RTS packet (slot reservation request) to the intended destination through the RTS/BI part of the current minislot. On successfully receiving this RTS packet, the receiver node responds by sending a CTS packet through theCTS/BI of the same minislot. If the sender node receives this CTS successfully, then it gets the reservation for the current slot and can use the remaining minislots, that is, minislots 2 to*m.* Otherwise, it continues the contention process through the subsequent minislots of the same slot.

Figure 2.17. Frame structure in D-PRMA.



In order to prioritize nodes transmitting voice traffic (voice nodes) over nodes transmitting normal data traffic (data nodes), two rules are followed in D-PRMA. According to the first rule, the voice nodes are allowed to start contending from minislot 1 with probability $p = 1$; data nodes can start contending only with probability $p < 1$. For the remaining $(m - 1)$ minislots, both the voice nodes and the data nodes are allowed to contend with probability $p < 1$. This is because the reservation process for a voice node is triggered only after the arrival of voice traffic at the node; this avoids unnecessary reservation of slots. According to the second rule, only if the node winning the minislot contention is a voice node, is it permitted to reserve the same slot in each subsequent frame until the end of the session. If a data node wins the contention, then it is allowed to use only one slot, that is, the current slot, and it has to make fresh reservations for each subsequent slot. Nodes that are located within the radio coverage of the receiver should not be permitted to transmit simultaneously when the receiver is receiving packets. If permitted, packets transmitted by them may collide with the packets of the on-going traffic being received at the receiver. Though a node which is located outside the range of a receiver is able to hear packets transmitted by the sender, it should still be allowed to transmit simultaneously. The above requirements, in essence, mean that the protocol must be free of the hidden terminal and

exposed terminal problems. In D-PRMA, when a node wins the contention in minislot 1, other terminals must be prevented from using any of the remaining (*m* - 1) minislots in the same slot for contention (*requirement 1*). Also, when a slot is reserved in subsequent frames, other nodes should be prevented from contending for those reserved slots (*requirement 2*). The RTS-CTS exchange mechanism taking place in the reservation process helps in trying to satisfy *requirement 1*. A node that wins the contention in minislot 1 starts transmitting immediately from minislot 2. Any other node that wants to transmit will find the channel to be busy from minislot 2. Since an RTS can be sent only when the channel is idle, other neighboring nodes would not contend for the channel until the on-going transmission gets completed. A node sends an RTS in the RTS/BI part of a minislot. Only a node that receives an RTS destined to it is allowed to use the CTS/BI part of the slot for transmitting the CTS. So the CTS packet does not suffer any collision due to simultaneous RTS packet transmissions. This improves the probability for a successful reservation. In order to avoid the hidden terminal problem, all nodes hearing the CTS sent by the receiver are not allowed to transmit during the remaining period of that same slot. In order to avoid the exposed terminal problem, a node hearing the RTS but not the CTS (sender node's neighbor) is still allowed to transmit. But, if the communication is duplex in nature, where a node may transmit and receive simultaneously, even such exposed nodes (that hear RTS alone) should not be allowed to transmit. Therefore, D-PRMA makes such a node defer its transmissions for the remaining time period of the same slot. If an RTS or CTS packet collides, and a successful reservation cannot be made in the first minislot, then the subsequent (*m* - 1) minislots of the same slot are used for contention. For satisfying *requirement 2,* the following is done. The receiver of the reserved slot transmits a busy indication (BI) signal through the RTS/BI part of minislot 1 of the same slot in each of the subsequent frames, without performing a carrier-sense. The sender also performs a similar function, transmitting the BI through the CTS/BI part of minislot 1 of the same slot in each subsequent frame. When any node hears a BI signal, it does not further contend for that slot in the current frame. Because of this, the reserved slot in each subsequent frame is made free of contention. Also, making the receiver transmit the BI signal helps in eliminating the hidden terminal problem, since not all neighbors of the receiver can hear from the sender. Finally, after a node that had made the reservation completes its data transmission and does not anymore require a reserved slot, it just stops transmitting the BI signal. D-PRMA is more suited for voice traffic than for data traffic applications.

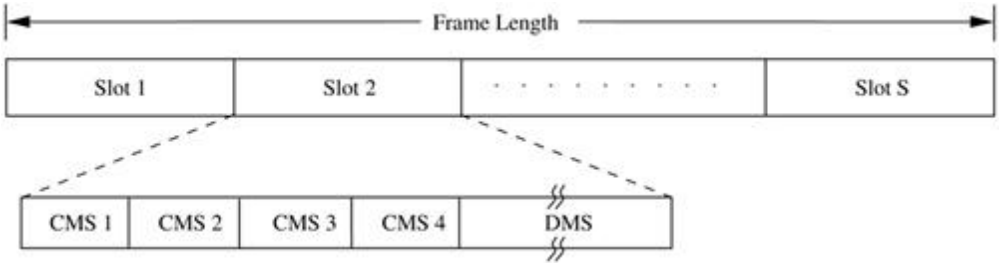
2.8.2 Collision Avoidance Time Allocation Protocol

The collision avoidance time allocation protocol (CATA) is based on dynamic topology dependent transmission scheduling. Nodes contend for and reserve time slots by means of a distributed reservation and handshake mechanism. CATA supports broadcast, unicast, and multicast transmissions simultaneously. The operation of CATA is based on two basic principles:

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

• Usage of negative acknowledgments for reservation requests, and control packet transmissions at the beginning of each slot, for distributing slot reservation information to senders of broadcast or multicast sessions. Time is divided into equal-sized frames, and each frame consists of *S* slots (Figure 2.18). Each slot is further divided into five minislots. The first four minislots are used for transmitting control packets and are called control minislots (CMS1, CMS2, CMS3, and CMS4). The fifth and last minislot, called data minislot (DMS), is meant for data transmission. The data minislot is much longer than the control minislots as the control packets are much smaller in size compared to data packets.

Figure 2.18. Frame format in CATA.



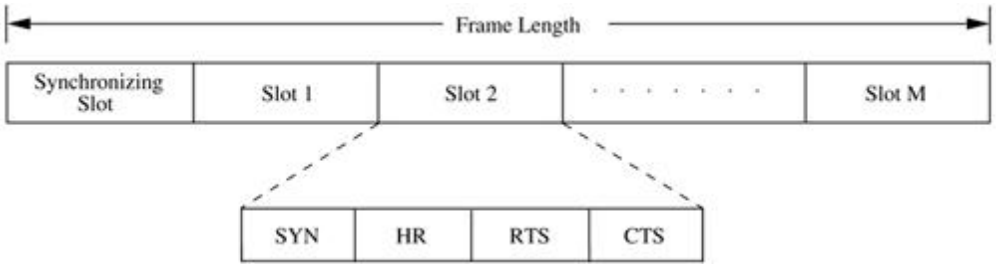
Each node that receives data during the DMS of the current slot transmits a slot reservation (SR) packet during the CMS1 of the slot. This serves to inform other neighboring potential sender nodes about the currently active reservation. The SR packet is either received without error at the neighbor nodes or causes noise at those nodes, in both cases preventing such neighbor nodes from attempting to reserve the current slot. Every node that transmits data during the DMS of the current slot transmits a request-to-send (RTS) packet during CMS2 of the slot. This RTS packet, when received by other neighbor nodes or when it collides with other RTS packets at the neighbor nodes, causes the neighbor nodes to understand that the source node is scheduled to transmit during the DMS of the current slot. Hence they defer their transmissions during the current slot. The control minislots CMS3 and CMS4 are used as follows. The sender of an intended reservation, if it senses the channel to be idle during CMS1, transmits an RTS packet during CMS2. The receiver node of a unicast session transmits a clear-to-send (CTS) packet during CMS3. On receiving this packet, the source node understands that the reservation was successful and transmits data during the DMS of that slot, and during the same slot in subsequent frames, until the unicast flow gets terminated. Once the reservation has been made successfully in a slot, from the next slot onward, both the sender and receiver do not transmit anything during CMS3, and during CMS4 the sender node alone transmits a not-to-send (NTS) packet. The purpose of the NTS packet is explained below. If a node receives an RTS packet for broadcast or multicast during CMS2, or if it finds the channel to be free during CMS2, it remains idle and does not transmit anything during CMS3 and CMS4. Otherwise, it sends a not-to-send (NTS)

packet during CMS4. The NTS packet serves as a negative acknowledgment; a potential multicast or broadcast source node that receives the NTS packet during CMS4, or that detects noise during CMS4, understands that its reservation request had failed, and it does not transmit during the DMS of the current slot. If it finds the channel to be free during CMS4, which implies that its reservation request was successful, it starts transmitting the multicast or broadcast packets during the DMS of the slot. The length of the frame is very important in CATA. For any node (say, node A) to broadcast successfully, there must be no other node (say, node B) in its two-hop neighborhood that transmits simultaneously. If such a node B exists, then if node B is within node A's one-hop neighborhood, node A and node B cannot hear the packets transmitted by each other. If node B is within the two-hop neighborhood of node A, then the packets transmitted by nodes A and B would collide at their common neighbor nodes. Therefore, for any node to transmit successfully during one slot in every frame, the number of slots in each frame must be larger than the number of two-hop neighbor nodes of the transmitting node. The worst-case value of the frame length, that is, the number of slots in the frame, would be $Min(d_2 + 1, N)$, where d is the maximum degree (degree of a node refers to the count of one-hop neighbors of the node) of a node in the network, and N is the total number of nodes in the network. CATA works well with simple single-channel half-duplex radios. It is simple and provides support for collision-free broadcast and multicast traffic.

2.8.3 Hop Reservation Multiple Access Protocol The hop reservation multiple access protocol (HRMA) is a multichannel MAC protocol which is based on simple half-duplex, very slow frequency-hopping spread spectrum (FHSS) radios. It uses a reservation and handshake mechanism to enable a pair of communicating nodes to reserve a frequency hop, thereby guaranteeing collision-free data transmission even in the presence of hidden terminals. HRMA can be viewed as a time slot reservation protocol where each time slot is assigned a separate frequency channel. Out of the available L frequency channels, HRMA uses one frequency channel, denoted by f_0 , as a dedicated synchronizing channel. The nodes exchange synchronization information on f_0 . The remaining L - 1 frequencies are divided into frequency pairs (denoted by (f_i,), i = 1, 2, 3, .., M), thereby restricting the length of the hopping sequence to M. f_i is used for transmitting and receiving hop-reservation (HR) packets, request-to-send (RTS) packets, clearto- send (CTS) packets, and data packets. is used for sending and receiving acknowledgment (ACK) packets for the data packets received or transmitted on frequency f_i. In HRMA, time is slotted, and each slot is assigned a separate frequency hop, which is one among the M frequency hops in the hopping sequence. Each time slot is divided into four periods, namely, synchronizing period, HR period, RTS period, and CTS period, each period meant for transmitting or receiving the synchronizing packet, FR packet, RTS packet, and CTS packet, respectively. All idle nodes, that is, nodes that do not transmit or receive packets currently, hop together. During the synchronizing period of each slot, all idle nodes hop to the synchronizing frequency f_0 and exchange synchronization information. During the HR, RTS, and CTS periods, they just stay idle, dwelling on the common frequency hop assigned to each slot. In addition to the synchronization period used for synchronization purposes, an exclusive synchronization slot

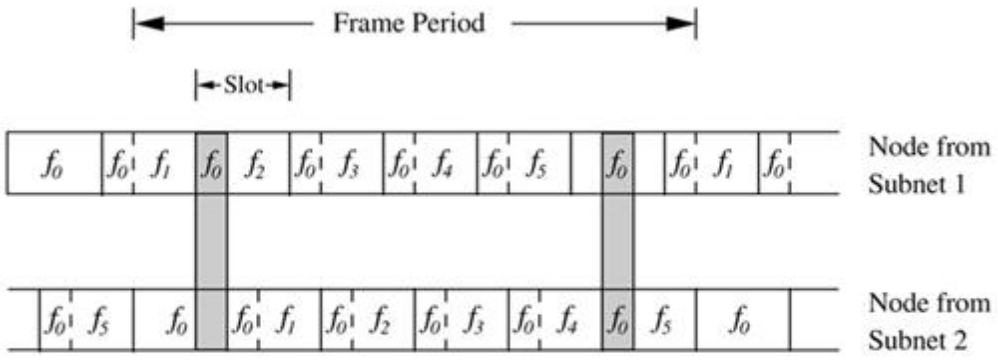
is also defined at the beginning of each HRMA frame (Figure 2.19). This slot is of the same size as that of the other normal slots. All idle nodes dwell on the synchronizing frequency f_0 during the synchronizing slot and exchange synchronization information that may be used to identify the beginning of a frequency hop in the common hopping sequence, and also the frequency to be used in the immediately following hop. Thus the HRMA frame, as depicted in Figure 2.19, is composed of the single synchronizing slot, followed by *M* consecutive normal slots.

Figure 2.19. Frame format in HRMA.



When a new node enters the network, it remains on the synchronizing frequency f_0 for a long enough period of time so as to gather synchronization information such as the hopping pattern and the timing of the system. If it receives no synchronization information, it assumes that it is the only node in the network, broadcasts its own synchronization information, and forms a one node system. Since synchronization information is exchanged during every synchronization slot, new nodes entering the system can easily join the network. If μ is the length of each slot and μ_s the length of the synchronization period on each slot, then the dwell time of f_0 at the beginning of each frame would be $\mu + \mu_s$. Consider the case where nodes from two different disconnected network partitions come nearby. Figure 2.20depicts the worst-case frequency overlap scenario. In the figure, the maximum number of frequency hops $M = 5$. It is evident from the figure that within any time period equal to the duration of a HRMA frame, any two nodes from the two disconnected partitions always have at least two overlapping time periods of length μ_s on the synchronizing frequency f_0. Therefore, nodes belonging to disconnected network components can easily merge into a single network.

Figure 2.20. Merging of subnets.



f_0 synchronizing frequency

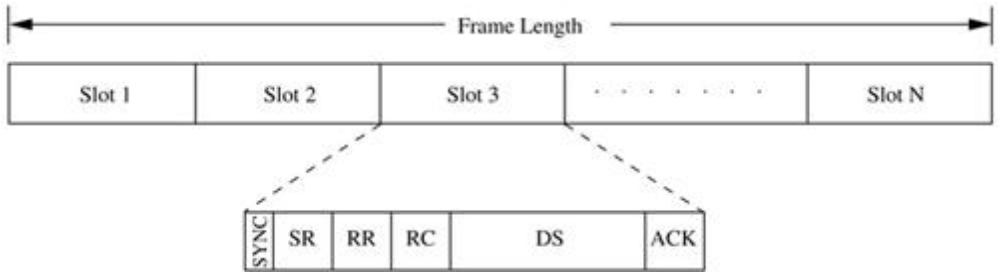
M = 5

When a node receives data to be transmitted, it first listens to the HR period of the immediately following slot. If it hears an HR packet, it backs off for a randomly chosen period (which is a multiple of slot time). If it finds the channel to be free during the SR period, it transmits an RTS packet to the destination during the RTS period of the slot and waits for the CTS packet. On receiving the RTS, the destination node transmits the CTS packet during the CTS period of the same slot, stays on the same frequency currently being used, and waits for the data packet. If the source node receives the CTS packet correctly, it implies that the source and receiver nodes have successfully reserved the current hop. In case the source node does not receive any CTS packet, it backs off for a random number of time slots and repeats the entire process again. The source and receiver nodes dwell on the same reserved frequency throughout the data transmission process, which starts immediately after the CTS period. As mentioned earlier, a separate frequency (, i = 1, 2, ..., M) is used for transmitting acknowledgments. After transmitting each data packet, the source node hops onto this acknowledgment frequency. The receiver sends an acknowledgment (ACK) packet back to the source on this acknowledgment frequency. Once the ACK packet transmission/reception is over, both the source and receiver hop back to the reserved frequency to continue with the data transmission. After the CTS period of a slot, the idle nodes that do not transmit or receive packets hop onto the synchronization frequency f_0 and exchange synchronization information. They dwell on $f_0$for a time period of $\mu_s$and then hop onto the next frequency hop in the common hopping sequence. The data packets transmitted can be of any size. Data transmission can take place through a single packet or through a train of packets. A maximum dwell period has been defined in order to prevent nodes from hogging onto a particular frequency channel. Therefore, the transmission time for the data packet, or the train of data packets, should not exceed this maximum dwell time. Suppose the sender needs to transmit data packets across multiple frames, then it informs the receiver node through the header of the data packet it transmits. On reading this information, the receiver node transmits an HR packet during the HR period of the same slot in the next frame. The neighbor nodes of the receiver on hearing this HRpacket refrain from using the frequency hop reserved. On receiving the HR packet, the source node of the session sends an RTS packet during the RTS period and jams other RTS packets (if any) destined to its neighbors, so that the neighbor nodes do not interfere

on the reserved frequency hop. Both the sender and the receiver remain silent during the CTS period, and data transmission resumes once this CTS period gets over.

2.8.4 Soft Reservation Multiple Access with Priority Assignment Soft reservation multiple access protocol with priority assignment (SRMA/PA) was developed with the main objective of supporting integrated services of real-time and non-realtime applications in ad hoc wireless networks, at the same time maximizing the statistical multiplexing gain. Nodes use a *collision-avoidance* handshake mechanism and a*soft reservation* mechanism in order to contend for and effect reservation of time slots. The soft reservation mechanism allows any urgent node, transmitting packets generated by a real-time application, to take over the radio resource from another node of a non-real-time application on an on-demand basis. SRMA/PA is a TDMA-based protocol in which nodes are allocated different time slots so that the transmissions are collision-free. The main features of SRMA/PA are a unique frame structure and soft reservation capability for distributed and dynamic slot scheduling, dynamic and distributed access priority assignment and update policies, and a time-constrained back-off algorithm. Time is divided into frames, with each frame consisting of a fixed number (*N*) of time slots. The frame structure is shown in Figure 2.21. Each slot is further divided into six different fields, SYNC, soft reservation (SR), reservation request (RR), reservation confirm (RC), data sending (DS), and acknowledgment (ACK). The SYNC field is used for synchronization purposes. The SR, RR, RC, and ACK fields are used for transmitting and receiving the corresponding control packets. The DS field is used for data transmission. The SR packet serves as a busy tone. It informs the nodes in the neighborhood of the transmitting node about the reservation of the slot. The SR packet also carries the access priority value assigned to the node that has reserved the slot. When an idle node receives a data packet for transmission, the node waits for a free slot and transmits the RR packet in the RR field of that slot. A node determines whether or not a slot is free through the SR field of that slot. In case of a voice terminal node, the node tries to take control of the slot already reserved by a data terminal if it finds its priority level to be higher than that of the data terminal. This process is called *soft reservation.* This makes the SRMA/PA different from other protocols where even if a node has lower access priority compared to other ready nodes, it proceeds to complete the transmission of the entire data burst once it has reserved the channel.

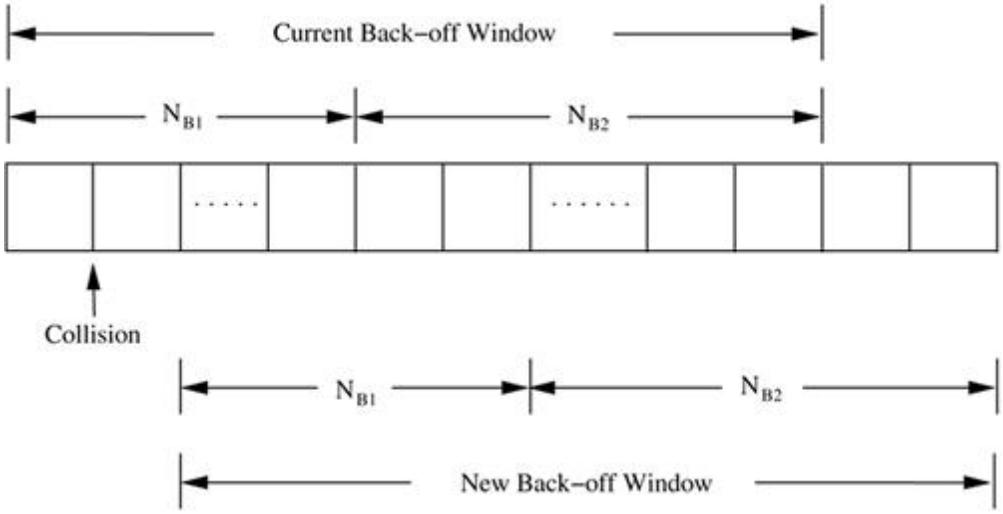
Figure 2.21. Frame structure in SRMA/PA.



Priority levels are initially assigned to nodes based on the service classes (real-time or non-realtime) in a static manner. Once the node acquires the channel, the corresponding slot stays reserved for the node until the node completes transmitting the entire data burst. The node is assigned a pre specified priority, or , respectively, for voice and data terminals. R denotes that the node is a reserved node, that is, a node that has successfully reserved the slot. It is required that , such that delay-sensitive voice applications get preference over normal data applications. Whenever the reservation attempt fails due to collision, the access priority of the node is updated based on the urgency of its packets. A node that is currently transmitting is said to be in the active state. A node is said to be in the idle state if it does not have any packet to be transmitted. In the active state itself, nodes can be in one of the two states: access state and reserved state. Access state is one in which the node is backlogged and is trying to reserve a slot for transmission. The node is said to be in the reserved state if it has already reserved the slot for transmission. Whenever the access priority level of a voice terminal in the access state becomes greater than that of a data terminal in the reserved state, which could be known from the SR field, the corresponding slot is taken over by the prioritized voice terminal. In order to effect this mechanism, the values of priority levels must be such that , where and are the access priority values of a voice terminal and data terminal, respectively, after its n_{th} reservation attempt results in a collision. This soft reservation feature of SRMA/PA, where a voice terminal can take over the slots reserved by a data terminal whenever, due to the urgent nature of its traffic, its access priority becomes higher than that of the data terminal, helps in maximizing the statistical multiplexing gain for voice-data integrated services. The RR-RC-DS-ACK exchange mechanism of SRMA/PA is similar to the RTS-CTS-DATAACK exchange mechanism of MACAW . The RR and RC packets help in eliminating the hidden terminal problem. The major difference between SRMA/PA and CATA protocol is that, while in CATA the slot reservation (SR) packet is transmitted by the receiver of the session, in SRMA/PA it is sent by the source node. Also, the soft reservation feature of SRMA/PA is absent in CATA. The access priorities are assigned to nodes and updated in a distributed and dynamic manner. This allows dynamic sharing of the shared channel. On receiving a new packet for transmission, an idle node becomes active. Now, transition to the access state is made with the initial access priority assigned a value or , depending on whether it is a voice or data terminal. If the random access attempt to effect the reservation by transmitting an RRpacket ends up in a collision, then the access priorities of the node concerned are increased as follows: where Δp_v and Δp_d are the incremental access priorities for voice and data services, respectively. They reflect the urgency of the traffic queued at the two types of terminals, and are given as below: where τ_s is the slot duration, τ_r is the residual lifetime for the voice service, l_Q is the queue length, and α is a scaling coefficient. In order that the access priority of a voice terminal is always higher than that of a data terminal, the following constraint is followed: Though dynamic assignment and update of access priority values are followed in SRMA/PA, collisions among nodes with the same priority and carrying traffic of the same service types cannot be avoided completely. Collisions occur during the RR field of the slot. In order to avoid collisions, a binary exponential back-off algorithm is used for

non-real-time connections, and a modified binary exponential back-off algorithm is used for real-time connections. The modified algorithm implements a priority access policy in order to meet the delay requirements of realtime sessions. Here the back-off window is divided into two different regions, each region having a length of N_{B1} and N_{B2}, respectively, for real-time and non-real-time traffic. Each node checks the laxity of its head-of-line packet (laxity is the difference between the maximum access delay allowed and the residual lifetime of the packet). If the laxity exceeds the threshold T_{limit} slots, one slot out of the N_{B1} slots is selected randomly. Otherwise, one out of the N_{B2} slots is chosen randomly, that is, if the node is unable to make a reservation within the given time (T_{limit} slots), it is given a higher priority compared to other non-real-time nodes by choosing a slot from N_{B1}; otherwise, the node selects a slot from N_{B2}. The RR packet is transmitted on this chosen slot. Here again, if more than one node selects the same random slot and their RR packets collide again, a new back-off window starts immediately after the current slot. This is shown in Figure 2.22. The above back-off mechanism, which gives high preference to nodes transmitting delay-sensitive traffic, helps in guaranteeing the QoS requirements of realtime services in the network. The parameters N_{B1}, N_{B2}, and T_{limit} significantly affect the performance of the protocol, and must be chosen carefully based on the traffic load expected on the network.

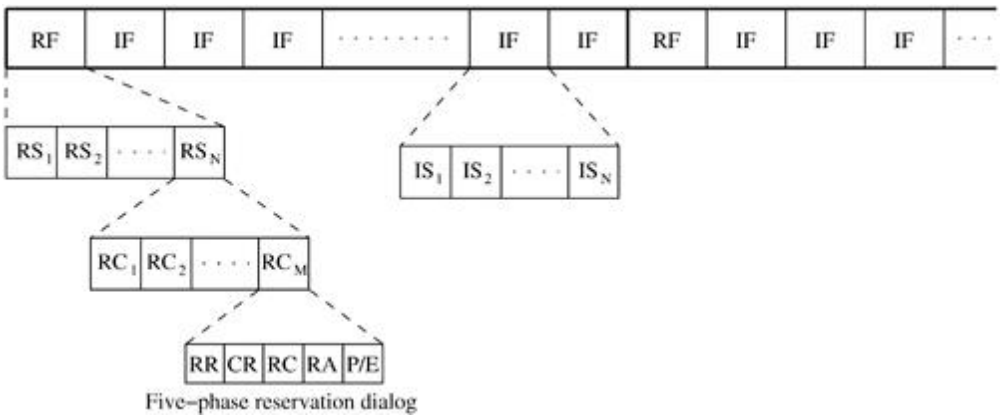
Figure 2.22. Back-off windows.



2.8.5 Five-Phase Reservation Protocol The five-phase reservation protocol (FPRP) is a single-channel time division multiple access (TDMA)-based broadcast scheduling protocol. Nodes use a contention mechanism in order to acquire time slots. The protocol is fully distributed, that is, multiple reservations can be simultaneously made throughout the network. No ordering among nodes is followed; nodes need not wait for making time slot reservations. The slot reservations are made using a five-phase reservation process. The reservation process is localized; it involves only the nodes located within the two-hop radius of the node concerned. Because of this, the protocol is insensitive to the network size, that is, it is scalable. FPRP also

ensures that no collisions occur due to the hidden terminal problem. Time is divided into frames. There are two types of frames: reservation frame (RF) and information frame (IF). Each RF is followed by a sequence of IFs. Each RF has *N* reservation slots (RS), and each IF has *N* information slots (IS). In order to reserve an IS, a node needs to contend during the corresponding RS. Based on these contentions, a TDMA schedule is generated in the RF and is used in the subsequent IFs until the next RF. The structure of the frames is shown in Figure 2.23. Each RS is composed of *M* reservation cycles (RC). Within each RC, a five-phase dialog takes place, using which a node reserves slots. If a node wins the contention in an RC, it is said to have reserved the IS corresponding to the current RS in the subsequent IFs of the current frame. Otherwise, the node contends during the subsequent RCs of the current RS until itself or any other node (1-hop or 2-hop neighbor) succeeds. During the corresponding IS, a node would be in one of the following three states: transmit (T), receive (R), or blocked (B). The five-phase dialog ensures that the protocol is free from the hidden node problem, and also ensures that once a reservation is made by a node with a high probability, it gets sole access to the slot within its neighborhood.

Figure 2.23. Frame structure in FPRP.



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

1. Reservation request phase: Nodes that need to transmit packets send reservation request (RR) packets to their destination nodes.

2. Collision report phase: If a collision is detected by any node during the reservation request phase, then that node broadcasts a collision report (CR) packet. The corresponding source nodes, upon receiving the CR packet, take necessary action.

3. Reservation confirmation phase: A source node is said to have won the contention for a slot if it does not receive any CR messages in the previous phase. In order to confirm the reservation request made in the reservation request phase, it sends a reservation confirmation (RC) message to the destination node in this phase.

4. Reservation acknowledgment phase: In this phase, the destination node acknowledges reception of the RC by sending back a reservation acknowledgment (RA) message to the source. The hidden nodes that receive this message defer their transmissions during the reserved slot.

5. Packing and elimination (P/E) phase: Two types of packets are transmitted during this phase: packing packet and elimination packet. The details regarding the use of these packets will be described later in this section. Each of the above five phases is described below.

Reservation request phase:

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

Collision report phase:

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

Reservation confirmation phase:

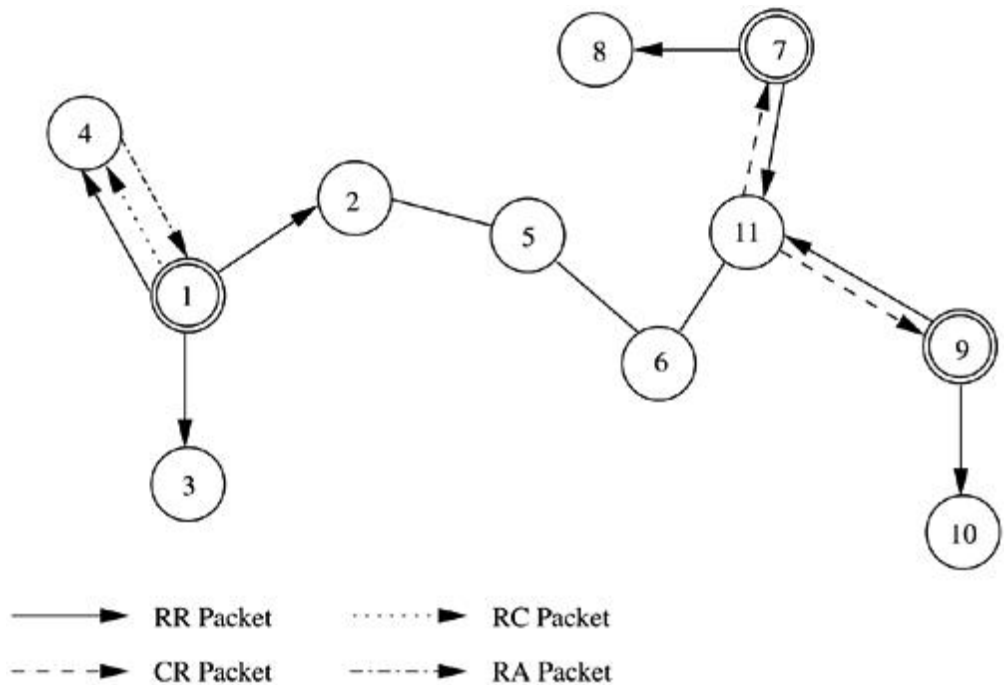
An RN that does not receive any CR packet in the previous phase, that is, a TN, sends an RC packet to the destination node. Each neighbor node that receives this packet understands that the slot has been reserved, and defers its transmission during the corresponding information slots in the subsequent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

Reservation acknowledgment phase:

On receiving the RC packet, the intended receiver node responds by sending an RA packet back to the TN. This is used to inform the TN that the reservation has been established. In case the TN is isolated and is not connected to any other node in the network, then it would not receive the RA packet, and thus becomes aware of the fact that it is isolated. Thus the RC packet prevents such isolated nodes from transmitting further. The reservation acknowledgment phase also serves another purpose. Other two-hop neighbor nodes that receive this RA packet get blocked from transmitting. Therefore, they do not disturb the transmission that is to take place in the reserved slots. When more than two TNs are located nearby, it results in a deadlock condition. Such situations may occur when there is no common neighbor node present when the RNs transmit

RR packets. Collisions are not reported in the next phase, and so each node claims success and becomes a TN. Deadlocks are of two types: isolated and non-isolated. An isolated deadlock is a condition where none of the deadlocked nodes is connected to any non-deadlocked node. In the non isolated deadlock situation, at least one deadlocked node is connected to a non-deadlocked neighbor node. The RA phase can resolve isolated deadlocks. None of the nodes transmits RA, and hence the TNs abort their transmissions. *Packing/elimination (P/E) phase:* In this phase, a packing packet (PP) is sent by each node that is located within two hops from a TN, and that had made a reservation since the previous P/E phase. A node receiving a PP understands that there has been a recent success in slot reservation three hops away from it, and because of this some of its neighbors would have been blocked during this slot. The node can take advantage of this and adjust its contention probability *p,* so that convergence is faster. In an attempt to resolve a non-isolated deadlock, each TN is required to transmit an elimination packet (EP) in this phase, with a probability 0.5. A deadlocked TN, on receiving an EP before transmitting its own EP, gets to know about the deadlock. It backs off by marking the slot as reserved and does not transmit further during the slot. Consider Figure 2.24. Here nodes 1, 7, and 9 have packets ready to be transmitted to nodes 4, 8, and 10, respectively. During the reservation request phase, all three nodes transmit RR packets. Since no other node in the two-hop neighborhood of node 1 transmits simultaneously, node 1 does not receive any CR message in the collision report phase. So node 1 transmits an RC message in the next phase, for which node 4 sends back an RA message, and the reservation is established. Node 7 and node 9 both transmit the RR packet in the reservation request phase. Here node 9 is within two hops from node 7. So if both nodes 7 and 9 transmit simultaneously, their RR packets collide at common neighbor node 11. Node 11 sends a CR packet which is heard by nodes 7 and 9. On receiving the CR packet, nodes 7 and 9 stop contending for the current slot.

Figure 2.24. FPRP - Example.



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

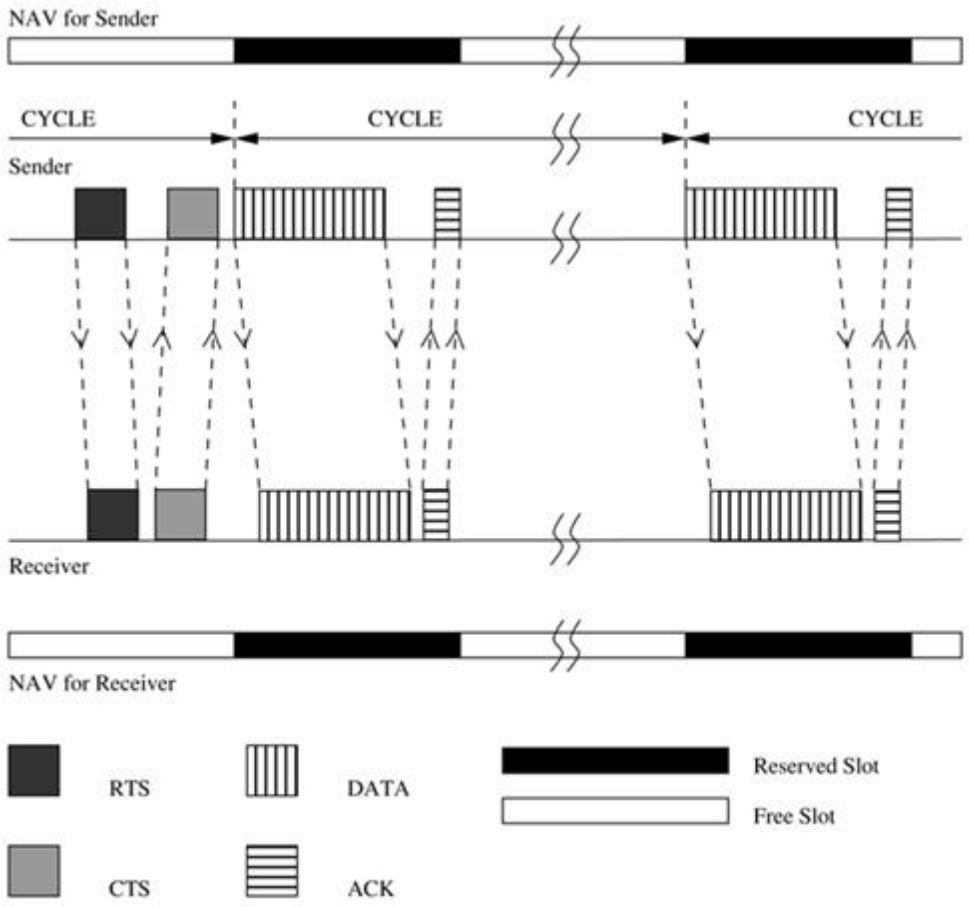
2.8.6MACA with Piggy-Backed Reservation

MACA with piggy-backed reservation (MACA/PR) is a protocol used to provide real-time traffic support in multi-hop wireless networks. The MAC protocol used is based on the MACAW protocol, with the provisioning of non-persistent CSMA (as in FAMA). The main components of MACA/PR are: a MAC protocol, a reservation protocol, and a QoS routing protocol. MACA/PR differentiates real-time packets from the best-effort packets. While providing guaranteed bandwidth support for real-time packets, at the same time it provides reliable transmission of best-effort packets. Time is divided into slots. The slots are defined by the reservations made at nodes, and hence are asynchronous in nature with varying lengths. Each node in the network maintains a reservation table (RT) that records all the reserved transmit and receive slots/windows of all nodes within its transmission range. In order to transmit a non-real-time packet, a MACAW -based MAC protocol is used. The ready node (a node which has packets ready for transmission) first waits for a free slot in the RT. Once it finds a free slot, it again waits for an additional random time of the order of a single-hop round-trip delay time, after which it senses the channel. If the channel is found to be still free, the node transmits an RTS packet, for which the receiver, if it is ready to receive packets, responds with a CTS packet. On receiving the CTS packet, the source node sends a DATA packet, and the receiver, on receiving the packet without any error, finally sends an ACK packet back to the source. The RTS and CTS control packets contain, in them, the time duration in which the DATA packet is to be transmitted. A nearby node that hears these packets avoids transmission during that time. If, after

the random waiting time, the channel is found to be busy, the node waits for the channel to become idle again, and then repeats the same procedure. For real-time traffic, the reservation protocol of MACA/PR functions as follows. The sender is assumed to transmit real-time packets at certain regular intervals, say, every CYCLE time period. The first data packet of the session is transmitted in the usual manner just as a best-effort packet would be transmitted. The source node first sends an RTS packet, for which the receiver node responds with a CTS packet. Now the source node sends the first DATA packet of the real-time session. Reservation information for the next DATA packet to be transmitted (which is scheduled to be transmitted after CYCLE time period) is piggy-backed on this current DATA packet. On receiving this DATA packet, the receiver node updates its reservation table with the piggy-backed reservation information. It then sends an ACK packet back to the source. The receiver node uses the ACK packet to confirm the reservation request that was piggy-backed on the previous DATA packet. It piggy-backs the reservation confirmation information on the ACK packet. Neighbor nodes that hear the DATA and ACK packets update their reservation tables with the reservation information carried by them, and refrain from transmitting when the next packet is to be transmitted. Unlike MACAW ,MACA/PR does not make use of RTS/CTS packets for transmission of the subsequent DATA packets. After receiving the ACK, the source node directly transmits the next DATA packet at its scheduled transmission time in the next CYCLE. This DATA packet in turn would carry reservation information for the next DATA packet. Real-time data transmission, hence, occurs as a series of DATA-ACK packet exchanges. The real-time packets (except for the first packet of the session that is used to initiate the reservation process) are transmitted only once. If an ACK packet is not received for a DATA packet, the source node just drops the packet. The ACK packet therefore serves the purpose of renewing the reservation, in addition to recovering from packet loss. If the source node fails to receive ACK packets for a certain number of consecutive DATA packets, it then assumes the reservation to have been lost. It restarts the real-time session again with an RTS-CTS control packet exchange, either on a different slot on the same link, or on a different link in case of a path break. In order to transmit an RTS to the receiver node, the source needs to find a slot that is free at both the nodes. For maintaining consistent information regarding free slots at all nodes, MACA/PRuses periodic exchange of reservation tables. This periodic table exchange automatically overcomes the hidden terminal problem. When a hidden terminal receives a reservation table from a node, it refrains from transmitting in the reserved slots of that node. Slot reservation information maintained in the reservation tables is refreshed every cycle. If the reservation is not refreshed for a certain number of consecutive cycles, it is then dropped. The transmission of packets in MACA/PR is depicted in Figure 2.25. It can be seen from the figure that theRTS-CTS exchange is used only for transmitting the first packet of the session. Since each DATA packet carries reservation information for the next DATA packet that would be transmitted after CYCLE time period, RTS-CTS exchange is not required for the subsequent DATA packets. Neighbor nodes that receive DATA packets update their reservation tables accordingly and do not contend for the channel during the reserved slots. The network

allocation vector (NAV) at each node reflects the current and future state of the channel as perceived by the node.

Figure 2.25. Packet transmission in MACA/PR.



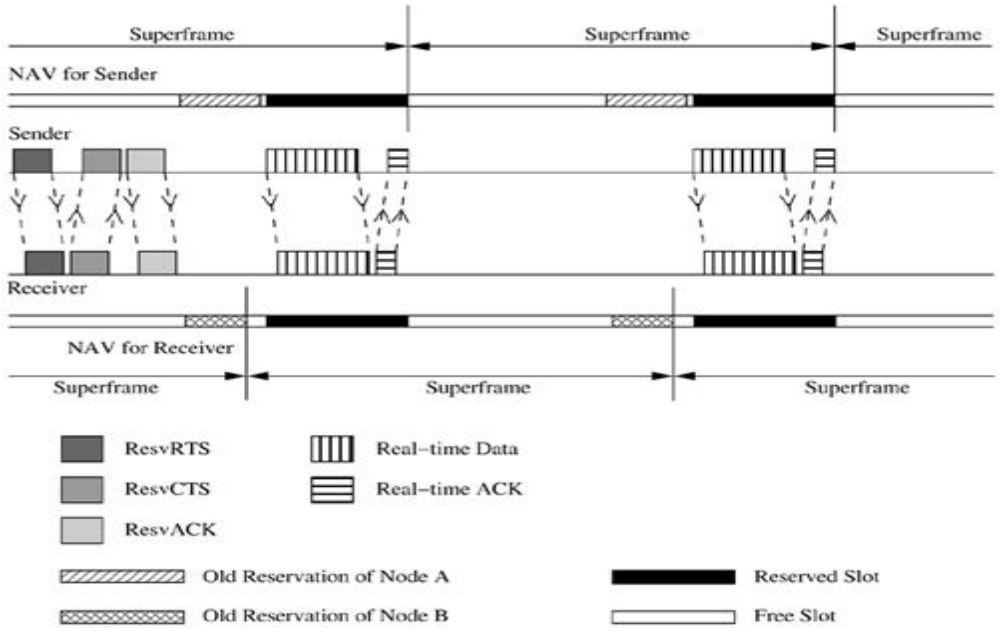
Best-effort packet transmissions and real-time packet transmissions can be interleaved at nodes, with higher priority being given to real-time packets. For real-time packets, MACA/PReffectively works as a TDM system, with a superframe time of CYCLE. The best-effort packets are transmitted in the empty slots (which have not been reserved) of the cycle. When a new node joins the network, it initially remains in the listening mode during which it receives reservation tables from each of its neighbors and learns about the reservations made in the network. After this initial period, the node shifts to its normal mode of operation. The QoS routing protocol used with MACA/PR is the destination sequenced distance vector (DSDV) routing protocol [16] (described in detail in Chapter 7). Bandwidth constraint has been introduced in the routing process. Each node periodically broadcasts to its neighbors the (bandwidth, hop distance) pairs for each preferred path, that is, for each bandwidth value, to each destination. The number of preferred paths is equal to the maximum number of slots in a cycle. After this is done, if a node receives a real-time packet with a certain bandwidth requirement that cannot be satisfied using the current available paths, the packet is dropped and no ACK packet is

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

2.8.7 Real-Time Medium Access Control Protocol

The real-time medium access control protocol (RTMAC) provides a bandwidth reservation mechanism for supporting real-time traffic in ad hoc wireless networks. RTMA Cconsists of two components, a MAC layer protocol and a QoS routing protocol. The MAC layer protocol is a real-time extension of the IEEE 802.11 DCF. The QoS routing protocol is responsible for end-to-end reservation and release of bandwidth resources. The MAC layer protocol has two parts: a medium-access protocol for best-effort traffic and a reservation protocol for real-time traffic. A separate set of control packets, consisting of *ResvRTS*, *ResvRTSResvCTS*, and *ResvACK*, is used for effecting bandwidth reservation for real-time packets. RTS, CTS, and ACK control packets are used for transmitting best-effort packets. In order to give higher priority for real-time packets, the wait time for transmitting a *ResvRTS* packet is reduced to half of DCF inter-frame space (DIFS), which is the wait time used for best-effort packets. Time is divided into superframes. As can be seen from Figure 2.26, the super frame for each node may not strictly align with the other nodes. Bandwidth reservations can be made by a node by reserving variable-length time slots on super frames, which are sufficient enough to carry the traffic generated by the node. The core concept of RTMAC is the flexibility of slot placement in the super frame. Each super frame consists of a number of reservation-slots (resv-slots). The time duration of each resv-slot is twice the maximum propagation delay. Data transmission normally requires a block of resv-slots. A node that needs to transmit real-time packets first reserves a set of resv-slots. The set of resv-slots reserved by a node for a connection on a superframe is called a connection-slot. A node that has made reservations on the current superframe makes use of the same connection-slot in the successive superframes for transmitting packets. Each node maintains a reservation table containing information such as the sender id, receiver id, and starting and ending times of reservations that are currently active within its direct transmission range.

Figure 2.26. Reservation mechanism in RTMAC.



In RTMAC, no time synchronization is assumed. The protocol uses relative time for all reservation purposes. When a node receives this relative-time-based information, it converts the relative time to absolute time by adding its current time maintained in its clock. A three-way handshake protocol is used for effecting the reservation. For example, node A, which wants to reserve a slot with node B, sends a *ResvRTS* packet which contains the relative time information of starting and ending of the connection-slot (a number of resv-slots) to be reserved. Node B, on receiving this packet, first checks its reservation table to see whether it can receive on those resvslots. If it can, it replies by sending a *ResvCTS* packet containing the relative time information of the same resv-slots to be reserved. Neighbor nodes of the receiver, on receiving the *ResvCTS*, update their reservation tables accordingly. Source node A, on receiving the *ResvCTS* packet, responds by sending a *ResvACK* packet. This packet also carries relative time information regarding the reserved slots. The *ResvACK* packet is meant for the neighbor nodes of the source node (node A) which are not aware of the reservation as they may not receive the *ResvCTS* packet. Such nodes update their reservation tables on receiving the *ResvACK* packet. Transmission of the *ResvACK* packet completes the reservation process. Once the reservation is made, real-time packets are transmitted in these reserved slots. Transmission of each real-time packet is followed by the transmission of a real-time ACK (RTACK) packet by the receiver. The bandwidth reservation process is illustrated in Figure 2.26. In the figure, NAV indicates the network allocation vector maintained at each node. As mentioned earlier in Section 2.8.6, the NAV at a node reflects the current and future state of the channel as perceived by the node. The sender node first transmits a *ResvRTS* packet indicating the connection-slot (represented by the offset time from the current time for the beginning and end of the connection-slot) to be reserved. The receiver node on receiving this packet checks its NAVand finds that the requested connection-slot is free. So it responds by sending a *ResvCTS*packet carrying the same connection-slot information. The sender node, on receiving this packet, completes the reservation

process by sending a *ResvACK* packet. The corresponding connection-slot is marked as reserved at both the sender and the receiver. This is indicated inFigure 2.26 by the dark-shaded regions in the NAVs of the sender and receiver. Once the reservation is made, the real-time session gets started, and packets are transmitted in the reserved connection-slot by means of *Real-timeData – Real-timeACK* exchanges. If the receiver node receives the *ResvRTS* packet on a slot which has already been reserved by one of its neighbor nodes, it does not respond with a *ResvCTS* packet. It just discards the received *ResvRTS* packet. This is because, if the node responds with a negative or positive ACK, the ACK packet may cause collisions with the reservations made by its neighbor. The sender node times out and retries later. In case the *ResvRTS* is received on a free slot, but the requested connection-slot is not free at the receiver node, the receiver sends a negative CTS(*ResvNCTS*) back to the sender node. On receiving this, the sender node reattempts following the same procedure but with another free connection-slot. If the real-time session gets finished, or a route break is detected by the sender node, the node releases the resources reserved for that session by sending a reservation release RTS(*ResvRelRTS*) packet. The *ResvRelRTS* packet is a broadcast packet. Nodes hearing this packet update their reservation tables in order to free the corresponding connection slots. In case the receiver node receives this (*ResvRelRTS*) packet, it responds by broadcasting a (*ResvRelCTS*) packet. The receiver's neighbor nodes, on receiving this (*ResvRelCTS*) packet, free up the corresponding reservation slots. In case the downstream node of the broken link does not receive the *ResvRelRTS* packet, since it also does not receive any DATA packet belonging to the corresponding connection, it times out and releases the reservations made. A QoS routing protocol is used with RTMAC to find an end-to-end path that matches the QoS requirements (bandwidth requirements) of the user. The QoS routing protocol used here is an extension of the destination sequenced distance vector (DSDV) routing protocol. When a node receives a data packet for a new connection, the node reserves bandwidth on the forward link and forwards the packet to the next node on the path to the destination. In order to maintain a consistent view of reservation tables of neighboring nodes at each node, each node transmits its reservation information along with the route update packet, which is defined as part of the DSDV protocol. The routing protocol can specify a specific connection-slot to be reserved for a particular connection; this gives flexibility for the routing protocol to decide on the positioning of the connection-slot. But generally, the first available connection-slot is used. One of the main advantages of RTMAC is its bandwidth efficiency. Since nodes operate in the asynchronous mode, successive reservation slots may not strictly align with each other. Hence small fragments of free slots may occur in between reservation slots. If the free slot is just enough to accommodate a DATA and ACK packet, then RTMAC can make use of the free slot by transmitting *ResvRTS-ResvCTS-ResvACK* in some other free slot. Such small free slots cannot be made use of in other protocols such as MACA/PR, which require the free slot to accommodate the entire RTS-CTS-DATA-ACK exchange. Another advantage of RTMAC is its asynchronous mode of operation where nodes do not require any global time synchronization.

2.9 CONTENTION-BASED MAC PROTOCOLS WITH SCHEDULING MECHANISMS

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

2.9.1 Distributed Priority Scheduling and Medium Access in Ad Hoc Networks

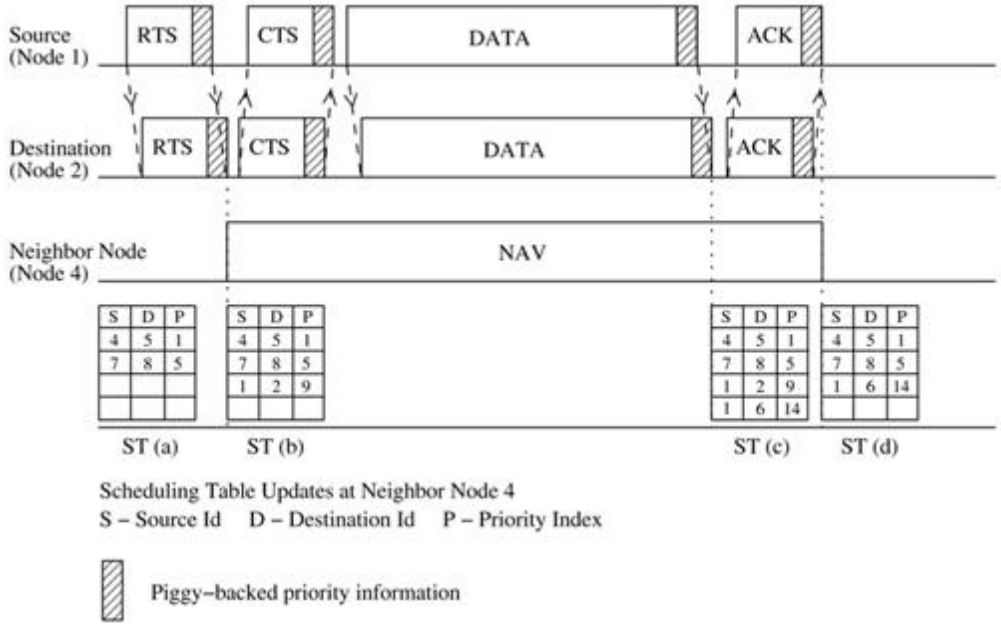
This work, proposed in, presents two mechanisms for providing quality of service (QoS) support for connections in ad hoc wireless networks. The first technique, called distributed priority scheduling (DPS), piggy-backs the priority tag of a node's current and head-of-line packets on the control and data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

Distributed Priority Scheduling

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

received DATA packet. Neighbor nodes receiving the DATA or ACK packets retrieve the piggy-backed information and update their STs accordingly. When a node hears an ACK packet, it removes from its ST any entry made earlier for the corresponding DATA packet. Figure 2.27 illustrates the piggy-backing and table update mechanism. Node 1 needs to transmit a DATA packet (with priority index value 9) to node 2. It first transmits an RTSpacket carrying piggy-backed information about this DATA packet. The initial state of the STof node 4 which is a neighbor of nodes 1 and 2 is shown in ST (a). Node 4, on hearing thisRTS packet, retrieves the piggybacked priority information and makes a corresponding entry in its ST, as shown in ST (b). The destination node 2 responds by sending a CTS packet. The actual DATA packet is sent by the source node once it receives the CTS packet. This DATA packet carries piggy-backed priority information regarding the head-of-line packet at node 1. On hearing this DATA packet, neighbor node 4 makes a corresponding entry for the head-of-line packet of node 1, in its ST. ST (c) shows the new updated status of the ST at node 4. Finally, the receiver node sends an ACK packet to node 1. When this packet is heard by node 4, it removes the entry made for the corresponding DATA packet from its ST. The state of the scheduling table at the end of this data transfer session is depicted in ST (d).

Figure 2.27. Piggy-backing and scheduling table update mechanism in DPS.



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

$$\text{back} - \text{off} = \begin{cases} Uniform[0, (2^t CW_{min}) - 1] & r = 1, n < nmax \\ \alpha \times CW_{min} + Uniform[0, \gamma CW_{min} - 1] & r > 1, n = 0 \\ Uniform[0, (2^n \gamma CW_{min}) - 1] & r > 1, n \geq 1 \end{cases}$$

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

Multi-Hop Coordination

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

$$d_{i,j}{}^{k} = \begin{cases} t_i{}^{k} + \delta_{i,1}{}^{k}, & j = 1 \\ d_{i,j-1}{}^{k} + \delta_{i,j}{}^{k}, & j > 1 \end{cases}$$

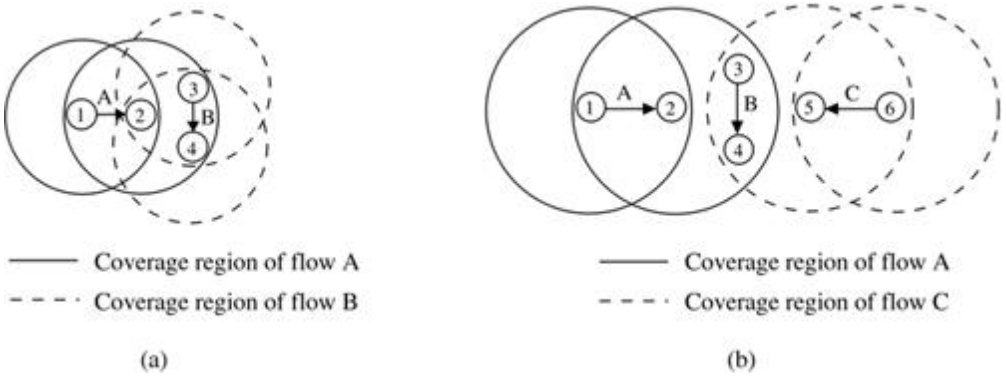
where the increment of the priority index is a non-negative function of *i, j,* , and . Because of this mechanism, if a packet suffers due to excess delay at the upstream nodes, the downstream nodes increase the priority of the packet so that the packet is able to meet its end-toend delay target. Similarly, if a packet arrives very early due to lack of contention at the upstream nodes, then the priority of that packet would be reduced at the downstream nodes. Any suitable scheme can be used for obtaining the values for . One simple scheme, called uniform delay budget allocation, works as follows. For a flow *i* with an end-to-end delay target of *D,* the increment in priority index value for a packet belonging to that flow, at hop *j*, is given as, , where *K* is the length of the flow's path. The distributed priority scheduling and multi-hop coordination schemes described above are fully distributed schemes. They can be utilized for carrying time-sensitive traffic on ad hoc wireless networks.

2.9.2 Distributed Wireless Ordering Protocol

The distributed wireless ordering protocol (DWOP) [19] consists of a media access scheme along with a scheduling mechanism. It is based on the distributed priority scheduling scheme proposed in. DWOP ensures that packets access the medium according to the order specified by an ideal reference scheduler such as first-in-first-out (FIFO), virtual clock, or earliest deadline first. In this discussion, FIFO is chosen as the reference scheduler. In FIFO, packet priority indices are set to the arrival times of packets. Similar to DPS, control packets are used in DWOP to piggy-back priority information regarding head-of-line packets of nodes. As the targeted FIFO

schedule would transmit packets in order of the arrival times, each node builds up a scheduling table (ST) ordered according to the overheard arrival times. The key concept in DWOP is that a node is made eligible to contend for the channel only if its locally queued packet has a smaller arrival time compared to all other arrival times in its ST(all other packets queued at its neighbor nodes), that is, only if the node finds that it holds the next region-wise packet in the hypothetical FIFO schedule. Two additional table management techniques, receiver participation and stale entry elimination, are used in order to keep the actual schedule close to the reference FIFO schedule. DWOP may not suffer due to information asymmetry. Since in most networks all nodes are not within the radio range of each other, a transmitting node might not be aware of the arrival times of packets queued at another node which is not within its direct transmission range. This information asymmetry might affect the fair sharing of bandwidth. For example, in Figure 2.28 (a), the sender of flow B would be aware of the packets to be transmitted by the sender of flow A, and so it defers its transmission whenever a higher priority packet is queued at the sender of flow A. But the sender of flow A is not aware of the arrival times of packets queued at the sender of flow B and hence it concludes that it has the highest priority packet in its neighborhood. Therefore, node 1 unsuccessfully tries to gain access to the channel continuously. This would result in flow B receiving an unfair higher share of the available bandwidth. In order to overcome this information asymmetry problem, the *receiver participation* mechanism is used.

Figure 2.28. (a) Information asymmetry. (b) Perceived collisions.



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

off state after completing the transmission of its current packet. The back-off period $T_{back-off}$ is given by

$$T_{back-off} = R \times (EIFS + DIFS + T_{success} + CW_{min})$$

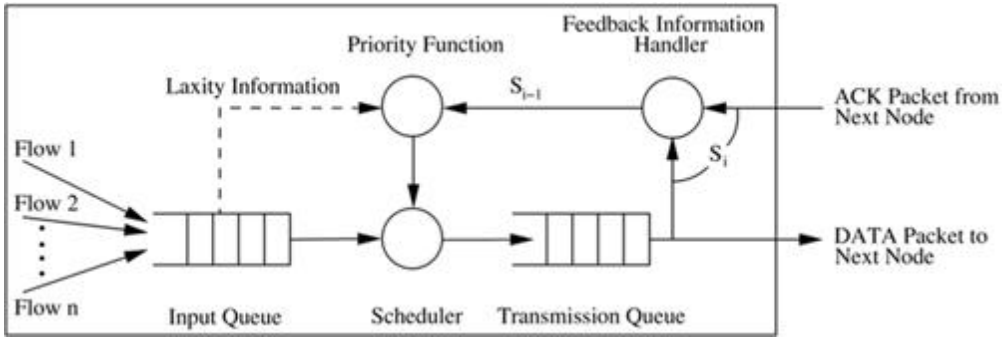
where $T_{success}$ is the longest possible time required to transmit a packet successfully, including the RTS-CTS-DATA-ACK handshake. Thus the node backs off, allowing higher priority packets in the neighborhood of the receiver to get transmitted first. In order to obtain a perfect FIFO schedule, the receiver can very well be made not to reply to the out-of-order requests (RTS) of the sender. This would cause the sender to time out and back off, thereby avoiding any out-of-order transmission. But since the sender has already expended its resources in transmitting the RTS successfully, it is allowed to complete the transmission of its current packet. This is a trade-off between achieving perfect FIFO scheduling and high system utilization. Since in DWOP a node's access to the medium is dependent on its rank in the receiver node's ST (the rank of a node denotes the position of the node's entry in the receiver node's ST as per the priority of the corresponding packet), information maintained in the ST must be consistent with the actual network scenario. The stale entry elimination mechanism makes sure that the STs are free of stale entries. An entry is deleted from the STonly after an ACK packet for the corresponding entry is heard by the node. In case the ACK packet collides at the node, the corresponding entry in the ST will never be removed. This may cause a large deviation from the ideal FIFO schedule. Figure 2.28 (b) shows an example-perceived collisions scenario. The sender and receiver of flow B might have stale entries because of collisions caused by packets belonging to flow A and flow C at the sender and receiver of flow B. It can be observed that, in case there is a stale entry in the ST of a node, the node's own headof- line packet position remains fixed, while other entries below the head-of-line entry keep changing. The above observation is used as a stale entry detection method. Thus, when a node observes that its rank remains fixed while packets whose priorities are below the priority of its head-of-line packet are being transmitted, it concludes that it may have one or more stale entries in its ST. The node simply deletes the oldest entry from its ST, assuming it to be the stale entry. This mechanism thus eliminates stale entries from the STs of nodes. In summary, DWOP tries to ensure that packets get access to the channel according to the order defined by a reference scheduler. The above discussion was with respect to the FIFOscheduler. Though the actual schedule deviates from the ideal FIFO schedule due to information asymmetry and stale information in STs, the receiver participation and the stale entry elimination mechanisms try to keep the actual schedule as close as possible to the ideal schedule.

2.9.3 Distributed Laxity-Based Priority Scheduling Scheme

The distributed laxity-based priority scheduling (DLPS) scheme [20] is a packet scheduling scheme, where scheduling decisions are made taking into consideration the states of neighboring nodes and the feedback from destination nodes regarding packet losses. Packets are reordered based on their uniform laxity budgets (ULBs) and the packet delivery ratios of the flows to

which they belong. Each node maintains two tables: scheduling table (ST) and packet delivery ratio table (PDT). The ST contains information about packets to be transmitted by the node and packets overheard by the node, sorted according to their *priority index* values. Priority index expresses the priority of a packet. The lower the priority index, the higher the packet's priority. The PDT contains the count of packets transmitted and the count of acknowledgment (ACK) packets received for every flow passing through the node. This information is used for calculating current packet delivery ratio of flows (explained later in this section). A node keeps track of packet delivery ratios (used for calculating priority index of packets) of all flows it is aware of by means of a feedback mechanism. Figure 2.29 depicts the overall functioning of the feedback mechanism. Incoming packets to a node are queued in the node's input queue according to their arrival times. The scheduler sorts them according to their priority values and inserts them into the transmission queue. The highest priority packet from this queue is selected for transmission. The node, after transmitting a packet, updates the count of packets transmitted so far in its PDT. The destination node of a flow, on receiving data packets, initiates a feedback by means of which the count of DATA packets received by it is conveyed to the source through ACK packets traversing the reverse path. These two pieces of information, together denoted by S_i in Figure 2.29, are received by the feedback information handler (FIH). The FIH, in parallel, also sends the previous state information S_{i-1} to the priority function module (PFM). The ULB of each packet in ST is available at the node (ULB calculation will be explained later in this section). This information is also sent to PFM, which uses the information fed to it to calculate the priority indices of packets in the ST.

Figure 2.29. Feedback mechanism. ***Reproduced with permission from [20], © Elsevier, 2004.***



Using the count of DATA packets transmitted (*pktsSent*) and count information carried by ACK packets (*acksRcvd*), available in PDT, packet delivery ratio (*PDR*) of the flow at any given time is computed as **(2.9.1)**

$$PDR = \frac{acksRcvd}{pktsSent}$$

Priority index of a packet (*PI*) is defined as **(2.9.2)**

$$PI = \frac{PDR}{M} \times ULB$$

Here, $ULB = \frac{deadline - currentTime}{remHops}$ is the uniform laxity budget of the packet, and *M* is a userdefined parameter representing the desired packet delivery ratio for the flow. *deadline* is the endto- end deadline target of the packet and is equal to (*packet creation time + end-to-end delay target*). *currentTime* denotes the current time according to the node's local clock. When greater numbers of packets belonging to a flow meet their delay targets, the term would have a high value. Hence priority index would be high for packets of that flow, and therefore the actual priority of the packets would be low. When very few packets of a flow meet their delay targets, the value of would be much less, thereby lowering the priority index and increasing the priority of packets of that flow. *ULB* also plays an equally important role. Since *remHops*, the number of hops remaining to be traversed, is in the denominator of the expression for *ULB*, when a packet is near its source and needs to traverse several hops to reach its destination, its priority index value will be lowered, thereby increasing its priority. When it nears its destination, the fewer number of hops to be traversed tends to increase the priority index, thereby lowering its priority. RTS and CTS packets transmitted by a node are modified to carry piggy-backed information regarding the *highest priority packet queued at the node.* Similarly, DATA and ACK packets transmitted by a node carry piggy-backed information corresponding to the *highest priority packet entry in the ST of the node.* A node hearing any packet retrieves the piggy-backed priority information, calculates the priority index of the corresponding packet, and adds a corresponding entry in its ST. A DATA packet also carries information about itself. The end-to-end delay target, remaining number of hops, actual source ID, and the flow ID constitute this information. A node, on receiving a DATA packet, using the above information and the information maintained in thePDT, can obtain the priority index of the packet (*PI*), as given in Equation 2.9.2. Since the priority index of a packet keeps changing with time, it needs to be updated constantly. Each node, before calculating its backoff period and before inserting a new entry into its ST, recalculates and updates the priority index of each entry in its ST. When a node hears a DATA packet, if an entry for the corresponding packet exists in its ST, then that entry is deleted. The sender node deletes its entry from the ST only when an ACK for the transmitted DATA packet is received. It may happen that a DATA packet transmitted is not heard by a node which had previously been located within the transmission range of a sender node holding the highest priority packet in its locality. This might be because of reasons such as node mobility and channel errors. In such cases, the stale entries might affect the desired scheduling of packets. Another reason for stale entries in the ST is that, when the network load is high, some of the packets would miss their deadline targets while waiting in the node's queue itself. Such packets will never be transmitted. In order to remove stale entries, whenever table updates are performed, entries whose deadline targets have been missed already are deleted from the ST. The objective of the back-off mechanism used in DLPS is to reflect the priority of the node's highest priority packet on the back-off period to be taken by the node. If *r* is the rank (rank of an entry is the position of that entry in the scheduling table of the node), in ST of the node, of the current packet

to be sent, *n* is the number of retransmission attempts made for the packet, and *nmax* is the maximum number of retransmission attempts permitted, then the back-off interval is given by **(2.9.3) (2.9.4) (2.9.5)** where CW_{min} is the minimum size of the contention window, and *M* is the desired packet delivery ratio. This means that if the packet has the highest rank in the broadcast region of the node, then it has the lowest back-off period according to Equation 2.9.3 and faces much less contention. Else, if it is the first time the packet is being transmitted, then the back-off distribution follows the second scheme as in Equation 2.9.4, where the back-off is more than that for the first case. Here the current *PDR* of the flow affects the back-off period. If *PDR* is considerably less, then the first term would be less, and if it is high, then the first term would be high and the node would have to wait for a longer time. Finally, if the packet does not fit into these two categories, then the backoff value is as per the third scheme in Equation 2.9.5, and is the longest of the three cases. The higher the value of *ULB,* the longer the back-off period. DLPS delivers a higher percentage of packets within their delay targets and has lower average end-to-end delay in packet delivery when compared to the 802.11 DCF and the DPS schemes.

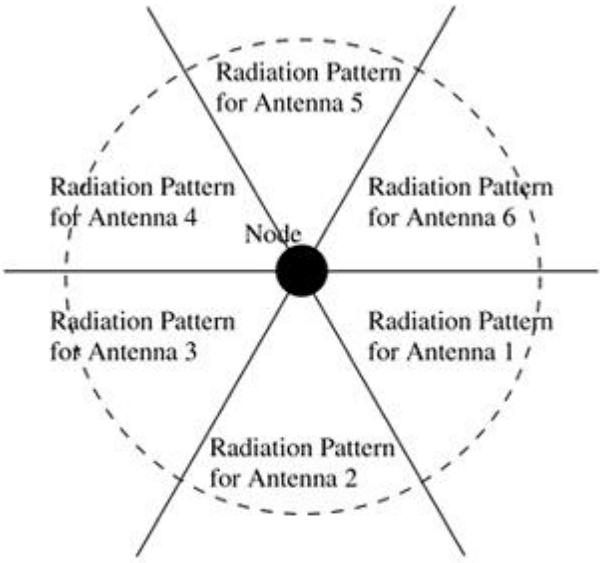
2.10 MAC PROTOCOLS THAT USE DIRECTIONAL ANTENNAS

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

2.10.1MAC Protocol Using Directional Antennas

The MAC protocol for mobile ad hoc networks using directional antennas that was proposed in makes use of directional antennas to improve the throughput in ad hoc wireless networks. The mobile nodes do not have any location information by means of which the direction of the receiver and sender nodes could be determined. The protocol makes use of an RTS/CTS exchange mechanism, which is similar to the one used in MACA . The nodes use directional antennas for transmitting and receiving data packets, thereby reducing their interference to other neighbor nodes. This leads to an increase in the throughput of the system. Each node is assumed to have only one radio transceiver, which can transmit and receive only one packet at any given time. The transceiver is assumed to be equipped with *M* directional antennas, each antenna having a conical radiation pattern, spanning an angle of radians (Figure 2.30). It is assumed that the transmissions by adjacent antennas never overlap, that is, the complete attenuation of the transmitted signal occurs outside the conical pattern of the directional antenna. The MAC protocol is assumed to be able to switch every antenna individually or all the antennas together to the *active* or *passive* modes. The radio transceiver uses only the antennas that are in the active mode. If a node transmits when all its antennas are active, then the transmission's radiation pattern is similar to that of an omnidirectional antenna. The receiver node uses receiver diversity while receiving on all antennas. This means that the receiver node uses the signal from the

antenna which receives the incoming signal at maximum power. In the normal case, this selected antenna would be the one whose conical pattern is directed toward the source node whose signal it is receiving. It is assumed that the radio range is the same for all directional antennas of the nodes. In order to detect the presence of a signal, a threshold signal power value is used. A node concludes that the channel is active only if the received signal strength is higher than this threshold value. **Figure 2.30. Radiation patterns of directional antennas.**

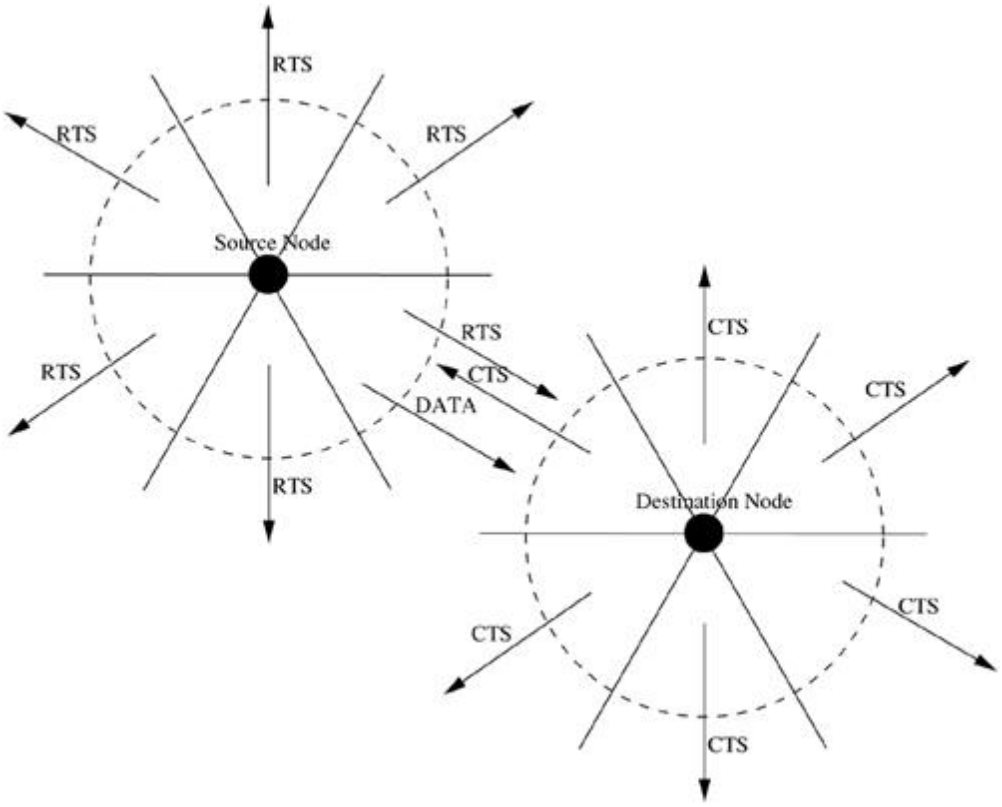


Node with 6 antennas (M=6)

The protocol works as follows. The packet-exchange mechanism followed for transmitting each data packet is depicted in Figure 2.31. In the example shown in the figure, each node is assumed to have six directional antennas. The main concept in this protocol is the mechanism used by the transmitting and receiving nodes to determine the directions of each other. The MAC layer at the source node must be able to find the direction of the intended next-hop receiver node so that the data packet could be transmitted through a directional antenna. It is the same case with the receiver. The receiver node must be able to determine the direction of the sender node before starting to receive data packets. This is performed in the following manner. An idle node is assumed to be listening to the on-going transmissions on all its antennas. The sender node first transmits an RTS packet addressed to the receiver. This RTS is transmitted through all the antennas of the node (omnidirectional transmission). The intended receiver node, on receiving this RTS packet, responds by transmitting a CTSpacket, again on all its antennas (omnidirectional transmission). The receiver node also notes down the direction of the sender by identifying the antenna that received the RTS packet with maximum power. The source, on receiving the CTS packet, determines the direction of the receiver node in a similar manner. The neighbor nodes that receive the RTS or CTS packets defer their transmissions for appropriate periods of time. After receiving the CTS, the source node transmits the next data packet through

the chosen directional antenna. All other antennas are switched off and remain idle. The receiver node receives this data packet only through its selected antenna.

Figure 2.31. Packet transmission.



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

2.10.2 Directional Busy Tone-Based MACProtocol

The directional busy tone-based MAC protocol adapts the DBTMA protocol for use with directional antennas. It uses directional antennas for transmitting the RTS, CTS, and data frames, as well as the busy tones. By doing so, collisions are reduced significantly. Also, spatial reuse of the channel improves, thereby increasing the capacity of the channel. Each node has a directional antenna which consists of *N* antenna elements, each covering a fixed sector spanning an angle of (360/*N*) degrees. For a unicast transmission, only a single antenna element is used. For broadcast transmission, all the *N* antenna elements transmit simultaneously. When a node is idle (not transmitting packets), all antenna elements of the node keep sensing the channel. The node is assumed to be capable of identifying the antenna element on which the incoming signal is received with maximum power. Therefore, while receiving, exactly one antenna element collects the signals. In an ad hoc wireless network, nodes may be mobile most of the time. It is assumed

that the orientation of sectors of each antenna element remains fixed. The protocol uses the same two busy tones BT_t and BT_r used in the DBTMA protocol. The purpose of the busy tones is the same. Before transmitting anRTS packet, the sender makes sure that the BT_r tone is not active in its neighborhood, so that its transmissions do not interfere with packets being received at a neighboring receiver node. Similarly, a receiver node, before transmitting a CTS, verifies that a BT_t is not active in its neighborhood. This is done to make sure that the data the node is expected to receive does not collide with any other on-going transmission. The modified directional DBTMA protocol operates as follows. A node that receives a data packet for transmission first transmits an RTS destined to the intended receiver in all directions (omnidirectional transmission). On receiving this RTS, the receiver node determines the antenna element on which the RTS is received with maximum gain. The node then sends back a directional CTS to the source using the selected antenna element (which points toward the direction of the sender). It also turns on the busy tone BT_r in the direction toward the sender. On receiving the CTS packet, the sender node turns on theBT_t busy tone in the direction of the receiver node. It then starts transmitting the data packet through the antenna element on which the previous CTS packet was received with maximum gain. Once the packet transmission is over, it turns off the BT_t signal. The receiver node, after receiving the data packet, turns off the BT_r signal. The directional busy tones can permit simultaneous transmissions in the neighborhood of a transmitting or a receiving node. For example, in Figure 2.32 (a), where omnidirectional busy tones are being used, when a transmission is going on from node A to node B, node D is not permitted to receive any data as it hears the BT_t tone transmitted by node A. But when directional busy tone transmissions are used (Figure 2.32 (b)), it can be seen that node D can simultaneously receive data from node C. Another example is shown in Figure 2.33. When omnidirectional busy tones are used (Figure 2.33 (a)), node C is not permitted to transmit to node D when node B is receiving data from node A. But when directional busy tones are used (Figure 2.33 (b)), node C does not receive any BT_r tone from node B and so it is free to transmit to node D even while the transmission between nodes A and B is going on. **Figure 2.32. Directional DBTMA: Example 1.**

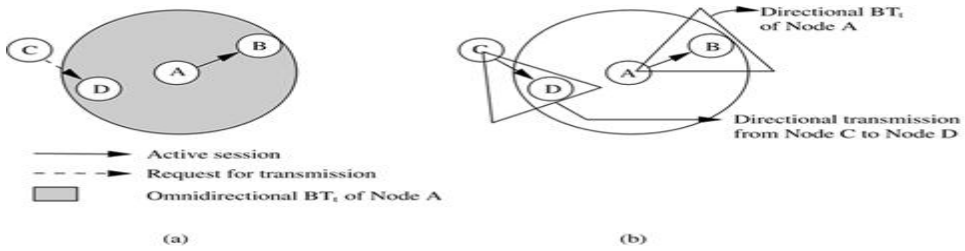
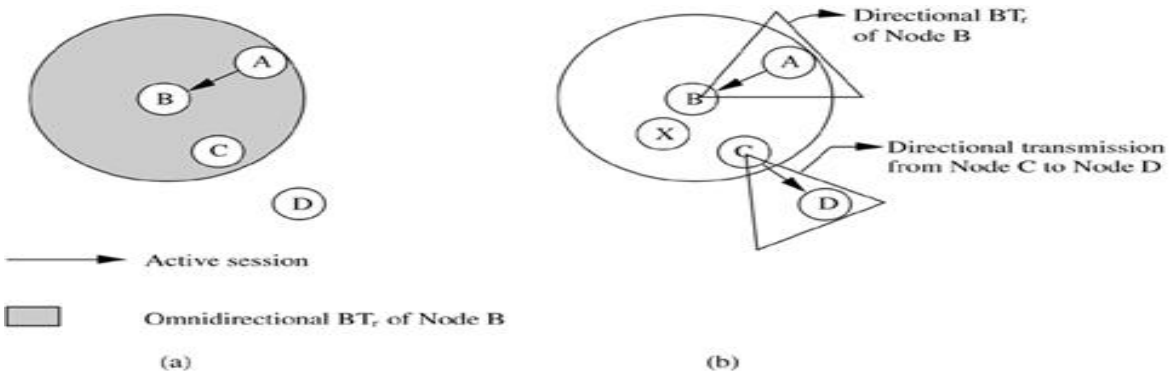


Figure 2.33. Directional DBTMA: Example 2.



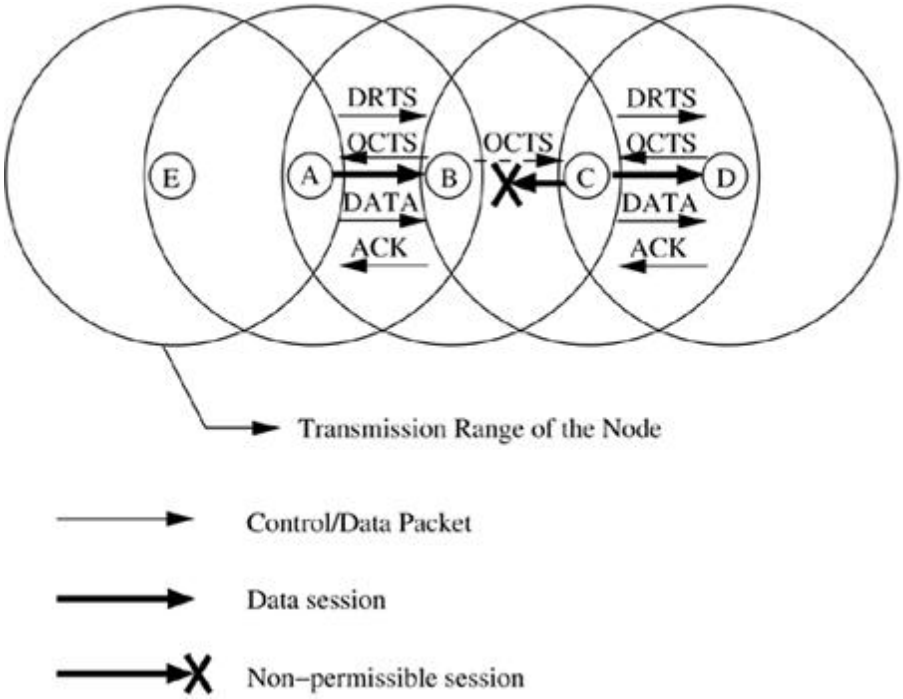
But this scheme may cause collisions in certain scenarios. For example, in Figure 2.33 (b), node C is free to transmit to node X when node B is receiving packets from node A. The packets transmitted to node X may collide with those being received at node B. Therefore, this protocol is not guaranteed to be collision-free. But the overall performance of the protocol (which uses directional busy tone transmissions) is better than that of the DBTMAprotocol. In the case of vehicle-mounted nodes, the basic assumption that the orientation of sectors of each antenna element remains fixed may not be valid.

2.10.3 Directional MAC Protocols for Ad Hoc Wireless Networks

Two MAC schemes using directional antennas are proposed in. It is assumed that each node knows about the location of its neighbors as well as its own location. The physical location information can be obtained by a node using the global positioning system (GPS). In the IEEE 802.11 DCF scheme, a node that is aware of a nearby on-going transmission will not participate in a transmission itself. In the directional MAC (D-MAC) protocols proposed in, a similar logic is applied on a per-antenna basis. If a node has received an RTS or CTS packet related to an on-going transmission on a particular antenna, then that particular antenna is not used by the node till the other transmission is completed. This antenna stays blocked for the duration of that transmission. The key concept here is, though a particular antenna of a node may remain blocked, the remaining antennas of the node can be used for transmissions. This improves the throughput of the system. An omnidirectional transmission is possible only if none of the antennas of the node is blocked. In the first directional MAC scheme (DMAC-1), a directional antenna is used for transmittingRTS packets. CTS packet transmissions are omnidirectional. Consider Figure 2.34. Here node A, which needs to transmit a packet to node B, first transmits a directional RTS (DRTS) packet to node B. Node B, on receiving this packet, responds by transmitting an omnidirectional CTS (OCTS) packet. Once the OCTS is received without any error by node A, node A sends a data packet using a directional antenna. When node B receives the data packet, it immediately transmits a directional ACK (DACK) packet. Node C would receive theOCTS packet from node B. At node C, only the directional antenna pointing toward node B would be blocked due to this. Node C can freely transmit to node D using another directional antenna. Thus it can be seen that in DMAC-1, usage of directional antennas improves

the performance by allowing simultaneous transmissions, which are not permitted when only omnidirectional antennas are used.

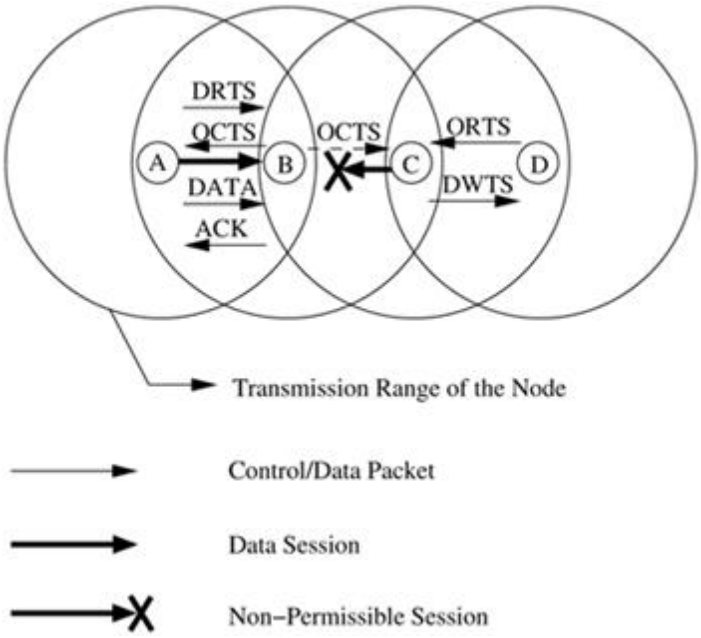
Figure 2.34. Operation of DMAC protocol.



In the second directional MAC scheme (DMAC-2) proposed in [23], both directional RTS(DRTS) as well as omnidirectional RTS (ORTS) transmissions are used. In DMAC-1, the usage of DRTS may increase the probability of control packet collisions. For example, consider Figure 2.34. Here node A initiates a DRTS transmission to node B. The DRTS packet is not heard by node E, and so it is not aware of the transmission between nodes A and B. Suppose node E sends a packet to node A, then that packet may collide with the OCTS orDACK packets transmitted by node B. The probability of control packet collisions is reduced in DMAC-2. In DMAC-2, a node that wants to initiate a data transfer may send an ORTS or aDRTS as per the following two rules. (1) If none of the directional antennas at the node are blocked, then the node sends an ORTS packet. (2) Otherwise, the node sends a DRTS packet, provided the desired directional antenna is not blocked. Consider the same example in Figure 2.34. Here when node A initiates data transfer to node B, assuming all its antennas are not blocked, it sends an ORTS packet to node B. Node E would now receive this packet, and the antenna on which the ORTS packet was received would remain blocked for the duration of the transmission from node A to node B. If node E wants to send a packet to node A, it needs to wait for the duration of the transmission between nodes A and B, so that its directional antenna pointing toward node A becomes unblocked; only then can it start transmitting packets to node A. Thus, the combination of ORTS and DRTS packets in DMAC-2 reduces collisions between control packets. Consider Figure 2.35. Node B sends an OCTS packet to node A after receiving DRTS from node A. Node

C would be aware of the transmission and its antenna pointing toward node B would remain blocked for the duration of the transmission. Now suppose node D sends anORTS to node C. Since one of node C's antennas is blocked currently, it would not respond to the ORTS packet. This results in node D timing out and unnecessary retransmissions ofORTS packets to node C. To avoid this situation, another packet called directional wait-tosend (DWTS) is used. On receiving the ORTS from node D, node C transmits the DWTSpacket using a directional antenna toward node D. This DWTS packet carries the expected duration of the on-going transmission between nodes A and B. Node D, on receiving this packet, waits for the specified interval of time and then tries again.

Figure 2.35. Operation of DMAC protocol.



By enabling simultaneous collision-free transmissions, the directional MAC schemes proposed in [23] improve channel reuse, thereby increasing the capacity of the channel. This leads to a significant increase in the throughput of the system.

2.11 OTHER MAC PROTOCOLS

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

2.11.1 Multichannel MAC Protocol

The multichannel MAC protocol (MMAC) uses multiple channels for data transmission. There is no dedicated control channel. *N* channels that have enough spectral separation between each other are available for data transmission. Each node maintains a data structure called *PreferableChannelList* (PCL). The usage of the channels within the transmission range of the node is maintained in the PCL. Based on their usage, channels can be classified into three types.

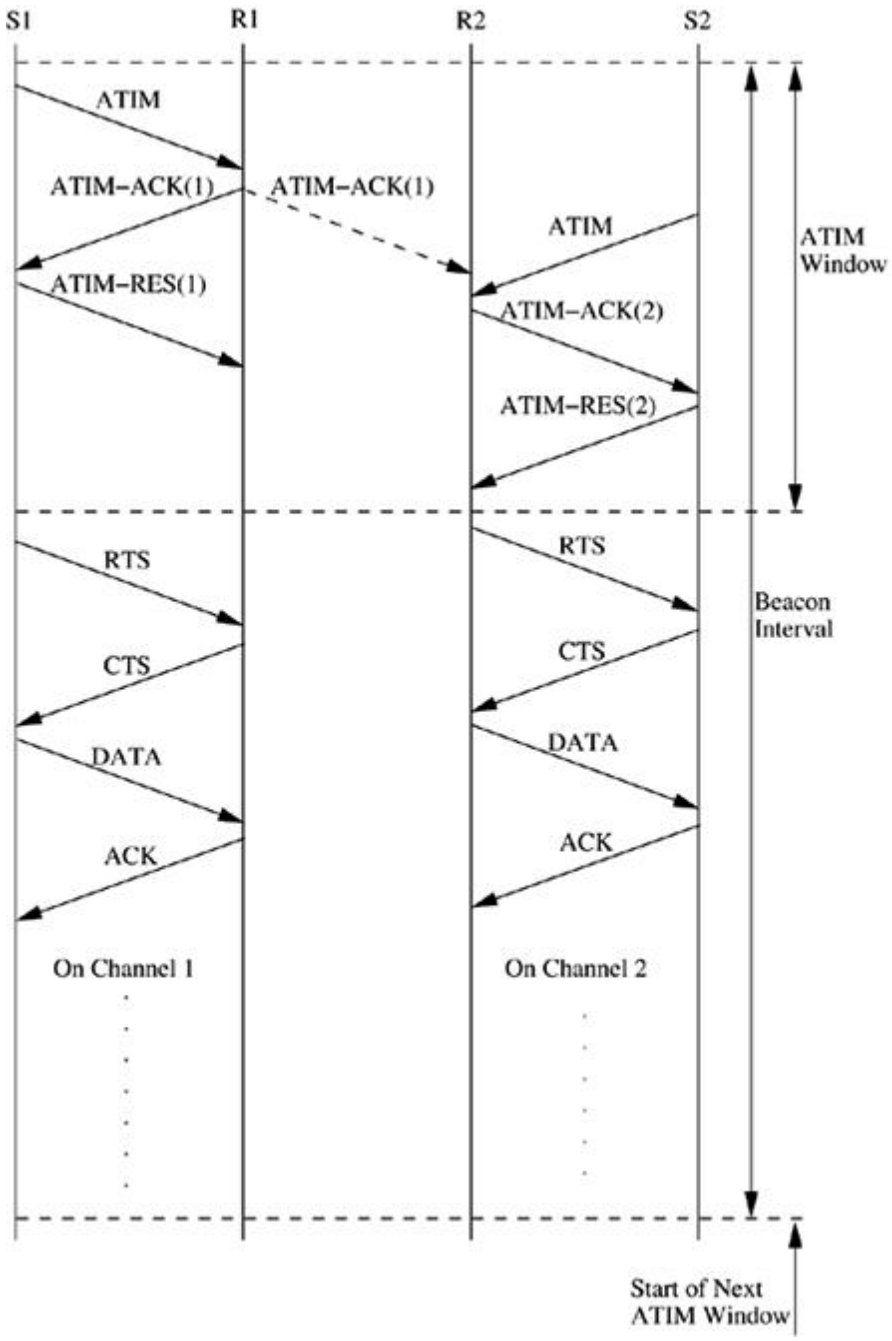
• High preference channel (HIGH): The channel has been selected by the current node and is being used by the node in the current beacon interval (beacon interval mechanism will be explained later). Since a node has only one transceiver, there can be only one HIGH channel at a time.

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

• Low preference channel (LOW): Such a channel is already being used in the transmission range of the node by other neighboring nodes. A counter is associated with each LOW state channel. For each LOW state channel, the count of source-destination pairs which have chosen the channel for data transmission in the current beacon interval is maintained. Time is divided into beacon intervals and every node is synchronized by periodic beacon transmissions. So, for every node, the beacon interval starts and ends almost at the same time. At the start of every beacon interval, there exists a time interval called the ad hoc traffic indication messages (ATIM) window. This window is used by the nodes to negotiate for channels for transmission during the current beacon interval. ATIM messages such as ATIM,ATIM-ACK (ATIM-acknowledgment), and ATIM-RES (ATIM-reservation) are used for this negotiation. The exchange of ATIM messages takes place on a particular channel called the*default channel*. The default channel is one of the multiple available channels. This channel is used for sending DATA packets outside the ATIM window, like any other channel. A node that wants to transmit in the current beacon interval sends an ATIM packet to the intended destination node. The ATIM message carries the PCL of the transmitting node. The destination node, upon receiving the packet, uses the PCL carried on the packet and its ownPCL to select a channel. It includes this channel information in the ATIM-ACK packet it sends to the source node. The source node, on receiving the ATIM-ACK packet, determines whether it can transmit on the channel indicated in the ATIM-ACK message. If so, it responds by sending the destination node an ATIM-RES packet. The ATIM-ACK and ATIM-RES packets are also used to notify the neighbor nodes of the receiver and sender nodes, respectively, about the channel that is going to be used for transmission in the current beacon interval. The nodes that hear these packets update their PCLs accordingly. At the end of theATIM window, the source and destination nodes switch to the agreed-upon channel and start communicating by exchanging RTS/CTS control packets. If the source node is not able to use the channel selected by the destination, it cannot transmit packets to that destination in the current beacon interval. It has to wait for the next beacon interval for again negotiating channels. The ATIM packets themselves may be lost due to collisions; in order to prevent this, each node waits for a randomly chosen back-off period (between 0 and CW_{min}) before transmitting the ATIM packet. Operation of the MMAC protocol is illustrated in Figure 2.36. At the beginning of the beacon interval, source node S1 sends an ATIM message to receiver R1. Receiver R1 responds by sending an ATIM-ACK packet (ATIM-ACK(1)) carrying the ID 1 of the channel it prefers (inFigure 2.36 the number within parentheses indicates the ID of the preferred channel).

Node S1, on receiving this packet, confirms the reservation by sending an ATIM-RES packet (ATIMRES(1)) for channel 1. The ATIM-ACK(1) packet sent by receiver R1 is also overheard by node R2. When node R2 receives an ATIM packet from source S2, it chooses a different channel with ID 2, and sends the channel information to source S2 through the ATIM-ACK packet (ATIMACK(2)). Since channel 2 is agreeable to node S2, it responds by sending theATIM-RES(2) packet, and the reservation gets established. Once the ATIM window finishes, the data transmission (through RTS-CTS-DATA-ACK packet exchange) between node pairs S1-R1 and S2-R2 takes place on the corresponding reserved channels, channel 1 and channel 2, respectively.

Figure 2.36. Operation of MMAC protocol.



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

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

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

• Else if there exists a HIGH state channel in the PCL of node S, then this channel is selected. • Else if there exists a common MID state channel in the PCLs of both node S and node R, then that channel is selected. If many such channels exists, one of them is selected randomly. • Else if there exists a channel which is in the MID state at only one of the two nodes, then that channel is chosen. If many such channels exist, one of them is selected randomly.

• If all channels in both PCLs are in the LOW state, the counters of the corresponding channels at nodes S and R are added, and the channel with the least count is selected. Ties are broken arbitrarily. MMAC uses simple hardware. It requires only a single transceiver. It does not have any dedicated control channel. The throughput of MMAC is higher than that of IEEE 802.11 when the network load is high. This higher throughput is in spite of the fact that in MMAC only a single transceiver is used at each node. Unlike other protocols, the packet size in MMAC need not be increased in order to take advantage of the presence of an increased number of channels.

2.11.2 Multichannel CSMAMAC Protocol

In the multichannel CSMA MAC protocol (MCSMA), the available bandwidth is divided into several channels. A node with a packet to be transmitted selects an idle channel randomly. The protocol also employs the notion of *soft* channel reservation, where preference is given to the channel that was used for the previous successful transmission. Though the principle used in MCSMA is similar to the frequency division multiple access (FDMA) schemes used in cellular networks, the major difference here is that there is no centralized infrastructure available, and channel assignment is done in a distributed fashion using carrier-sensing. The operation of the protocol is discussed below. The total available bandwidth is divided into N non-overlapping channels (N is independent of the number of hosts in the network), each having a bandwidth of (W/N), where W is the total bandwidth available for communication. The channels may be created in the frequency domain (FDMA) or in the code domain (CDMA). Since global synchronization between nodes is not available in ad hoc wireless networks, channel division in the time domain (TDMA) is not used. An idle node (which is not transmitting packets) continuously monitors all the N channels. A channel whose total received signal strength[1] (TRSS) is below the sensing threshold (ST) of the node is marked IDLE by the node. The time at which TRSS drops below ST is also noted for each IDLE channel. Such channels are put in the *free-channels* list. [1] The total received signal strength of a signal is calculated by the sum of contributions arising from the various individual multipath components of the signal. When an idle node receives a packet to be transmitted, it does the following. If the free-channels list is empty, it waits for any channel to become IDLE. It then waits for an additional long interframe

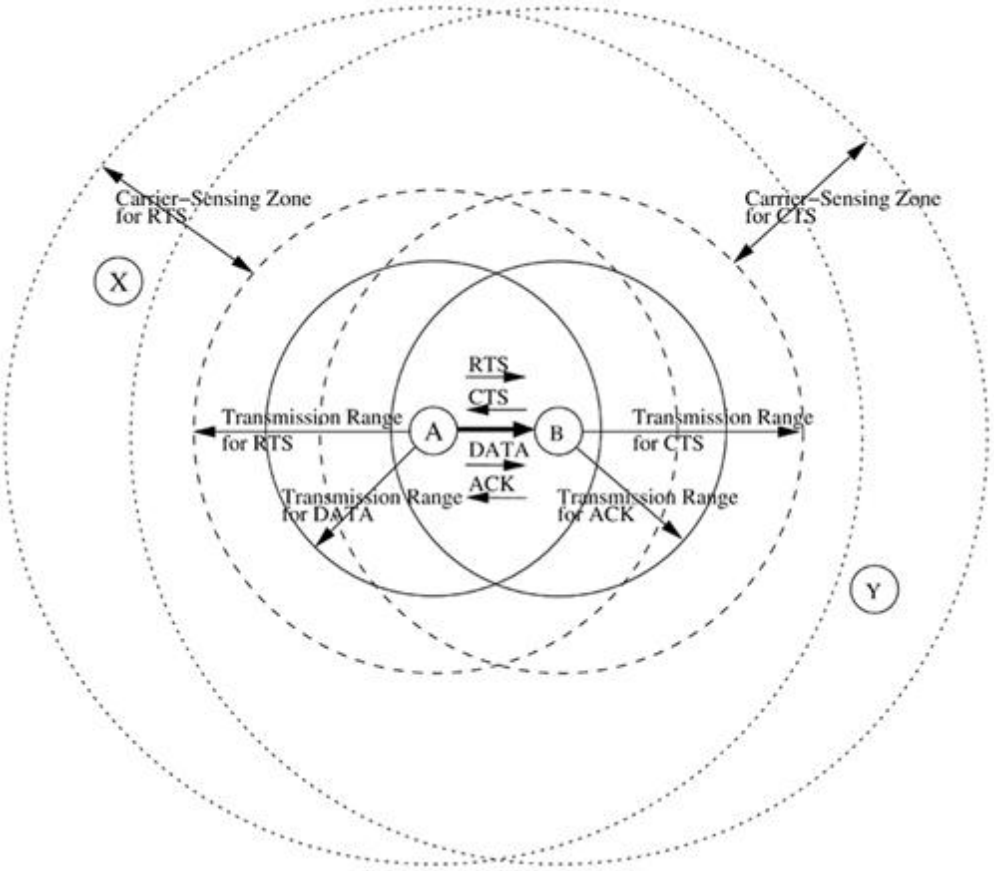
space (LIFS) time, and for another random access back-off period. If the channel remains idle for this entire wait period, then the node starts transmitting its packets on this channel. In case the free-channels list is non-empty, the node first checks whether the channel it used for its most recent successful transmission is included in the list. If so, the node uses this channel for its new transmission. Otherwise, one among the IDLE channels available in the free-channels list is randomly chosen (using a uniform random number generator). Before the actual packet transmission, the node checks the TRSS of the chosen channel. If it had remained below ST for at least LIFS period of time, then the node immediately initiates packet transmission. Otherwise, the node initiates back-off delay after the LIFS time period. During the back-off period, if the TRSS of the chosen channel goes above ST, then the back-off is immediately canceled. A new back-off delay is scheduled when the TRSS again goes below the ST. After successfully transmitting a packet (indicated by an acknowledgment from the receiver), the sender node notes the ID of the channel used. This channel would be given preference when a new channel is to be selected for its next transmission. When the number of channels N is sufficiently large, each node tends to *reserve* a channel for itself. This is because a node prefers the channel used in its last successful transmission for its next transmission also. This reduces the probability of two contending nodes choosing the same channel for transmission. Nodes are expected to dynamically select channels for transmissions in a mutually exclusive manner, so as to enable parallel interference-free transmissions. Even at high traffic loads, due to the tendency of every node to choose a*reserved* channel for itself, the chances of collisions are greatly reduced. The number of channels into which the available bandwidth is split is a very important factor affecting the performance of the protocol. If the number of channels is very large, then the protocol results in very high packet transmission times.

2.11.3 Power Control MAC Protocol for Ad Hoc Networks

The power control MAC protocol (PCM) allows nodes to vary their transmission power levels on a per-packet basis. The PCM protocol is based on the power control protocol used in , which is referred to as the *BASIC* protocol in this section. In what follows, the working of the BASIC power control protocol is briefly described. This is followed by a discussion of the PCM protocol. In the BASIC scheme, the RTS and CTS packets are transmitted with maximum power $p_{max.}$The RTS-CTS handshake is used for deciding upon the transmission power for the subsequent DATA and ACK packet transmissions. This can be done using two methods. In the first method, source node A transmits the RTS with maximum power $p_{max.}$ This RTS is received at the receiver with signal level p_r. The receiver node B can calculate the minimum required transmission power level $p_{desired}$ for the DATA packet, based on the received power level p_r , the transmitted power level p_{max} , and the noise level at receiver B. Node B then specifies this $p_{desired}$ in the CTS packet it transmits to node A. Node A transmits the DATA packet using power level $p_{desired}$. In the second method, when the receiver node B receives anRTS packet, it responds with a CTS packet at the usual maximum power level $p_{max.}$ When the source node receives this CTS packet, it calculates $p_{desired}$ based on the received power levelp_r and transmitted power level p_{max} as

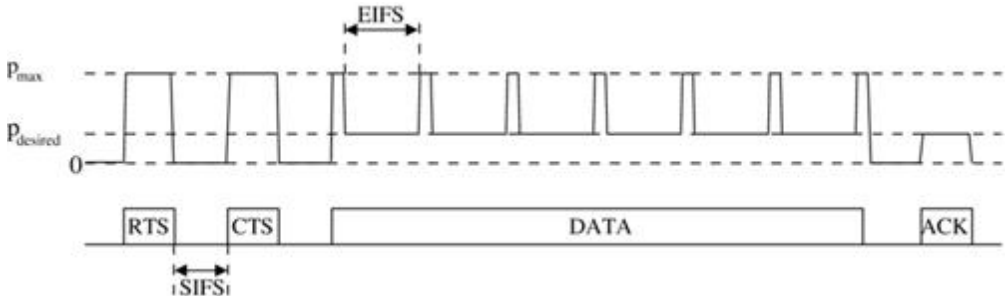
where Rx_{thresh} is the minimum necessary received signal strength and c is a constant. The source node uses power level $p_{desired}$ to transmit the DATA packet. Similarly, the receiver uses the signal power of the received RTS packet to determine the power level to be used, $p_{desired}$, for the ACK packet. This method assumes the attenuation between the sender and receiver nodes to be the same in both directions. It also assumes the noise level at the nodes to be below a certain predefined threshold value. Thus, the BASIC scheme uses maximum transmit power for RTS and CTS packets, and only necessary power levels for the DATA and ACK packets. But this scheme has a drawback. Consider Figure 2.37. Node A sends an RTS to node B, for which node B sends a CTS packet. Since these packets are sent at maximum power, nodes X and Y that are located in the carrier sensing zones of nodes A and B, respectively (when a node N1 is in the carrier-sensing zone of node N2, node N1 can sense the signal from node N2, but the received signal strength is not high enough to decode it correctly), defer their transmissions for a sufficient enough period of time [extended inter-frame space (EIFS) period of time] so as to not interfere with the RTSCTS exchange. But since the DATA and ACK transmissions use only the minimum necessary power, the DATA transmitted by node A cannot be sensed by node X, and the ACK packet transmitted by node B cannot be sensed by node Y. So if nodes X and Y transmit after the EIFS period (which is set in their NAVs on sensing the RTS or CTS packets), the packet transmitted by node X would collide at node A with the ACK packet from node B, and the packet transmitted by node Y would collide with the DATA packet at node B.

Figure 2.37. Packet transmission in BASIC scheme.



PCM modifies this scheme so as to minimize the probability of collisions. The source and receiver nodes transmit the RTS and CTS packets, as usual, with maximum power p_{max}. Nodes in the carrier-sensing zones of the source and receiver nodes set their NAVs for EIFSduration when they sense the signal but are not able to decode it. The source node generally transmits with minimum necessary power, as in the BASIC scheme. But, in order to avoid collisions with packets transmitted by the nodes in its carrier-sensing zone, the source node transmits the DATA packet at maximum power level p_{max} periodically. The duration of each such transmission must be larger than the time required for physical carrier-sensing. Since the nodes in the carriersensing zone defer their transmissions for EIFS duration if they are not able to decode the received signal, the transmit power for the DATA packet is increased (and brought down back to original level) every EIFS duration. The power level changes forRTS-CTS-DATA-ACK transmissions is depicted in Figure 2.38. Thus this protocol prevents collisions of ACK packets at the sender node.

Figure 2.38. Transmission power pattern in PCM.

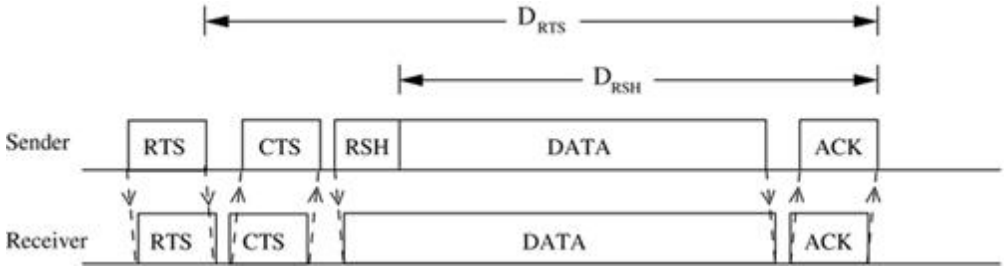


Hence with the above simple modification, the PCM protocol overcomes the problems faced in the BASIC scheme. PCM achieves throughput very close to that of the 802.11 protocol while using much less energy.

2.11.4 Receiver-Based Autorate Protocol The receiver-based autorate protocol (RBAR) [28] uses a novel rate adaptation approach. The rate adaptation mechanism is at the receiver node instead of being located at the sender. Rate adaptation is the process of dynamically switching data rates in order to match the channel conditions so that optimum throughput for the given channel conditions is achieved. Rate adaptation consists of two processes, namely, channel quality estimation and rate selection. The accuracy of the channel quality estimates significantly influences the effectiveness of the rate adaptation process. Therefore, it is important that the best available channel quality estimates are used for rate selection. Since it is the channel quality at the receiver node which determines whether a packet can be received or not, it can be concluded that the best channel quality estimates are available at the receiver. The estimates must be used as early as possible before they get stale. If the sender is to implement the rate adaptation process, significant delay would be involved in communicating the channel quality estimates from the receiver to the sender, which may result in the estimates becoming stale before being used. Therefore, the RBAR protocol advocates for rate adaptation at the receiver node rather than at the sender. Rate selection is done at the receiver on a per-packet basis during the RTS-CTS packet exchange. Since rate selection is done *during* the RTS-CTS exchange, the channel quality estimates are very close to the actual transmission times of the data packets. This improves the effectiveness of the rate selection process. The RTS and CTS packets carry the chosen modulation rate and the size of the data packet, instead of carrying the duration of the reservation. The packet transmission process is depicted in Figure 2.39. The sender node chooses a data rate based on some heuristic and inserts the chosen data rate and the size of the data packet into the RTS. When a neighbor node receives this RTS, it calculates the duration of the reservation D_{RTS} using the data rate and packet size carried on the RTS. The neighbor node then updates its NAV accordingly to reflect the reservation. While receiving the data packet, the receiver node generates an estimate of the channel conditions for the impending data transfer. Based on this estimate, it chooses an appropriate data rate. It stores the chosen data rate and the size of the packet on the CTS packet and transmits the CTS to the sender. Neighbor nodes receiving the CTS calculate the expected duration of the transmission and update their NAVs

accordingly. The source node, on receiving the CTSpacket, responds by transmitting the data packet at the rate chosen by the receiver node.

Figure 2.39. Packet transmission in RBAR.

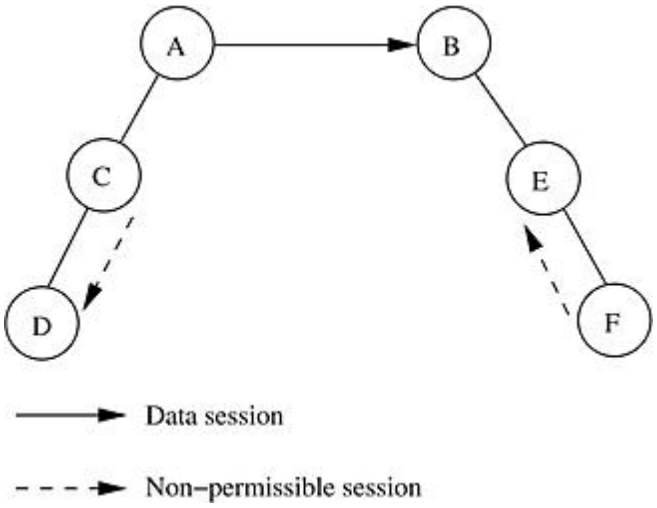


If the rates chosen by the sender and receiver are different, then the reservation duration D_{RTS} calculated by the neighbor nodes of the sender would not be valid. D_{RTS} time period, which is calculated based on the information carried initially by the RTS packet, is referred to as *tentative reservation.* In order to overcome this problem, the sender node sends the data packet with a special MAC header containing a *reservation subheader* (RSH). The RSHcontains a subset of header fields already present in the IEEE 802.11 data frame, along with a check sequence for protecting the subheader. The fields in the RSH contain control information for determining the duration of the transmission. A neighbor node with tentative reservation entries in its NAV, on hearing the data packet, calculates D_{RSH} , the new reservation period, and updates its NAV to account for the difference between D_{RTS} and $D_{RSH.}$ For the channel quality estimation and prediction algorithm, the receiver node uses a sample of the instantaneous received signal strength at the end of RTS reception. For the rate selection algorithm, a simple threshold-based technique is used. Here the rate is chosen by comparing the channel quality estimate [*e.g.*, signal to noise ratio (SNR)] against a series of thresholds representing the desired performance bounds (*e.g.*, a series of SNR thresholds). The modulation scheme with the highest data rate, satisfying the performance objective for the channel quality estimate, is chosen. RBAR employs an efficient quality estimation mechanism, which leads to a high overall system throughput. RBAR can be easily incorporated into many existing medium access control protocols.

2.11.5 Interleaved Carrier-Sense Multiple Access Protocol

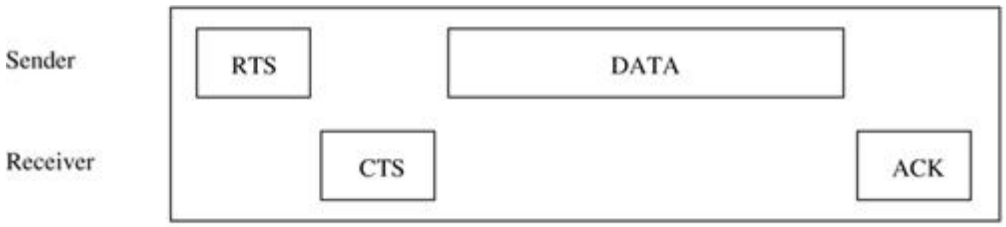
The interleaved carrier-sense multiple access protocol (ICSMA) efficiently overcomes the exposed terminal problem faced in ad hoc wireless networks. The inability of a source node to transmit, even though its transmission may not affect other ongoing transmissions, is referred to as the exposed terminal problem. For example, consider the topology shown inFigure 2.40. Here, when a transmission is going from node A to node B, nodes C and F would not be permitted to transmit to nodes D and E, respectively. Node C is called a sender-exposed node, and node E is called a receiver-exposed node. The exposed terminal problem reduces the bandwidth efficiency of the system.

Figure 2.40. Exposed terminal problem. ***Reproduced with permission from [29], © IEEE, 2003.***

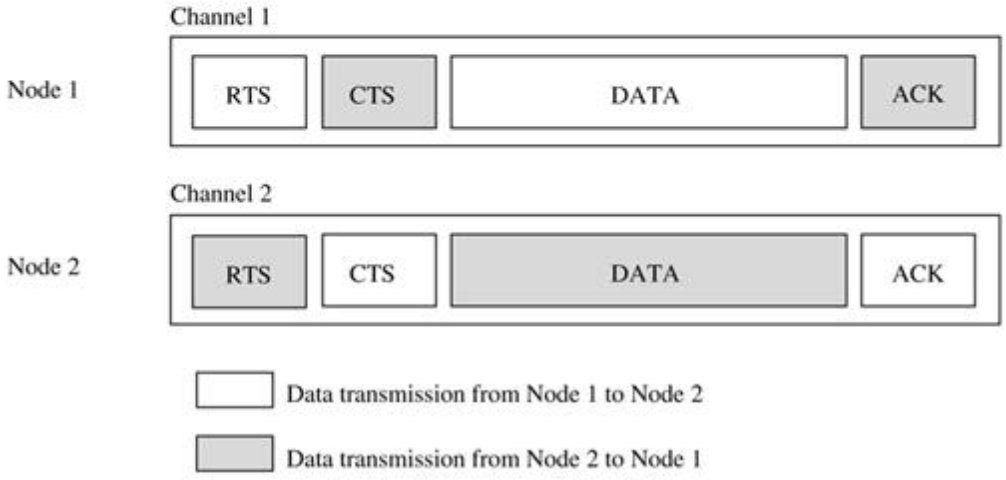


In ICSMA, the total available bandwidth is split into two equal channels (say, channel 1 and channel 2). The handshaking process is interleaved between the two channels, hence the name interleaved carrier-sense multiple access. The working of ICSMA is very simple. It uses the basic RTS-CTS-DATA-ACK exchange mechanism used in IEEE 802.11 DCF. If the source node transmits the RTS packet on channel 1, the receiver node, if it is ready to accept packets from the sender, responds by transmitting the CTS packet on channel 2. Each node maintains a data structure called extended network allocation vector (E-NAV), which is analogous to the network allocation vector (NAV) used in IEEE 802.11 DCF. On receiving an RTS packet, the receiver node checks its E-NAV and finds out whether free time slots are available. It sends the CTS only if free slots are available. The source node, on receiving this CTS, transmits the DATA packet on channel 1. The receiver acknowledges the reception of the DATA packet by transmitting the ACK on channel 2. The ICSMA channel access mechanism is illustrated inFigure 2.41. Figure 2.41 (a) shows the RTS-CTS-DATA-ACK exchange of 802.11 DCF. Figure 2.41 (b) shows simultaneous data packet transmissions between two nodes in ICSMA. After transmitting an RTS or a DATA packet on a channel, the sender node waits on the other channel for the CTS or ACK packet. If it does not receive any packet on the other channel, it assumes that the RTS or DATA packet it transmitted was lost, and retries again. Similarly at the receiver, after transmitting a CTS frame on one of the channels, the receiver node waits on the other channel for the DATA packet. If the DATA packet is not received within the timeout period, it retransmits the CTS packet.

Figure 2.41. Packet transmissions in (a) 802.11 DCF and (b) ICSMA. ***Reproduced with permission from [29], © IEEE, 2003.***



(a)



(b)

The performance improvement of ICSMA is attributed to the following facts: • Nodes that hear RTS in a particular channel (say, channel 1) and do not hear the corresponding CTS on the other channel (channel 2) conclude that they are only sender-exposed in channel 1. Therefore, if they have packets to send, they can use channel 1 to transmit RTS to other nodes. This would not have been possible in 802.11 DCF, where transmissions by a senderexposed node would have collided with the corresponding currently active sender node. • Nodes that hear only the CTS in a particular channel (say, channel 1) and had not heard the corresponding RTS on the other complementary channel (channel 2) realize that they are only receiver-exposed on channel 1 to the on-going transmission. If they receive any RTS on channel 2, they would not refrain from sending a CTS on channel 1 for the received RTS. This would also not have been possible in 802.11 DCF, where there would have been collision at the receiver of the on-going session between the CTS packet transmitted by this node and the DATA packets belonging to the on-going session. Also, if this CTS transmission is successful, then there might have been collisions between DATA packets belonging to the two sessions at both the receiver nodes. The E-NAV used in ICSMA is implemented as two linked lists of blocks, namely, the *SEL*ist and the *REL*ist. Each block in each linked list has a start time and an end time. A typical list looks like s_1, f_1; s_2, f_2; ...; s_k, f_k where S_i denotes the start time of the i_{th} block in the list and $f_i$denotes the finish time of the i_{th} block in the list. The *SEL*ist is used to determine if the node would be sender-exposed at any

given instant of time in the future. A node is predicted to be senderexposed at any time t if there is a block s_j, f_j in the *SEL*ist such that $s_j < t < f_j$. Similarly, the *REL*ist tells if the node would be receiver-exposed at any time in the future. A node is predicted to be receiver-exposed at any time t if there exists a block s_j, f_j in the *REL*ist such that $s_j < t < f_j$. The *SEL*ist and the *REL*ist are updated whenever the RTS and CTSpackets are received by the node. The *SEL*ist is modified when an RTS packet is received, and the *REL*ist is modified when a CTS packet is received by the node. The modification in the list might be adding a new block, modifying an existing block, or merging two or more existing blocks and modifying the resulting block. ICSMA is a simple two-channel MAC protocol for ad hoc wireless networks that reduces the number of exposed terminals and tries to maximize the number of simultaneous sessions.ICSMA was found to perform better than the 802.11 DCF protocol is terms of metrics such as throughput and channel access delay.

UNIT-III

ROUTING PROTOCOLS FOR AD HOC WIRELESS NETWORKS

3.1 INTRODUCTION

An ad hoc wireless network consists of a set of mobile nodes (hosts) that are connected by wireless links. The network topology (the physical connectivity of the communication network) in such a network may keep changing randomly. Routing protocols that find a path to be followed by data packets from a source node to a destination node used in traditional wired networks cannot be directly applied in ad hoc wireless networks due to their highly dynamic topology, absence of established infrastructure for centralized administration (*e.g.,* base stations or access points), bandwidth-constrained wireless links, and resource (energy)-constrained nodes. A variety of routing protocols for ad hoc wireless networks has been proposed in the recent past. This chapter first presents the issues involved in designing a routing protocol and then the different classifications of routing protocols for ad hoc wireless networks. It then discusses the working of several existing routing protocols with illustrations.

3.2 ISSUES IN DESIGNING A ROUTING PROTOCOL FOR AD HOC WIRELESS NETWORKS

The major challenges that a routing protocol designed for ad hoc wireless networks faces are mobility of nodes, resource constraints, error-prone channel state, and hidden and exposed terminal problems. A detailed discussion on each of the following is given below.

3.2.1 Mobility The network topology in an ad hoc wireless network is highly dynamic due to the movement of nodes, hence an on-going session suffers frequent path breaks. Disruption occurs either due to the movement of the intermediate nodes in the path or due to the movement of end nodes. Such

situations do not arise because of reliable links in wired networks where all the nodes are stationary. Even though the wired network protocols find alternate routes during path breaks, their convergence is very slow. Therefore, wired network routing protocols cannot be used in ad hoc wireless networks where the mobility of nodes results in frequently changing network topologies. Routing protocols for ad hoc wireless networks must be able to perform efficient and effective mobility management.

3.2.2 Bandwidth Constraint Abundant bandwidth is available in wired networks due to the advent of fiber optics and due to the exploitation of wavelength division multiplexing (WDM) technologies. But in a wireless network, the radio band is limited, and hence the data rates it can offer are much less than what a wired network can offer. This requires that the routing protocols use the bandwidth optimally by keeping the overhead as low as possible. The limited bandwidth availability also imposes a constraint on routing protocols in maintaining the topological information. Due to the frequent changes in topology, maintaining a consistent topological information at all the nodes involves more control overhead which, in turn, results in more bandwidth wastage. As efficient routing protocols in wired networks require the complete topology information, they may not be suitable for routing in the ad hoc wireless networking environment.

3.2.3 Error-Prone Shared Broadcast Radio Channel

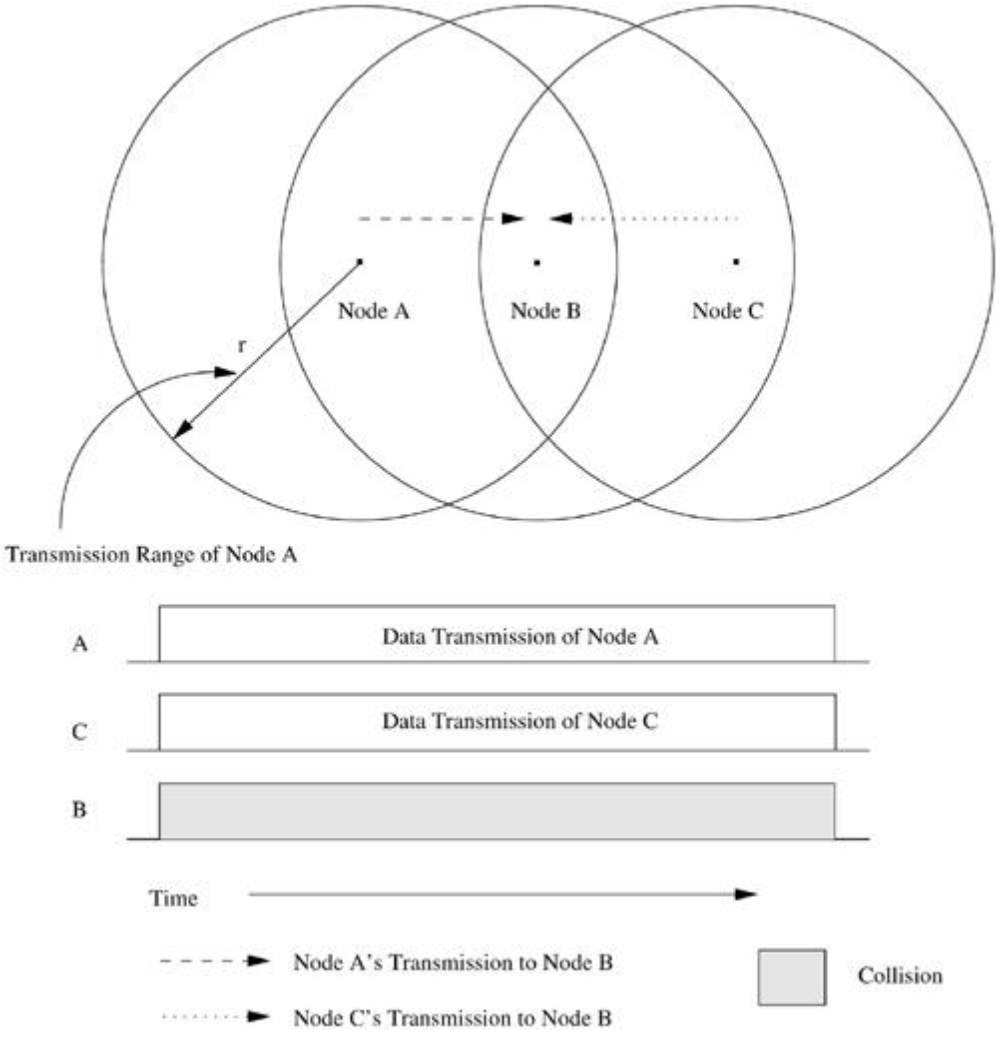
The broadcast nature of the radio channel poses a unique challenge in ad hoc wireless networks. The wireless links have time-varying characteristics in terms of link capacity and link-error probability. This requires that the ad hoc wireless network routing protocol interacts with the MAC layer to find alternate routes through better-quality links. Also, transmissions in ad hoc wireless networks result in collisions of data and control packets. This is attributed to the hidden terminal problem . Therefore, it is required that ad hoc wireless network routing protocols find paths with less congestion.

3.2.4 Hidden and Exposed Terminal Problems

The hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver. Collision occurs when both nodes transmit packets at the same time without knowing about the transmission of each other. For example, consider Figure 3.1. Here, if both node A and node C transmit to node B at the same time, their packets collide at node B. This is due to the fact that both nodes A and C are hidden from each other, as they are not within the direct transmission range of each other and hence do not know about the presence of each other. Solutions for this problem include medium access collision avoidance (MACA) , medium access collision avoidance for wireless (MACAW) , floor acquisition multiple access (FAMA) , and dual busy tone multiple access (DBTMA) . MACA requires that a transmitting node first explicitly notifies all potential hidden nodes about the forthcoming transmission by means of a two-way handshake control protocol called the RTS-CTS protocol exchange. Note that this may not solve the problem completely, but it reduces the probability of collisions. To increase the efficiency, an improved version of the MACA protocol known as MACAW has been proposed. This protocol requires that the receiver acknowledges each successful reception of a data packet. Hence, successful transmission is a fourway exchange mechanism, namely, RTS-CTS-Data-ACK. Even in the absence of bit errors and mobility, the RTS-CTS control packet exchange cannot ensure collision-free data transmission that has no interference from hidden terminals. One very important assumption made is that every node in the capture area of the receiver (transmitter) receives the CTS (RTS) cleanly. Nodes that do not hear either of these clearly can disrupt the successful transmission of the Data or the ACK packet. One particularly troublesome situation occurs when node A, hidden from the transmitter T and within the capture area of the receiver R, does not hear the CTS properly because it is within the capture area of node B that is transmitting and that is hidden from both R and T, as illustrated in Figure 3.2. In this case, node A did not successfully receive the CTS originated by node R and hence assumes that there is no on-going transmission in the neighborhood. Since

node A is hidden from node T, any attempt to originate its own RTS would result in collision of the on-going transmission between nodes T and R.

Figure 3.1. Hidden terminal problem.



The exposed terminal problem refers to the inability of a node which is blocked due to transmission by a nearby transmitting node to transmit to another node. Consider the example in Figure 3.3. Here, if a transmission from node B to another node A is already in progress, node C cannot transmit to node D, as it concludes that its neighbor, node B, is in transmitting mode and hence should not interfere with the on-going transmission. Thus, reusability of the radio spectrum is affected. For node C to transmit simultaneously when node B is transmitting, the transmitting frequency of node C must be different from its receiving frequency.

Figure 3.2. Hidden terminal problem with RTS-CTS-Data-ACK scheme.

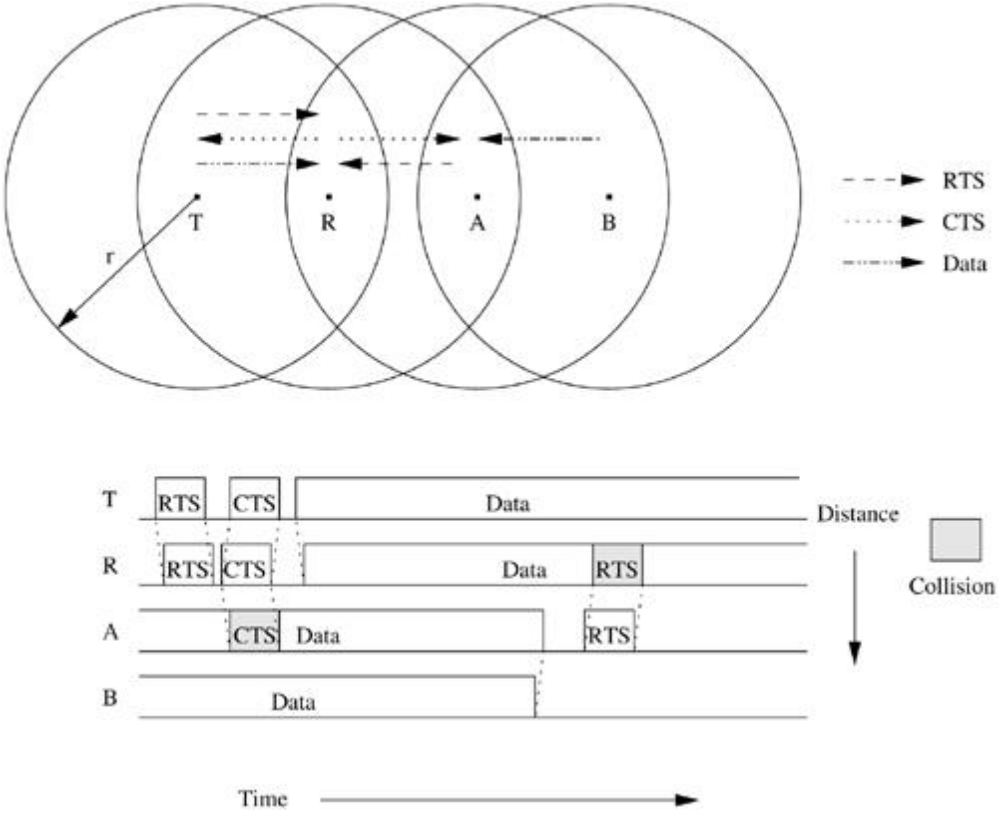
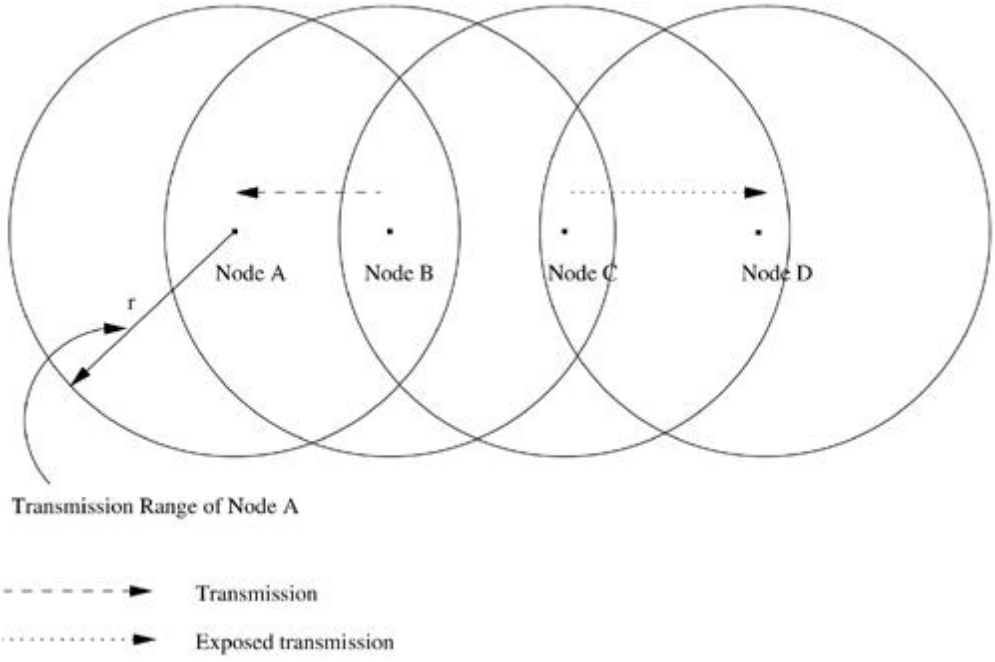


Figure 3.3. Exposed terminal problem.



3.2.5 Resource Constraints Two essential and limited resources that form the major constraint for the nodes in an ad hoc wireless network are battery life and processing power. Devices used in ad hoc wireless networks in most cases require portability, and hence they also have size and weight constraints along with the restrictions on the power source. Increasing the battery power and processing ability makes the nodes bulky and less portable. Thus ad hoc wireless network routing protocols must optimally manage these resources.

3.2.6 Characteristics of an Ideal Routing Protocol for Ad Hoc Wireless Networks

Due to the issues in an ad hoc wireless network environment discussed so far, wired network routing protocols cannot be used in ad hoc wireless networks. Hence ad hoc wireless networks require specialized routing protocols that address the challenges described above. A routing protocol for ad hoc wireless networks should have the following characteristics:

1. It must be fully distributed, as centralized routing involves high control overhead and hence is not scalable. Distributed routing is more fault tolerant than centralized routing, which involves the risk of single point of failure.

2. It must be adaptive to frequent topology changes caused by the mobility of nodes.

3. Route computation and maintenance must involve a minimum number of nodes. Each node in the network must have quick access to routes, that is, minimum connection setup time is desired.

4. It must be localized, as global state maintenance involves a huge state propagation control overhead.

5. It must be loop-free and free from stale routes.

6. The number of packet collisions must be kept to a minimum by limiting the number of broadcasts made by each node. The transmissions should be reliable to reduce message loss and to prevent the occurrence of stale routes. 7. It must

converge to optimal routes once the network topology becomes stable. The convergence must be quick.

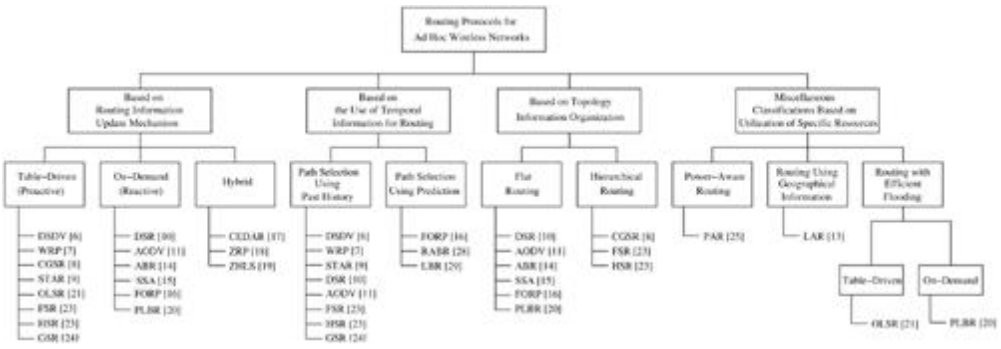
8. It must optimally use scarce resources such as bandwidth, computing power, memory, and battery power.

9. Every node in the network should try to store information regarding the stable local topology only. Frequent changes in local topology, and changes in the topology of parts of the network with which the node does not have any traffic correspondence, must not in any way affect the node, that is, changes in remote parts of the network must not cause updates in the topology information maintained by the node. 10. It should be able to provide a certain level of quality of service (QoS) as demanded by the applications, and should also offer support for time-sensitive traffic.

3.3 CLASSIFICATIONS OF ROUTING PROTOCOLS

Routing protocols for ad hoc wireless networks can be classified into several types based on different criteria. A classification tree is shown in Figure 3.4. Some of the classifications, their properties, and the basis of classifications are discussed below. The classification is not mutually exclusive and some protocols fall in more than one class. The deviation from the traditional routing metrics and path-finding processes that are employed in wired networks makes it worth further exploration in this direction. The routing protocols for ad hoc wireless networks can be broadly classified into four categories based on • Routing information update mechanism • Use of temporal information for routing • Routing topology • Utilization of specific resources

Figure 3.4. Classifications of routing protocols.



3.3.1 Based on the Routing Information Update Mechanism

Ad hoc wireless network routing protocols can be classified into three major categories based on the routing information update mechanism. They are:

1. **Proactive or table-driven routing protocols:** In table-driven routing protocols, every node maintains the network topology information in the form of routing tables by periodically exchanging routing information. Routing information is generally flooded in the whole network. Whenever a node requires a path to a destination, it runs an appropriate path-finding algorithm on the topology information it maintains.

2. **Reactive or on-demand routing protocols:** Protocols that fall under this category do not maintain the network topology information. They obtain the necessary path when it is required, by using a connection establishment process. Hence these protocols do not exchange routing information periodically.

3. **Hybrid routing protocols:** Protocols belonging to this category combine the best features of the above two categories. Nodes within a certain distance from the node concerned, or within a particular geographical region, are said to be within the routing zone of the given node. For routing within this zone, a table-driven approach is used. For nodes that are located beyond this zone, an on-demand approach is used.

3.3.2 Based on the Use of Temporal Information for Routing

This classification of routing protocols is based on the use of temporal information used for routing. Since ad hoc wireless networks are highly dynamic and path breaks are much more frequent than in wired networks, the use of temporal information regarding the lifetime of the wireless links and the lifetime of the paths selected assumes significance. The protocols that fall under this category can be further classified into two types:

1. **Routing protocols using past temporal information:** These routing protocols use information about the past status of the links or the status of links at the time of routing to make routing decisions. For example, the routing metric based on the availability of wireless links (which is the current/present

information here) along with a shortest path-finding algorithm, provides a path that may be efficient and stable at the time of path-finding. The topological changes may immediately break the path, making the path undergo a resource-wise expensive path reconfiguration process.

2. **Routing protocols that use future temporal information:** Protocols belonging to this category use information about the expected future status of the wireless links to make approximate routing decisions. Apart from the lifetime of wireless links, the future status information also includes information regarding the lifetime of the node (which is based on the remaining battery charge and discharge rate of the non-replenishable resources), prediction of location, and prediction of link availability.

3.3.3 Based on the Routing Topology

Routing topology being used in the Internet is hierarchical in order to reduce the state information maintained at the core routers. Ad hoc wireless networks, due to their relatively smaller number of nodes, can make use of either a flat topology or a hierarchical topology for routing.

1. **Flat topology routing protocols:** Protocols that fall under this category make use of a flat addressing scheme similar to the one used in IEEE 802.3 LANs. It assumes the presence of a globally unique (or at least unique to the connected part of the network) addressing mechanism for nodes in an ad hoc wireless network.

2. **Hierarchical topology routing protocols:** Protocols belonging to this category make use of a logical hierarchy in the network and an associated addressing scheme. The hierarchy could be based on geographical information or it could be based on hop distance.

3.3.4 Based on the Utilization of Specific Resources

1. **Power-aware routing:** This category of routing protocols aims at minimizing the consumption of a very important resource in the ad hoc wireless networks: the battery power. The routing decisions are based on minimizing the power consumption either locally or globally in the network. 2. **Geographical**

information assisted routing: Protocols belonging to this category improve the performance of routing and reduce the control overhead by effectively utilizing the geographical information available. The following section further explores the above classifications and discusses specific routing protocols belonging to each category in detail.

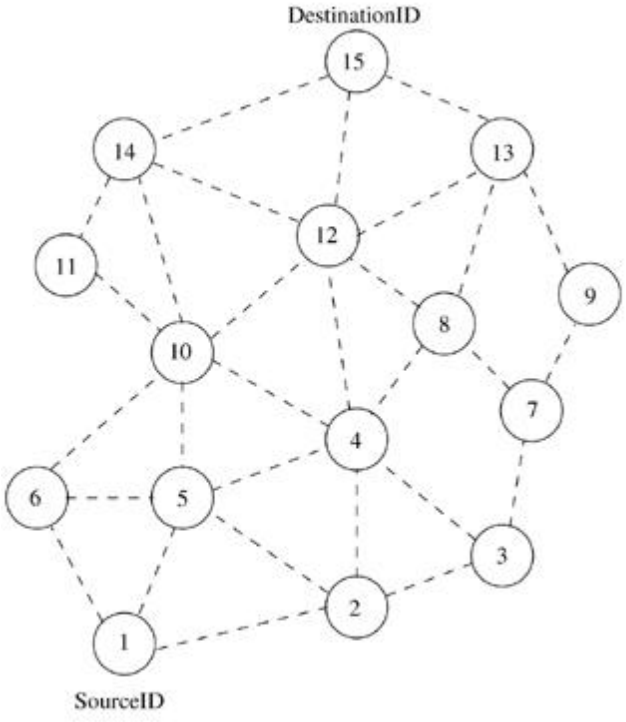
3.4 TABLE-DRIVEN ROUTING PROTOCOLS

These protocols are extensions of the wired network routing protocols. They maintain the global topology information in the form of tables at every node. These tables are updated frequently in order to maintain consistent and accurate network state information. The destination sequenced distance-vector routing protocol (DSDV), wireless routing protocol (WRP), source-tree adaptive routing protocol (STAR), and cluster-head gateway switch routing protocol (CGSR) are some examples for the protocols that belong to this category.

3.4.1 Destination Sequenced Distance-Vector Routing Protocol The destination sequenced distance-vector routing protocol (DSDV) is one of the first protocols proposed for ad hoc wireless networks. It is an enhanced version of the distributed Bellman-Ford algorithm where each node maintains a table that contains the shortest distance and the first node on the shortest path to every other node in the network. It incorporates table updates with increasing sequence number tags to prevent loops, to counter the count-to-infinity problem, and for faster convergence. As it is a table-driven routing protocol, routes to all destinations are readily available at every node at all times. The tables are exchanged between neighbors at regular intervals to keep an up-to-date view of the network topology. The tables are also forwarded if a node observes a significant change in local topology. The table updates are of two types: incremental updates and full dumps. An incremental update takes a single network data packet unit (NDPU), while a full dump may take multiple NDPUs. Incremental updates are used when a node does not observe significant changes in the local topology. A full dump is done either when the local topology changes significantly or when an incremental update requires more than a single NDPU. Table updates are initiated by a destination with a new sequence number which is always greater than the previous one. Upon receiving an updated table,

a node either updates its tables based on the received information or holds it for some time to select the best metric (which may be the lowest number of hops) received from multiple versions of the same update table from different neighboring nodes. Based on the sequence number of the table update, it may forward or reject the table. Consider the example as shown in Figure 3.5 (a). Here node 1 is the source node and node 15 is the destination. As all the nodes maintain global topology information, the route is already available as shown in Figure 3.5 (b). Here the routing table of node 1 indicates that the shortest route to the destination node (node 15) is available through node 5 and the distance to it is 4 hops, as depicted in Figure 3.5 (b).

Figure 3.5. Route establishment in DSDV.



(a) Topology graph of the network

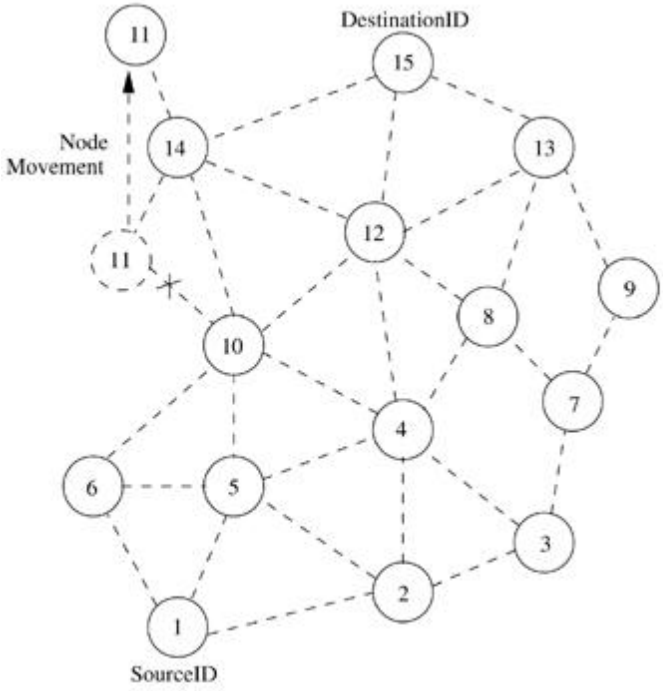
Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	6	3	176
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

(b) Routing table for Node 1

The reconfiguration of a path used by an on-going data transfer session is handled by the protocol in the following way. The end node of the broken link initiates a table update message with the broken link's weight assigned to infinity (∞) and with a sequence number greater than the stored

sequence number for that destination. Each node, upon receiving an update with weight ∞, quickly disseminates it to its neighbors in order to propagate the broken-link information to the whole network. Thus a single link break leads to the propagation of table update information to the whole network. A node always assigns an odd sequence number to the link break update to differentiate it from the even sequence number generated by the destination. Consider the case when node 11 moves from its current position, as shown in Figure 3.6. When a neighbor node perceives the link break, it sets all the paths passing through the broken link with distance as ∞. For example, when node 10 knows about the link break, it sets the path to node 11 as ∞ and broadcasts its routing table to its neighbors. Those neighbors detecting significant changes in their routing tables rebroadcast it to their neighbors. In this way, the broken link information propagates throughout the network. Node 1 also sets the distance to node 11 as ∞. When node 14 receives a table update message from node 11, it informs the neighbors about the shortest distance to node 11. This information is also propagated throughout the network. All nodes receiving the new update message with the higher sequence number set the new distance to node 11 in their corresponding tables. The updated table at node 1 is shown in Figure 3.6, where the current distance from node 1 to node 11 has increased from three to four hops.

Figure 3.6. Route maintenance in DSDV.



Routing Table for Node 1

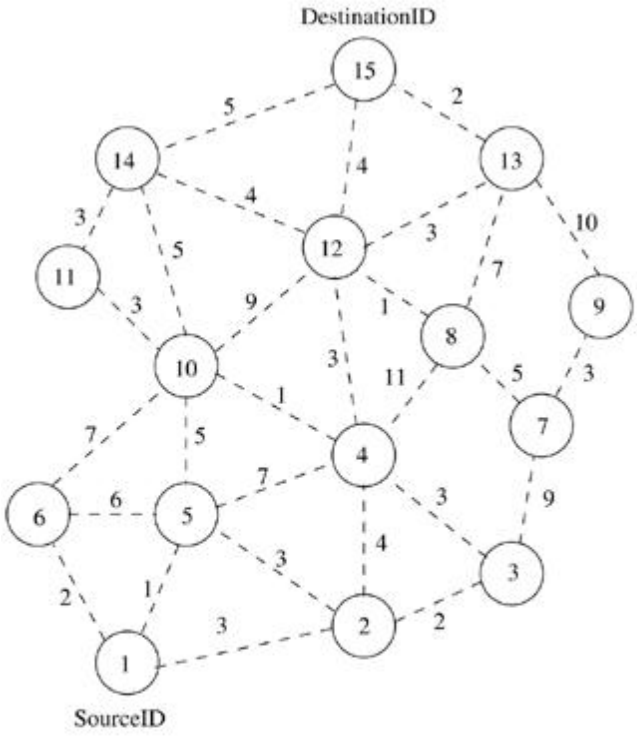
Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	5	4	180
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

Advantages and Disadvantages The availability of routes to all destinations at all times implies that much less delay is involved in the route setup process. The mechanism of incremental updates with sequence number tags makes the existing wired network protocols adaptable to ad hoc wireless networks. Hence, an existing wired network protocol can be applied to ad hoc wireless networks with many fewer modifications. The updates are propagated throughout the network in order to maintain an up-to-date view of the network topology at all the nodes. The updates due to broken links lead to a heavy control overhead during high mobility. Even a small network with high mobility or a large network with low mobility can completely choke the available bandwidth. Hence, this protocol suffers from excessive control overhead that is proportional to the number of nodes in the network and therefore is not scalable in ad hoc wireless networks, which have limited bandwidth and whose topologies are highly dynamic. Another disadvantage of DSDV is that in order to obtain information about a particular destination node, a node has to wait for a table update message initiated by the same destination node. This delay could result in stale routing information at nodes.

3.4.2 Wireless Routing Protocol The wireless routing protocol (WRP) , similar to DSDV, inherits the properties of the distributed Bellman-Ford algorithm. To counter the count-to-infinity problem and to enable faster convergence, it employs a unique method of maintaining information regarding the shortest distance to every destination node in the network and the penultimate hop node on the path to every destination node. Since WRP, like DSDV, maintains an up-to-date view of the network, every node has a readily available route to every destination node in the network. It differs from DSDV in table maintenance and in the update procedures. While DSDV maintains only one topology table, WRP uses a set of tables to maintain more accurate information. The tables that are maintained by a node are the following: distance table (DT), routing table (RT), link cost table (LCT), and a message retransmission list (MRL). The DT contains the network view of the neighbors of a node. It contains a matrix where each element contains the distance and the penultimate node reported by a neighbor for a particular destination. The RT contains the up-to-date view of the network for all known destinations. It keeps the shortest distance, the *predecessor* node (penultimate node), the *successor* node (the next node to reach the destination), and a flag indicating the status of the path. The path status may be a simple path (correct), or a loop (error), or the destination node not marked (null). The LCT contains the cost (*e.g.*, the number of hops to reach the destination) of relaying messages through each link. The cost of a broken link is ∞. It also contains the number of update periods (intervals between two successive periodic updates) passed since the last successful update was received from that link. This is done to detect link breaks. The MRL contains an entry for every update message that is to be retransmitted and maintains a counter for each entry. This counter is decremented after every retransmission of an update message. Each update message contains a list of updates. A node also marks each node in the RT that has to acknowledge the update message it transmitted. Once the counter reaches zero, the entries in the update message for which no acknowledgments have been received are to be retransmitted and the update message is deleted. Thus, a node detects a link break by the number of update periods missed since the last successful transmission. After receiving an update message, a node not only updates the

distance for transmitted neighbors but also checks the other neighbors' distance, hence convergence is much faster than DSDV. Consider the example shown in Figure 3.7, where the source of the route is node 1 and the destination is node 15. As WRP proactively maintains the route to all the destinations, the route to any destination node is readily available at the source node. From the routing table shown in Figure 3.7, the route from node 1 to node 15 has the next node as node 2. The predecessor node of 15 corresponding to this route is node 12. The predecessor information helps WRP to converge quickly during link breaks.

Figure 3.7. Route establishment in WRP.



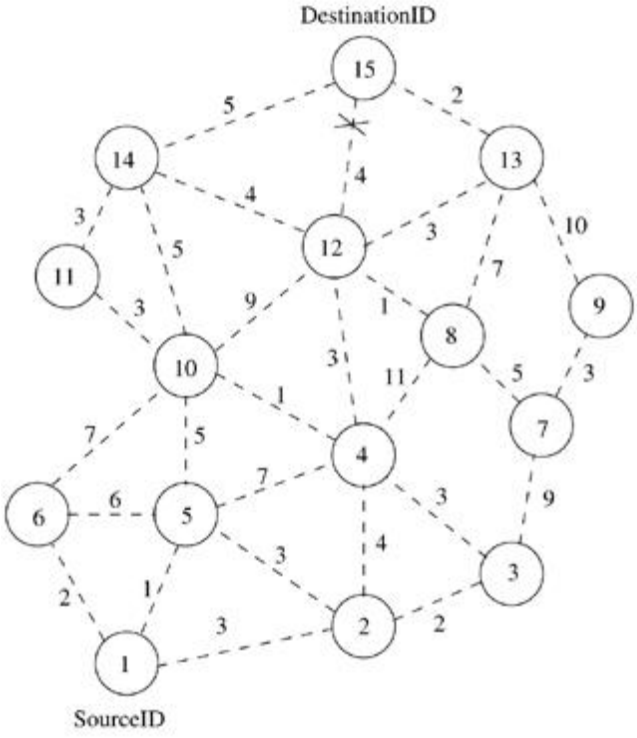
Routing Entry at Each Node for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	12	4
11	14	14	8
10	4	12	8
9	13	13	12
8	12	12	5
7	8	12	10
6	10	12	15
5	10	12	13
4	12	12	10
3	4	12	7
2	4	12	11
1	2	12	14

When a node detects a link break, it sends an update message to its neighbors with the link cost of the broken link set to ∞. After receiving the update message, all affected nodes update their minimum distances to the corresponding nodes (including the distance to the destination). The node that initiated the update message then finds an alternative route, if available from its

DT. Note that this new route computed will not contain the broken link. Consider the scenario shown in Figure 3.8. When the link between nodes 12 and 15 breaks, all nodes having a route to the destination with predecessor as node 12 delete their corresponding routing entries. Both node 12 and node 15 send update messages to their neighbors indicating that the cost of the link between nodes 12 and 15 is ∞. If the nodes have any other alternative route to the destination node 15, they update their routing tables and indicate the changed route to their neighbors by sending an update message. A neighbor node, after receiving an update message, updates its routing table only if the new path is better than the previously existing paths. For example, when node 12 finds an alternative route to the destination through node 13, it broadcasts an update message indicating the changed path. After receiving the update message from node 12, neighboring nodes 8, 14, 15, and 13 do not change their routing entry corresponding to destination 15 while node 4 and node 10 modify their entries to reflect the new updated path. Nodes 4 and 10 again send an update message to indicate the correct path to the destination for their respective neighbors. When node 10 receives node 4's update message, it again modifies its routing entry to optimize the path to the destination node (15) while node 4 discards the update entry it received from node 10.

Figure 3.8. Route maintenance in WRP.



Routing Entry at Each Node for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	13	5
11	14	14	8
10	4	13	9
9	13	13	12
8	12	13	6
7	8	13	11
6	10	13	16
5	10	13	14
4	12	13	8
3	4	13	11
2	4	13	12
1	2	13	15

Advantages and Disadvantages WRP has the same advantages as that of DSDV. In addition, it has faster convergence and involves fewer table updates. But the complexity of maintenance of multiple tables demands a larger memory and greater processing power from nodes in the ad hoc wireless network. At high mobility, the control overhead involved in updating table entries is almost the same as that of DSDV and hence is not suitable for highly dynamic and also for very large ad hoc wireless networks.

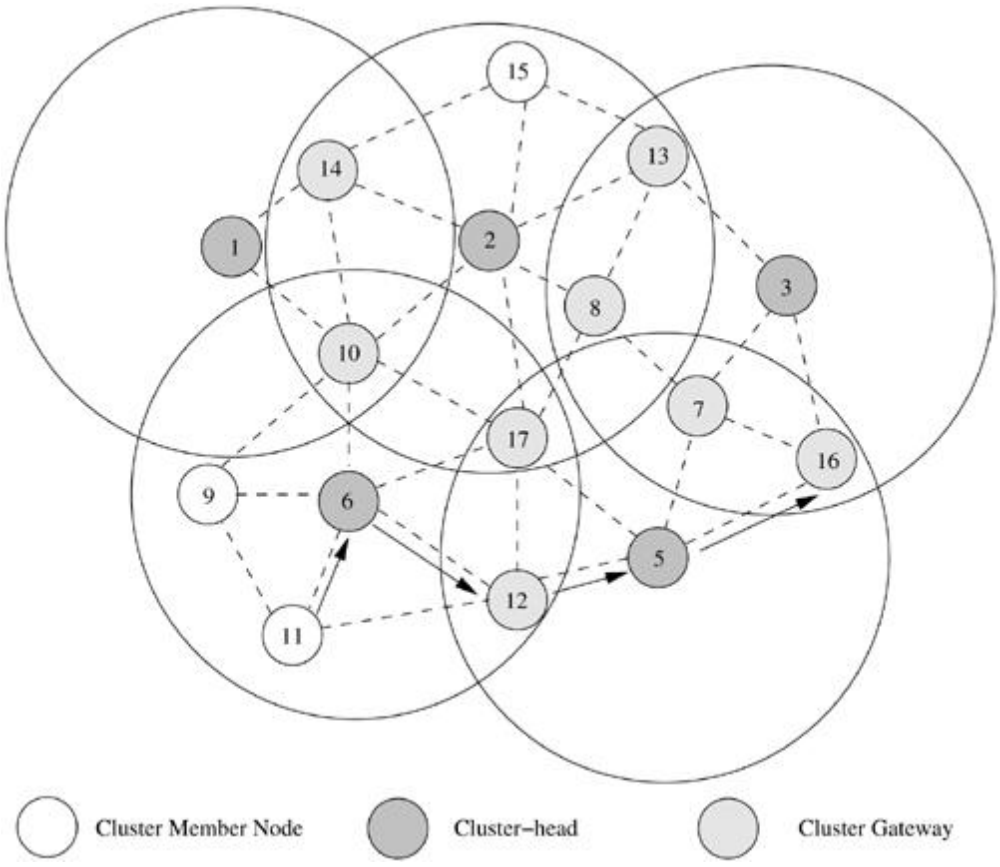
3.4.3 Cluster-Head Gateway Switch Routing Protocol

The cluster-head gateway switch routing protocol (CGSR) uses a hierarchical network topology, unlike other table-driven routing approaches that employ flat topologies. CGSR organizes nodes into clusters, with coordination among the members of each cluster entrusted to a special node named *clusterhead.* This cluster-head is elected dynamically by employing a *least cluster change (LCC)* algorithm . According to this algorithm, a node ceases to be a cluster-head only

if it comes under the range of another cluster-head, where the tie is broken either using the lowest ID or highest connectivity algorithm. Clustering provides a mechanism to allocate bandwidth, which is a limited resource, among different clusters, thereby improving reuse. For example, different cluster-heads could operate on different spreading codes on a CDMA system. Inside a cluster, the cluster-head can coordinate the channel access based on a token-based polling protocol. All member nodes of a cluster can be reached by the cluster-head within a single hop, thereby enabling the cluster-head to provide improved coordination among nodes that fall under its cluster. A token-based scheduling (assigning access token to the nodes in a cluster) is used within a cluster for sharing the bandwidth among the members of the cluster. CGSR assumes that all communication passes through the clusterhead. Communication between two clusters takes place through the common member nodes that are members of both the clusters. These nodes which are members of more than one cluster are called *gateways.* A gateway is expected to be able to listen to multiple spreading codes that are currently in operation in the clusters in which the node exists as a member. A gateway conflict is said to occur when a cluster-head issues a token to a gateway over a spreading code while the gateway is tuned to another code. Gateways that are capable of simultaneously communicating over two interfaces can avoid gateway conflicts. The performance of routing is influenced by token scheduling and code scheduling (assigning appropriate spreading codes to two different clusters) that are handled at cluster-heads and gateways, respectively. The routing protocol used in CGSR is an extension of DSDV. Every member node maintains a routing table containing the destination cluster-head for every node in the network. In addition to the cluster member table, each node maintains a routing table which keeps the list of next-hop nodes for reaching every destination cluster. The *cluster (hierarchical) routing protocol* is used here. As per this protocol, when a node with packets to be transmitted to a destination gets the token from its cluster-head, it obtains the destination cluster-head and the next hop node from the cluster member table and the routing table, respectively. CGSR improves the routing performance by routing packets through the cluster-heads and gateways. A path from any node *a* to any node *b* will be similar to $a - C_1 - G_1 - C_2 - G_2 - \ldots C_i - G_j \ldots G_n - b$, where G_i and C_j are

the i_{th} gateway and the j_{th} cluster-head, respectively, in the path. Figure 3.9 shows the cluster heads, *cluster gateways*, and normal cluster member nodes in an ad hoc wireless network. A path between node 11 and node 16 would follow 11 - 6 - 12 - 5 - 16. Since the cluster-heads gain more opportunities for transmission, the clusterheads, by means of a dynamic scheduling mechanism, can make CGSR obtain better delay performance for real-time flows. Route reconfiguration is necessitated by mainly two factors: firstly, the change in cluster-head and secondly, the stale entries in the cluster member table and routing table. CGSR depends on the table update mechanism to handle the latter problem, while the least cluster change algorithm handles the former.

Figure 3.9. Route establishment in CGSR.



Advantages and Disadvantages

CGSR is a hierarchical routing scheme which enables partial coordination between nodes by electing cluster-heads. Hence, better bandwidth utilization is possible. It is easy to implement priority scheduling schemes with token

scheduling and gateway code scheduling. The main disadvantages of CGSR are increase in path length and instability in the system at high mobility when the rate of change of cluster-heads is high. In order to avoid gateway conflicts, more resources (such as additional interfaces) are required. The power consumption at the cluster-head node is also a matter of concern because the battery-draining rate at the cluster-head is higher than that at a normal node. This could lead to frequent changes in the cluster-head, which may result in multiple path breaks.

3.4.4 Source-Tree Adaptive Routing Protocol

Source-tree adaptive routing protocol (STAR) proposed by Garcia-Luna-Aceves and Spohn is a variation of table-driven routing protocols, with the least overhead routing approach (LORA) as the key concept rather than the optimum routing approach (ORA) that was employed by earlier table-driven routing protocols. The ORA protocols attempt to update routing information quickly enough to provide optimum paths with respect to the defined metric (which may be the lowest number of hops), but with LORA, the routing protocol attempts to provide feasible paths that are not guaranteed to be optimal, but involve much less control overhead. In STAR protocol, every node broadcasts its *sourcetree* information. The source-tree of a node consists of the wireless links used by the node in its preferred path to destinations. Every node, using its adjacent links and the source-tree broadcast by its neighbors, builds a partial graph of the topology. During initialization, a node sends an update message to its neighbors. Also, every node is required to originate update messages about new destinations, the chances of routing loops, and the cost of paths exceeding a given threshold. Hence, each node will have a path to every destination node. The path, in most cases, would be sub-optimal. In the absence of a reliable link layer broadcast mechanism, STAR uses the following path-finding approach. When a node *s* has data packets to send to a particular destination *d*, for which no path exists in its source-tree, it originates an update message to all its neighbors indicating the absence of a path to *d*. This update message triggers another update message from a neighbor which has a path to *d*. Node *s* retransmits the update message as long as it does not have a path to *d* with increasing intervals between successive retransmissions. After getting the

source-tree update from a neighbor, the node *s* updates its source-tree and, using this, it finds a path to all nodes in the network. The data packet contains information about the path to be traversed in order to prevent the possibility of routing loop formation. In the presence of a reliable broadcast mechanism, STAR assumes implicit route maintenance. The link update message about the unavailability of a next-hop node triggers an update message from a neighbor which has an alternate source tree indicating an alternate next-hop node to the destination. In addition to path breaks, the intermediate nodes are responsible for handling the routing loops. When an intermediate node *k* receives a data packet to destination *d*, and one of the nodes in the packet's traversed path is present in node *k*'s path to the destination *d*, then it discards the packet and a *RouteRepair* update message is reliably sent to the node in the head of the route repair path. The route repair path corresponds to the path *k* to *x*, where *x* is the last router in the data packet's traversed path that is first found in the path *k* to *d*, that belongs to the source tree of *k*. The *RouteRepair* packet contains the complete source tree of node *k* and the traversed path of the packet. When an intermediate node receives a *RouteRepair* update message, it removes itself from the top of the route repair path and reliably sends it to the head of the route repair path.

Advantages and Disadvantages

STAR has very low communication overhead among all the table-driven routing protocols. The use of the LORA approach in this table-driven routing protocol reduces the average control overhead compared to several other on-demand routing protocols.

3.5 ON-DEMAND ROUTING PROTOCOLS

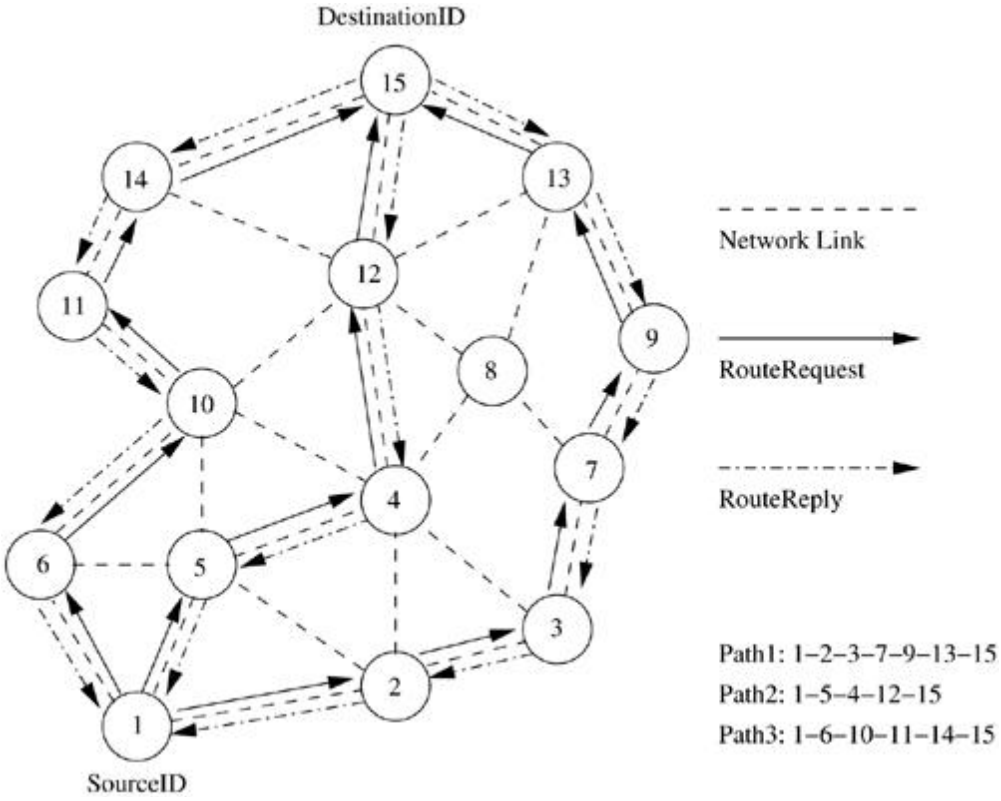
Unlike the table-driven routing protocols, on-demand routing protocols execute the path-finding process and exchange routing information only when a path is required by a node to communicate with a destination.

3.5.1 Dynamic Source Routing Protocol

Dynamic source routing protocol (DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. The major difference between this and the other on-demand routing protocols is that it is *beacon-less* and hence does not require periodic *hello*packet (*beacon*) transmissions, which are used by a node to inform its neighbors of its presence. The basic approach of this protocol (and all other on-demand routing protocols) during the route construction phase is to establish a route by flooding *RouteRequest* packets in the network. The destination node, on receiving a *RouteRequest* packet, responds by sending a *RouteReply* packet back to the source, which carries the route traversed by the *RouteRequest* packet received. Consider a source node that does not have a route to the destination. When it has data packets to be sent to that destination, it initiates a *RouteRequest* packet. This *RouteRequest* is flooded throughout the network. Each node, upon receiving a *RouteRequest* packet, rebroadcasts the packet to its neighbors if it has not forwarded already or if the node is not the destination node, provided the packet's time to live (TTL) counter has not exceeded. Each *RouteRequest* carries a sequence number generated by the source node and the path it has traversed. A node, upon receiving a *RouteRequest*packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate *RouteRequest.* The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same *RouteRequest* by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forward a *RouteRequest*packet during the route construction phase. A destination node, after receiving the first *RouteRequest*packet, replies to the source node through the reverse path the *RouteRequest* packet had traversed. In Figure 3.10, source node 1 initiates a *RouteRequest* packet to obtain a path for destination node 15. This protocol uses a route cache that stores all possible information extracted from the source route contained in a data packet. Nodes can also learn about the neighboring routes traversed by data packets if operated in the promiscuous mode (the mode of operation in which a node can receive the packets that are neither broadcast nor addressed to itself). This route cache is also used during the route construction

phase. If an intermediate node receiving a *RouteRequest* has a route to the destination node in its route cache, then it replies to the source node by sending a *RouteReply* with the entire route information from the source node to the destination node.

Figure 3.10. Route establishment in DSR.

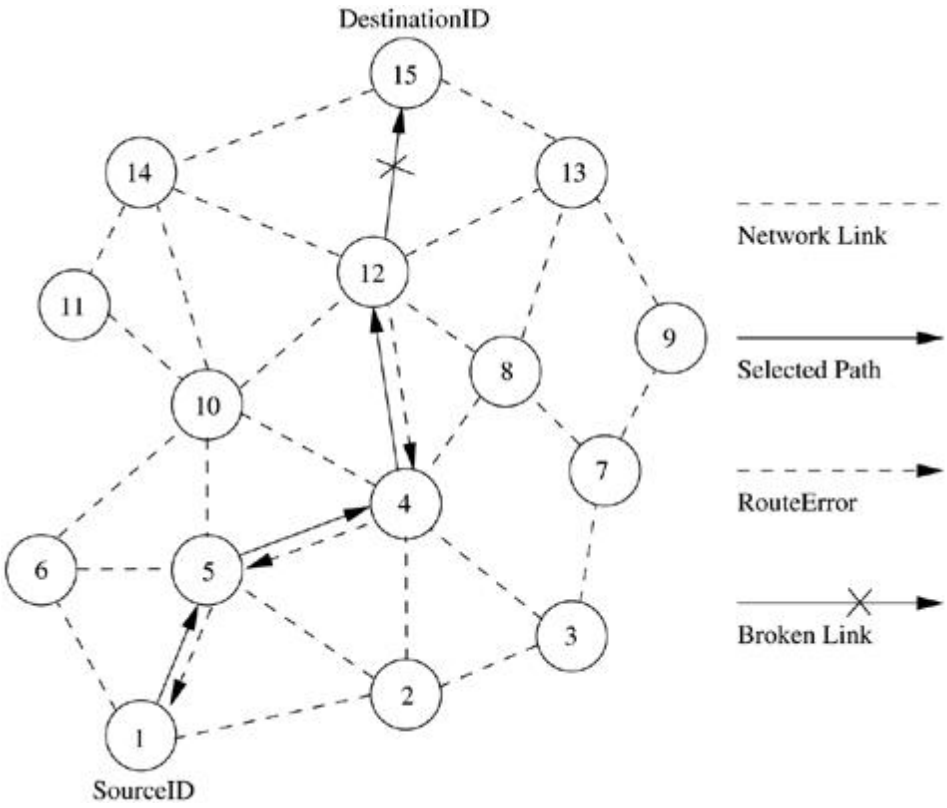


Optimizations

Several optimization techniques have been incorporated into the basic DSR protocol to improve the performance of the protocol. DSR uses the route cache at intermediate nodes. The route cache is populated with routes that can be extracted from the information contained in data packets that get forwarded. This cache information is used by the intermediate nodes to reply to the source when they receive a *RouteRequest* packet and if they have a route to the corresponding destination. By operating in the promiscuous mode, an intermediate node learns about route breaks. Information thus gained is used to update the route cache so that the active routes maintained in the route cache do not use such broken links. During network partitions, the affected nodes initiate

RouteRequest packets. An exponential backoff algorithm is used to avoid frequent *RouteRequest* flooding in the network when the destination is in another disjoint set. DSR also allows piggy-backing of a data packet on the*RouteRequest* so that a data packet can be sent along with the *RouteRequest.* If optimization is not allowed in the DSR protocol, the route construction phase is very simple. All the intermediate nodes flood the *RouteRequest* packet if it is not redundant. For example, after receiving the *RouteRequest* packet from node 1 (refer to Figure 3.10), all its neighboring nodes, that is, nodes 2, 5, and 6, forward it. Node 4 receives the *RouteRequest* from both nodes 2 and 5. Node 4 forwards the first *RouteRequest* it receives from any one of the nodes 2 and 5 and discards the other redundant/duplicate *RouteRequest* packets. The *RouteRequest* is propagated till it reaches the destination which initiates the *RouteReply*. As part of optimizations, if the intermediate nodes are also allowed to originate *RouteReply* packets, then a source node may receive multiple replies from intermediate nodes. For example, in Figure 3.11, if the intermediate node 10 has a route to the destination via node 14, it also sends the *RouteReply* to the source node. The source node selects the latest and best route, and uses that for sending data packets. Each data packet carries the complete path to its destination.

Figure 3.11. Route maintenance in DSR.



When an intermediate node in the path moves away, causing a wireless link to break, for example, the link between nodes 12 and 15 in Figure 3.11, a *RouteError* message is generated from the node adjacent to the broken link to inform the source node. The source node reinitiates the route establishment procedure. The cached entries at the intermediate nodes and the source node are removed when a *RouteError* packet is received. If a link breaks due to the movement of edge nodes (nodes 1 and 15), the source node again initiates the route discovery process.

Advantages and Disadvantages

This protocol uses a reactive approach which eliminates the need to periodically flood the network with table update messages which are required in a table-driven approach. In a reactive (on-demand) approach such as this, a route is established only when it is required and hence the need to find routes to all other nodes in the network as required by the table-driven approach is eliminated. The intermediate nodes also utilize the route cache information efficiently to reduce the control overhead. The disadvantage of this protocol is that the route

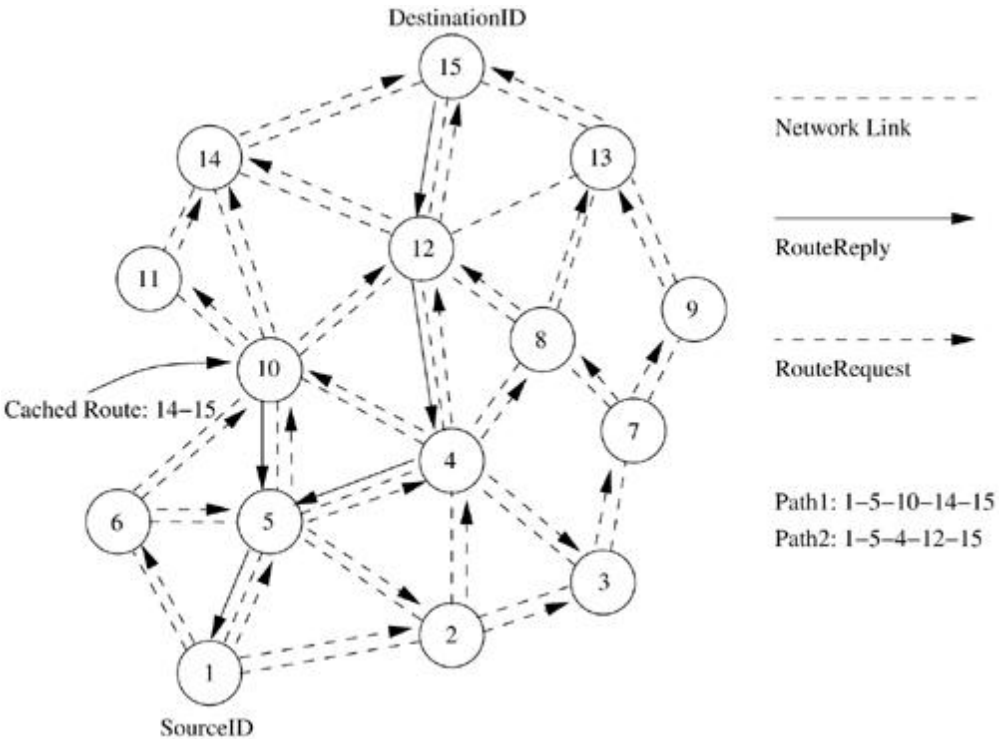
maintenance mechanism does not locally repair a broken link. Stale route cache information could also result in inconsistencies during the route reconstruction phase. The connection setup delay is higher than in table-driven protocols. Even though the protocol performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility. Also, considerable routing overhead is involved due to the source-routing mechanism employed in DSR. This routing overhead is directly proportional to the path length.

3.5.2 Ad Hoc On-Demand Distance-Vector Routing Protocol

Ad hoc on-demand distance vector (AODV) routing protocol uses an on demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and DSR stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on demand routing protocol, the source node floods the *RouteRequest* packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single *RouteRequest.* The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the DestSeqNum of the current packet received is greater than the last DestSeqNum stored at the node. A *RouteRequest* carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcSeqNum), the destination sequence number (DestSeqNum), the broadcast identifier (BcastID), and the time to live (TTL) field. DestSeqNum indicates the freshness of the route that is accepted by the source. When an intermediate node receives a *RouteRequest*, it either forwards it or prepares a *RouteReply* if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the *RouteRequest* packet. If a *RouteRequest* is received

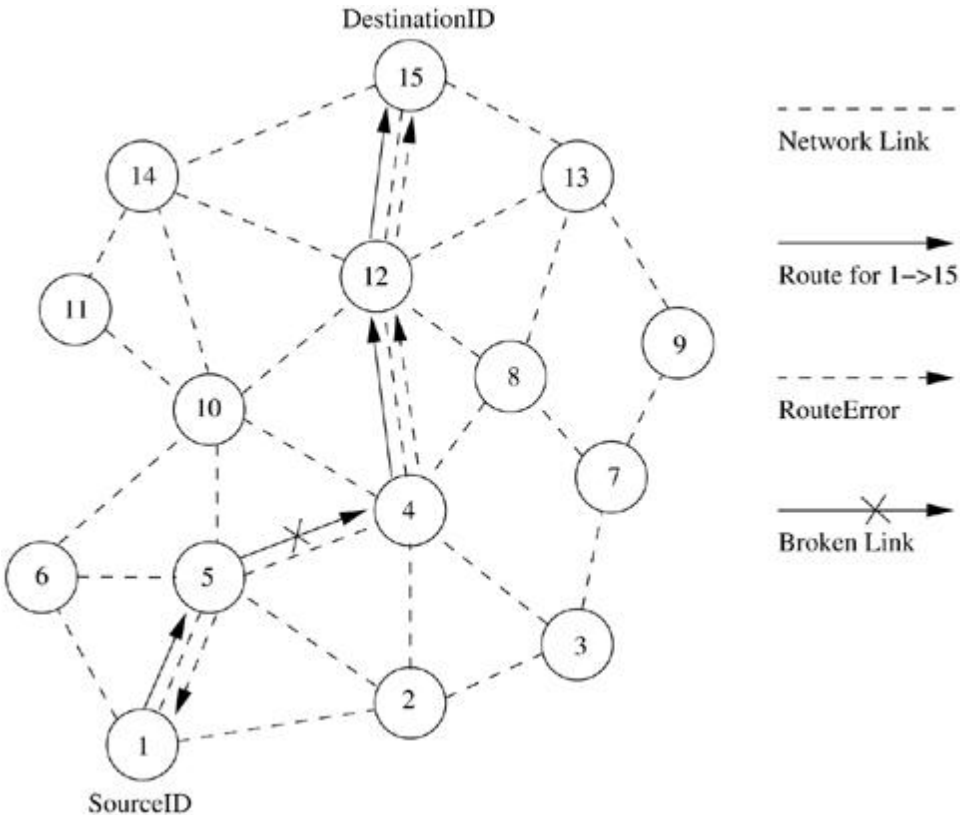
multiple times, which is indicated by the BcastID-SrcID pair, the duplicate copies are discarded. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send*RouteReply* packets to the source. Every intermediate node, while forwarding a *RouteRequest*, enters the previous node address and its BcastID. A timer is used to delete this entry in case a *RouteReply* is not received before the timer expires. This helps in storing an active path at the intermediate node as AODV does not employ source routing of data packets. When a node receives a *RouteReply* packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop toward the destination. Consider the example depicted in Figure 3.12. In this figure, source node 1 initiates a path-finding process by originating a *RouteRequest* to be flooded in the network for destination node 15, assuming that the *RouteRequest* contains the destination sequence number as 3 and the source sequence number as 1. When nodes 2, 5, and 6 receive the *RouteRequest* packet, they check their routes to the destination. In case a route to the destination is not available, they further forward it to their neighbors. Here nodes 3, 4, and 10 are the neighbors of nodes 2, 5, and 6. This is with the assumption that intermediate nodes 3 and 10 already have routes to the destination node, that is, node 15 through paths 10-14-15 and 3-7-9-13-15, respectively. If the destination sequence number at intermediate node 10 is 4 and is 1 at intermediate node 3, then only node 10 is allowed to reply along the cached route to the source. This is because node 3 has an older route to node 15 compared to the route available at the source node (the destination sequence number at node 3 is 1, but the destination sequence number is 3 at the source node), while node 10 has a more recent route (the destination sequence number is 4) to the destination. If the *RouteRequest* reaches the destination (node 15) through path 4-12-15 or any other alternative route, the destination also sends a *RouteReply* to the source. In this case, multiple*RouteReply* packets reach the source. All the intermediate nodes receiving a *RouteReply* update their route tables with the latest destination sequence number. They also update the routing information if it leads to a shorter path between source and destination.

Figure 3.12. Route establishment in AODV.



AODV does not repair a broken path locally. When a link breaks, which is determined by observing the periodical *beacon*s or through link-level acknowledgments, the end nodes (*i.e.*, source and destination nodes) are notified. When a source node learns about the path break, it reestablishes the route to the destination if required by the higher layers. If a path break is detected at an intermediate node, the node informs the end nodes by sending an unsolicited *RouteReply* with the hop count set as ∞. Consider the example illustrated in Figure 3.13. When a path breaks, for example, between nodes 4 and 5, both the nodes initiate *RouteError* messages to inform their end nodes about the link break. The end nodes delete the corresponding entries from their tables. The source node reinitiates the pathfinding process with the new BcastID and the previous destination sequence number.

Figure 3.13. Route maintenance in AODV.



Advantages and Disadvantages

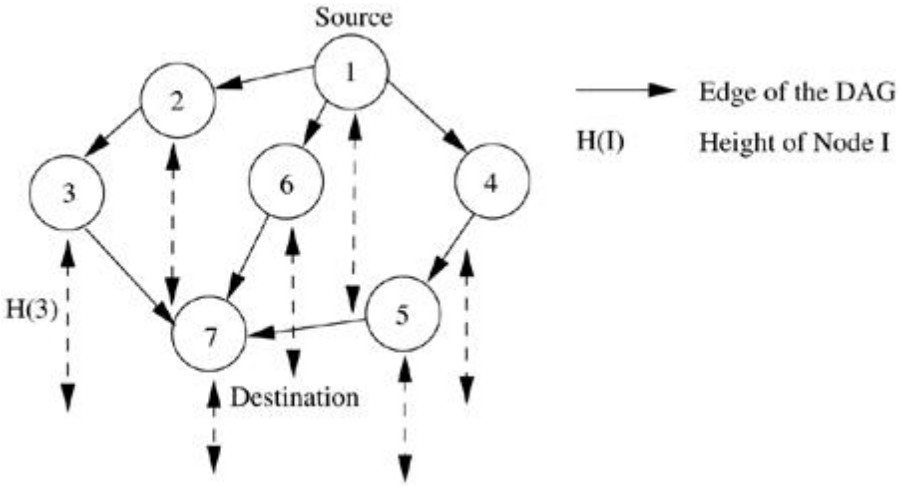
The main advantage of this protocol is that routes are established on demand and destination sequence numbers are used to find the latest route to the destination. The connection setup delay is less. One of the disadvantages of this protocol is that intermediate nodes can lead to inconsistent routes if the source sequence number is very old and the intermediate nodes have a higher but not the latest destination sequence number, thereby having stale entries. Also multiple *RouteReply* packets in response to a single *RouteRequest* packet can lead to heavy control overhead. Another disadvantage of AODV is that the periodic *beacon*ing leads to unnecessary bandwidth consumption.

3.5.3 Temporally Ordered Routing Algorithm

Temporally ordered routing algorithm (TORA) is a source-initiated on-demand routing protocol which uses a *link reversal algorithm* and provides loop-free multipath routes to a destination node. InTORA, each node maintains its one-hop local topology information and also has the capability to detect partitions. TORA has the unique property of limiting the control packets to a small region

during the reconfiguration process initiated by a path break. Figure 3.14 shows the distance metric used in TORAwhich is nothing but the length of the path, or the height from the destination. H(N) denotes the height of node N from the destination. TORA has three main functions: establishing, maintaining, and erasing routes.

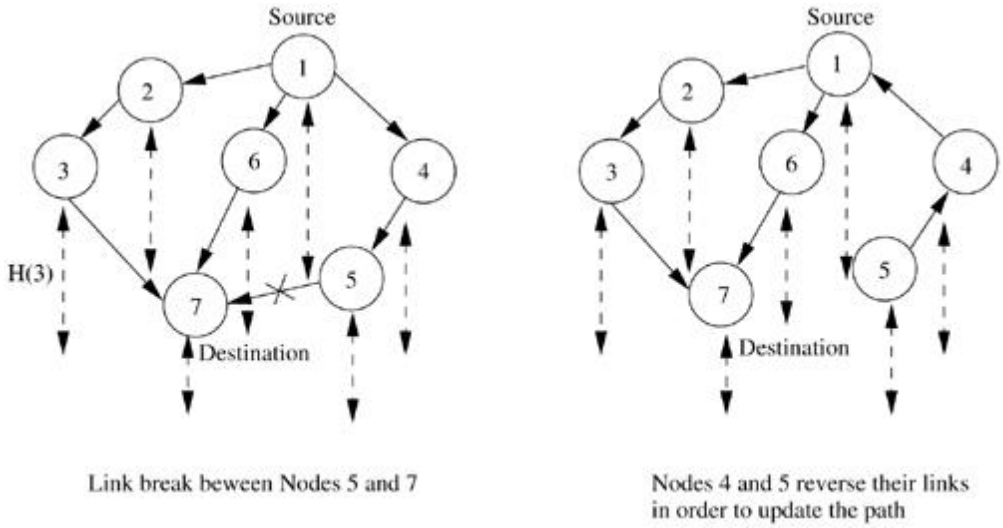
Figure 3.14. Illustration of temporal ordering in TORA.



The route establishment function is performed only when a node requires a path to a destination but does not have any directed link. This process establishes a destination-oriented directed acyclic graph (DAG) using a *Query/Update* mechanism. Consider the network topology shown in Figure 3.14. When node 1 has data packets to be sent to the destination node 7, a *Query* packet is originated by node 1 with the destination address included in it. This *Query* packet is forwarded by intermediate nodes 2, 3, 4, 5, and 6, and reaches the destination node 7, or any other node which has a route to the destination. The node that terminates (in this case, node 7) the *Query* packet replies with an *Update* packet containing its distance from the destination (it is zero at the destination node). In the example, the destination node 7 originates an *Update* packet. Each node that receives the *Update* packet sets its distance to a value higher than the distance of the sender of the *Update* packet. By doing this, a set of directed links from the node which originated the *Query* to the destination node 7 is created. This forms the DAG depicted in Figure 3.14. Once a path to the destination is obtained, it is considered to exist as long as the path is

available, irrespective of the path length changes due to the reconfigurations that may take place during the course of the data transfer session. When an intermediate node (say, node 5) discovers that the route to the destination node is invalid, as illustrated in Figure 3.15, it changes its distance value to a higher value than its neighbors and originates an *Update* packet. The neighboring node 4 that receives the *Update* packet reverses the link between 1 and 4 and forwards the *Update* packet. This is done to update the DAG corresponding to destination node 7. This results in a change in the DAG. If the source node has no other neighbor that has a path to the destination, it initiates a fresh *Query/Update* procedure. Assume that the link between nodes 1 and 4 breaks. Node 4 reverses the path between itself and node 5, and sends an update message to node 5. Since this conflicts with the earlier reversal, a partition in the network can be inferred. If the node detects a partition, it originates a *Clear* message, which erases the existing path information in that partition related to the destination.

Figure 3.15. Illustration of route maintenance in TORA.



Link break beween Nodes 5 and 7

Nodes 4 and 5 reverse their links in order to update the path

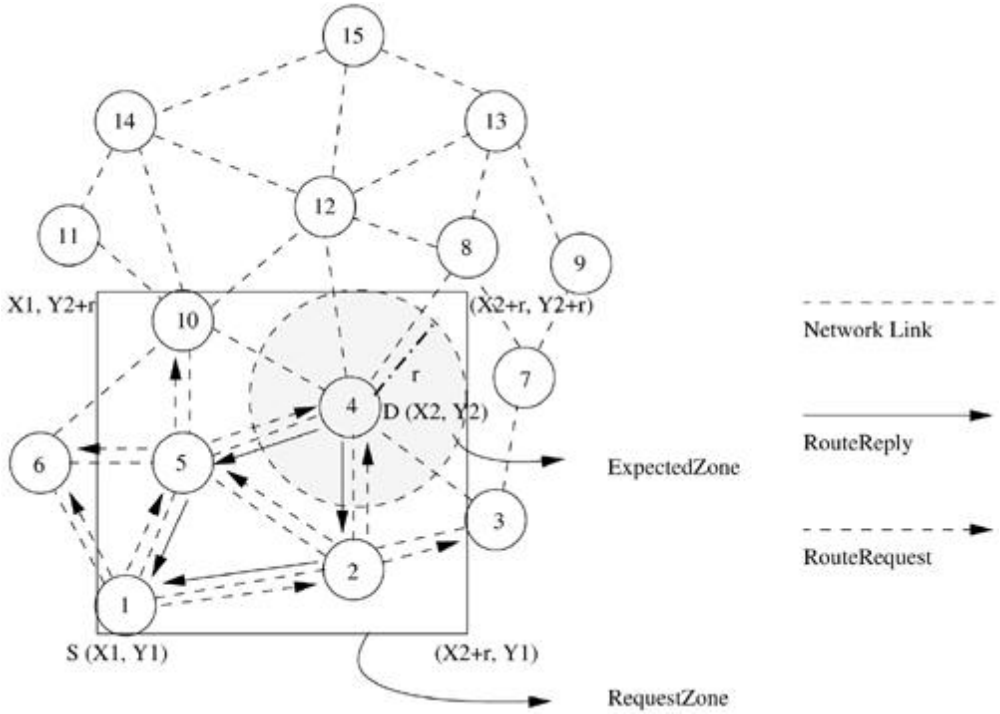
Advantages and Disadvantages

By limiting the control packets for route reconfigurations to a small region, TORA incurs less control overhead. Concurrent detection of partitions and subsequent deletion of routes could result in temporary oscillations and transient loops. The local reconfiguration of paths results in non-optimal routes.

3.5.4 Location-Aided Routing

Location-aided routing protocol (LAR) utilizes the location information for improving the efficiency of routing by reducing the control overhead. LAR assumes the availability of the global positioning system (GPS) for obtaining the geographical position information necessary for routing. LAR designates two geographical regions for selective forwarding of control packets, namely,*ExpectedZone* and *RequestZone*. The *ExpectedZone* is the region in which the destination node is expected to be present, given information regarding its location in the past and its mobility information (refer to Figure 3.16). In the event of non-availability of past information about the destination, the entire network area is considered to be the *ExpectedZone* of the destination. Similarly, with the availability of more information about its mobility, the *ExpectedZone* of the destination can be determined with more accuracy and improved efficiency. The *RequestZone* is a geographical region within which the path-finding control packets are permitted to be propagated. This area is determined by the sender of a data transfer session. The control packets used for path-finding are forwarded by nodes which are present in the *RequestZone* and are discarded by nodes outside the zone. In situations where the sender or the intermediate relay nodes are not present in the *RequestZone*, additional area is included for forwarding the packets. This is done when the first attempt for obtaining a path to a destination using the initial *RequestZone* fails to yield a path within a sufficiently long waiting time. In this case, the second attempt repeats the process with increased *RequestZone* size to account for mobility and error in location estimation. LAR uses flooding, but here flooding is restricted to a small geographical region. The nodes decide to forward or discard the control packets based on two algorithms, namely, LAR1 and LAR2.

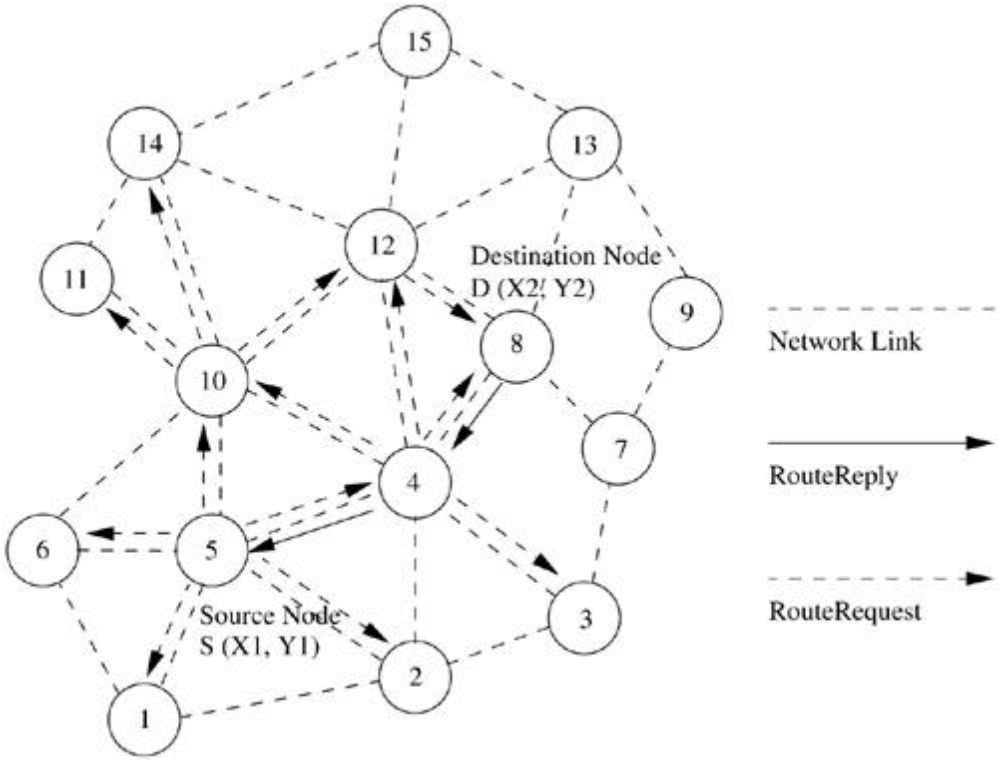
Figure 3.16. *RequestZone* and *ExpectedZone* in LAR1.



In the LAR1 algorithm, the source node (say, *S*) explicitly specifies the *Request- Zone* in the*RouteRequest* packet. As per LAR1, as illustrated in Figure 3.16, the *RequestZone* is the smallest rectangle that includes the source node (*S*) and the *ExpectedZone,* the sides of which are parallel to the X and Y axes, when the node *S* is outside the *ExpectedZone.* When node *S* is within the *ExpectedZone,*then the *RequestZone* is reduced to the *ExpectedZone* itself. Every intermediate node that receives the*RouteRequest* packet verifies the *RequestZone* information contained in the packet and forwards it further if the node is within the *RequestZone;* otherwise, the packet is discarded. In Figure 3.16, the source node (node 1) originates a *RouteRequest,* which is broadcast to its neighbors (2, 5, and 6). These nodes verify their own geographical locations to check whether they belong to the *ExpectedZone.*Nodes 2 and 5 find that they are inside the *ExpectedZone* and hence they forward the *RouteRequest.*But node 6 discards the packet. Finally, when the *RouteRequest* reaches the destination node (node 4), it originates a *RouteReply* that contains the current location and current time of the node. Also, as an option, the current speed of movement can be included in the *RouteReply* if that information is available with the node. Such information included in the *RouteReply* packet is used by the source node for future route establishment procedures. In LAR2 algorithm (Figure 3.17), the

source node *S* (node 5) includes the distance between itself and the destination node *D* (node 8) along with the (*X*, *Y*) coordinates of the destination node *D* in the*RouteRequest* packet instead of the explicit information about the *Expected Region.* When an intermediate node receives this *RouteRequest* packet, it computes the distance to the node *D.* If this distance is less than the distance from *S* to node $D + \delta$, where δ is a parameter of the algorithm decided based on the error in location estimation and mobility, then the *RouteRequest* packet is forwarded. Otherwise, the *RouteRequest* is discarded. Consider the example illustrated in Figure 3.17. Assume that the value of δ is 0 here. The *RouteRequest* packet originated by node 5 is received by nodes 1, 2, 4, 6, and 10. Only nodes 4 and 10 find that the distance between them and the destination is less than the distance between the node 5 and the destination node; other nodes discard the *RouteRequest.* A*RouteRequest* packet is forwarded only once and the distance between the forwarding node and *D* is updated in the *RouteRequest* packet for further relaying. When node 4 forwards the *RouteRequest*packet, it updates the packet with the distance between itself and the destination node *D.* This packet, after being received at neighbor node 3, is discarded due to the fact that the distance between node 3 and the node 8 is greater than the distance between nodes 4 and 8. Once the *RouteRequest* reaches node 8, it originates a *RouteReply* packet back to the source node 5, containing the path through which future data packets are to be propagated. In order to compensate for the location error (due to the inaccuracy of GPS information or due to changes in the mobility of the nodes), a larger *RequestZone*that can accommodate the amount of error that occurred is considered.

Figure 3.17. Route establishment in LAR2.



Advantages and Disadvantages

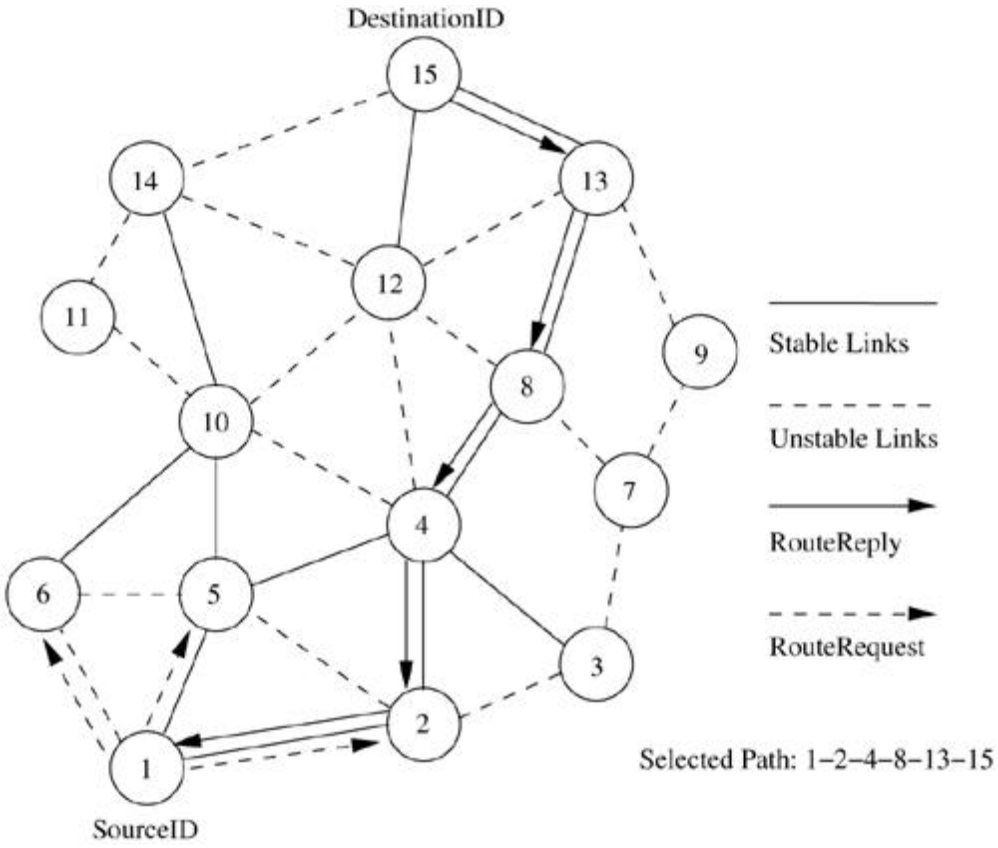
LAR reduces the control overhead by limiting the search area for finding a path. The efficient use of geographical position information, reduced control overhead, and increased utilization of bandwidth are the major advantages of this protocol. The applicability of this protocol depends heavily on the availability of GPS infrastructure or similar sources of location information. Hence, this protocol cannot be used in situations where there is no access to such information.

3.5.5 Associativity-Based Routing

Associativity-based routing (ABR) protocol is a distributed routing protocol that selects routes based on the stability of the wireless links. It is a *beacon*-based, on-demand routing protocol. A link is classified as stable or unstable based on its temporal stability. The temporal stability is determined by counting the periodic *beacon*s that a node receives from its neighbors. Each node maintains the count of its neighbors' *beacon*s and classifies each link as stable or unstable based on the *beacon* count corresponding to the neighbor node concerned. The link corresponding to a stable neighbor is termed as a stable

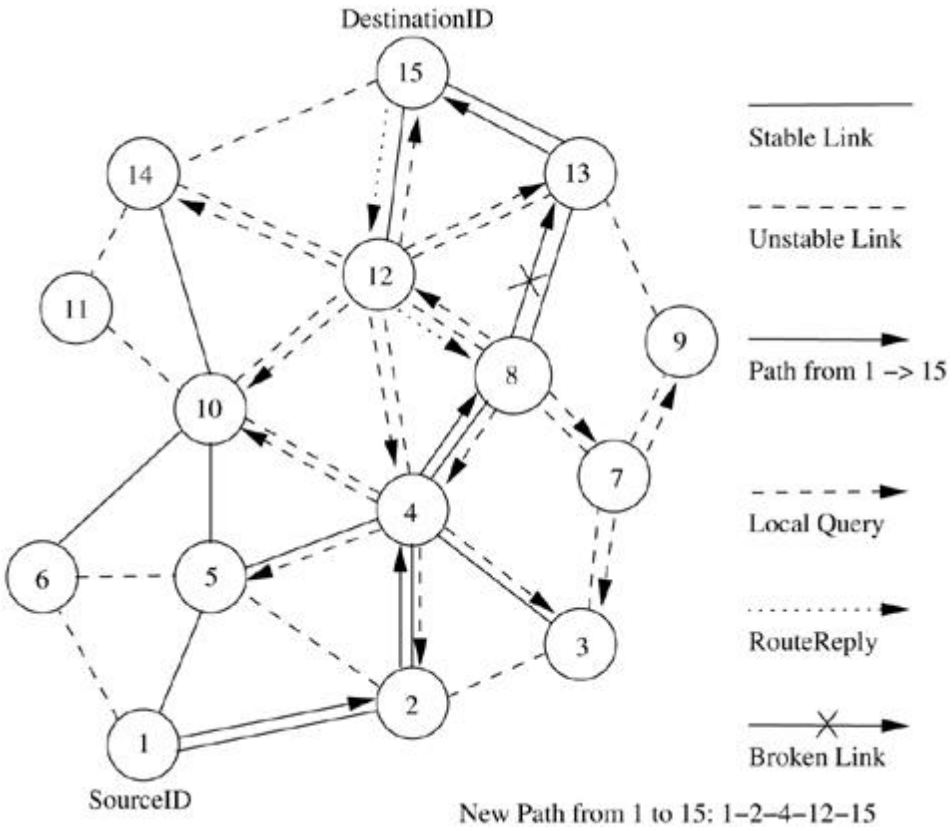
link, while a link to an unstable neighbor is called an unstable link. A source node floods *RouteRequest* packets throughout the network if a route is not available in its route cache. All intermediate nodes forward the *RouteRequest* packet. A *RouteRequest* packet carries the path it has traversed and the *beacon* count for the corresponding node in the path. When the first*RouteRequest* reaches the destination, the destination waits for a time period $T_{RouteSelectTime}$ to receive multiple *RouteRequest*s through different paths. After this time duration, the destination selects the path that has the maximum proportion of stable links. If two paths have the same proportion of stable links, the shorter of them is selected. If more than one shortest path is available, then a random path among them is selected as the path between source and destination. Consider Figure 3.18, in which the source node (node 1) initiates the *RouteRequest* to be flooded for finding a route to the destination node (node 15). The solid lines represent the *stable* links that are classified based on the *beacon* count, while dotted lines represent *unstable* links. ABR does not restrict any intermediate node from forwarding a *RouteRequest* packet based on the stable or unstable link criterion. It uses stability information only during the route selection process at the destination node. As depicted in Figure 3.18, the *RouteRequest* reaches the destination through three different routes. Route 1 is 1-5-10-14-15, route 2 is 1-5-4-12-15, and route 3 is 1-2-4-8-13- 15. ABR selects route 3 as it contains the highest percentage of stable links compared to route 1 and route 2. ABR gives more priority to stable routes than to shorter routes. Hence, route 3 is selected even though the length of the selected route is more than that of the other two routes.

Figure 3.18. Route establishment in ABR.



If a link break occurs at an intermediate node, the node closer to the source, which detects the break, initiates a local route repair process. In this process, the node locally broadcasts a route repair packet, termed the local query (LQ) broadcast, with a limited time to live (TTL), as illustrated in Figure 3.19where a TTL value of 2 is used. This way a broken link is bypassed locally without flooding a new*RouteRequest* packet in the whole network. If a node fails to repair the broken link, then its uplink node (the previous node in the path which is closer to the source node) reinitiates the LQ broadcast. This route repair process continues along the intermediate nodes toward the source node until it traverses half the length of the broken path or the route is repaired. In the former case, the source node is informed, which initiates a new route establishment phase.

Figure 3.19. Route maintenance in ABR.



Consider the example in Figure 3.19. When a path break occurs between nodes 8 and 13, the node adjacent to the broken link toward the source node, that is, node 8, initiates a local query broadcast in order to locally repair the broken path. The local query has limited scope with the maximum TTL value set to the remaining path length from the broken link to the destination. In the same figure, the broken path is repaired locally by bypassing the path segment 8-13-15 through segment 8-12-15.

Advantages and Disadvantages

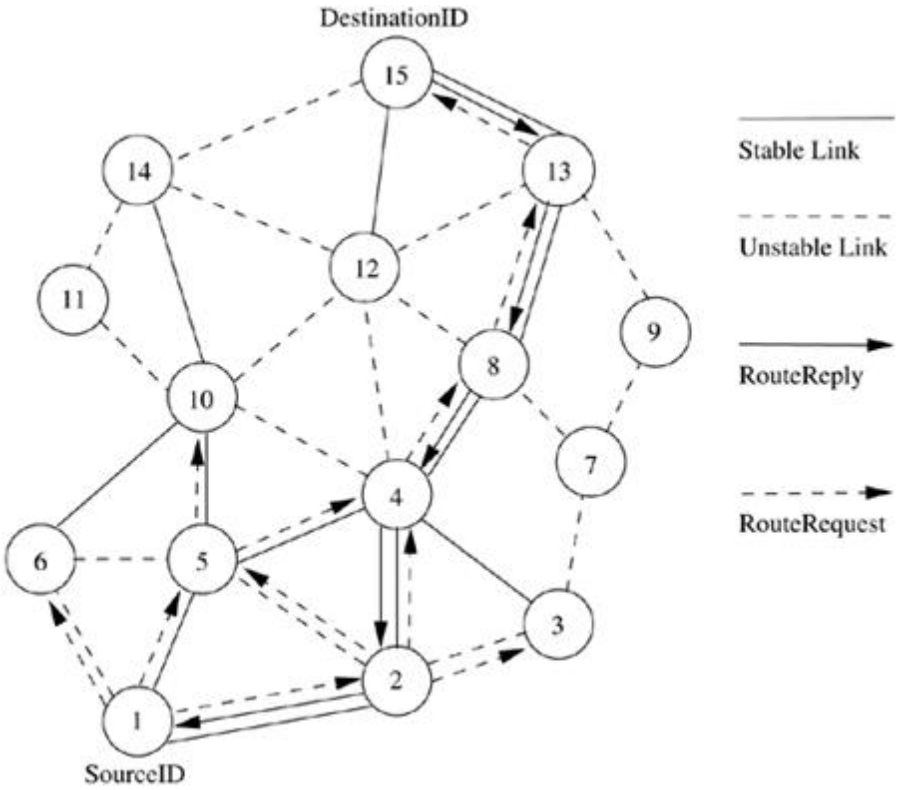
In ABR, stable routes have a higher preference compared to shorter routes. They result in fewer path breaks which, in turn, reduces the extent of flooding due to reconfiguration of paths in the network. One of the disadvantages of this protocol is that the chosen path may be longer than the shortest path between the source and destination because of the preference given to stable paths. Another disadvantage is that repetitive LQ broadcasts may result in high delays during route repairs.

3.5.6 Signal Stability-Based Adaptive Routing Protocol

Signal stability-based adaptive routing protocol (SSA) is an on-demand routing protocol that uses signal stability as the prime factor for finding stable routes. This protocol is *beacon*-based, in which the signal strength of the *beacon* is measured for determining link stability. The signal strength is used to classify a link as *stable* or *unstable*. This protocol consists of two parts: forwarding protocol (FP) and dynamic routing protocol (DRP). These protocols use an extended radio interface that measures the signal strength from *beacon*s. DRP maintains the routing table by interacting with the DRP processes on other hosts. FP performs the actual routing to forward a packet on its way to the destination. Every node maintains a table that contains the *beacon* count and the signal strength of each of its neighbors. If a node has received strong *beacon*s for the past few *beacons*, the node classifies the link as a *strong*/*stable* link. The link is otherwise classified as a *weak*/*unstable* link. Each node maintains a table called the signal stability table (SST), which is based on the signal strengths of its neighbors' *beacon*s. This table is used by the nodes in the path to the destination to forward the incoming *RouteRequest* over strong links for finding the most stable end-to-end path. If the attempt of forwarding a *RouteRequest* over the stable links fails to obtain any path to a destination, the protocol floods the *RouteRequest* throughout the network without considering the stability of links as the forwarding criterion. A source node which does not have a route to the destination floods the network with *RouteRequest* packets. But unlike other routing protocols, nodes that employ the SSA protocol process a *RouteRequest* only if it is received over a strong link. A *RouteRequest* received through a weak link is dropped without being processed. The destination selects the first *RouteRequest* packet received over strong links. The destination initiates a *RouteReply* packet to notify the selected route to the source. Consider Figure 3.20, where the source node (node 1) broadcasts a *RouteRequest* for finding the route to the destination node (node 15). In Figure 3.20, solid lines represent the *stable* links, while the dotted lines represent *weak* links. Unlike ABR, SSA restricts intermediate nodes from forwarding a*RouteRequest* packet if the packet had been received over a weak link. It forwards only *RouteRequest* packets received over stable links. In Figure 3.20, when the *RouteRequest* is initiated by

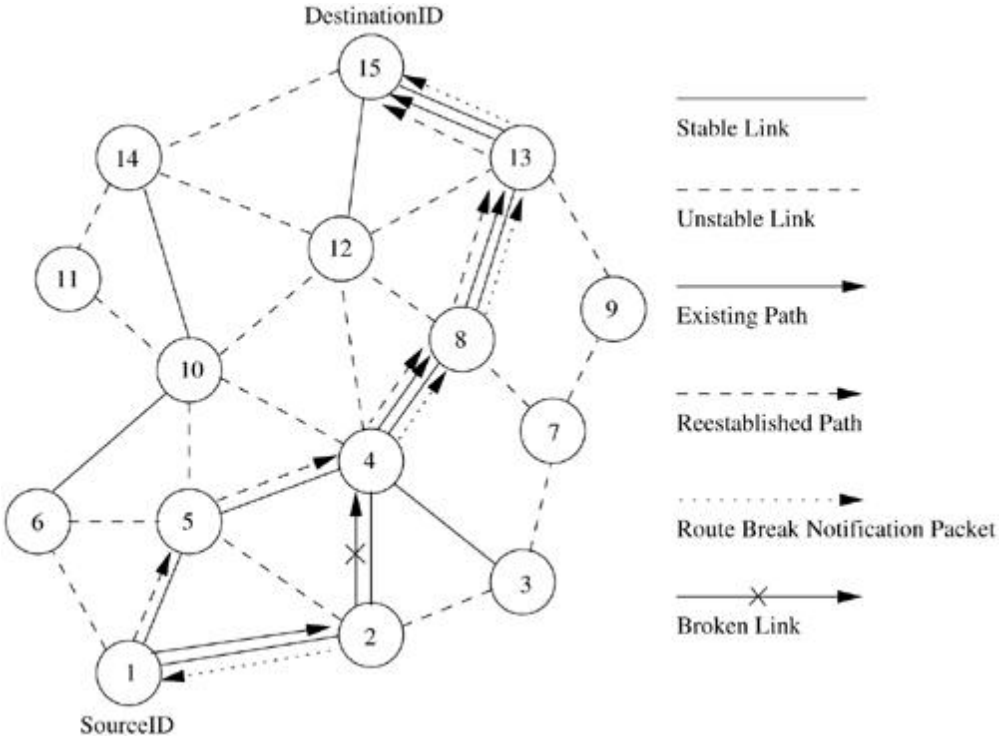
the source, it is to be processed by all its neighbors. But before processing, each neighbor node checks whether the *RouteRequest* packet was received through a stable link. If the *RouteRequest* had been received through a stable link and had not been sent already (*i.e.*, it is not a duplicate *RouteRequest*), it is forwarded by the node; otherwise, it is dropped. For example, when the *RouteRequest* from node 1 reaches nodes 2, 5, and 6, it is forwarded only by nodes 2 and 5 as the link between nodes 1 and 6 is weak. Similarly, the *RouteRequest* forwarded by node 2 is rejected by nodes 3 and 5, while node 4 forwards it to its neighbors, provided it has not already been forwarded. In this way, nodes forward the*RouteRequest* until it reaches the destination. In this example, only one route is established based on the strong link criterion. A *RouteRequest* propagated through path 1-5-10-14-15 is rejected by the destination as it receives a *RouteRequest* from node 14 on a weak link. The stable path consisting of strong links is 1- 2 (or 5)-4-8-13-15. The first *RouteRequest* packet that reaches the destination over a stable path is selected by the destination.

Figure 3.20. Route establishment in SSA.



As shown in Figure 3.21, when a link breaks, the end nodes of the broken link (*i. e.*, nodes 2 and 4) notify the corresponding end nodes of the path (*i. e.*, nodes 1 and 15). A source node, after receiving a route break notification packet, rebroadcasts the *RouteRequest* to find another stable path to the destination. Stale entries are removed only if data packets that use the stale route information fail to reach the next node. If the link between nodes 2 and 4 breaks, a new strong path is established through 1-5-4-8-13-15. If no strong path is available when a link gets broken (*e.g.*, link 8-13), then the new route is established by considering weak links also. This is done when multiple *RouteRequest* attempts fail to obtain a path to the destination using only the stable links.

Figure 3.21. Route maintenance in SSA.



Advantages and Disadvantages

The main advantage of this protocol is that it finds more stable routes when compared to the shortest path route selection protocols such as DSR and AODV. This protocol accommodates temporal stability by using *beacon* counts to classify a link as stable or weak. The main disadvantage of this protocol is

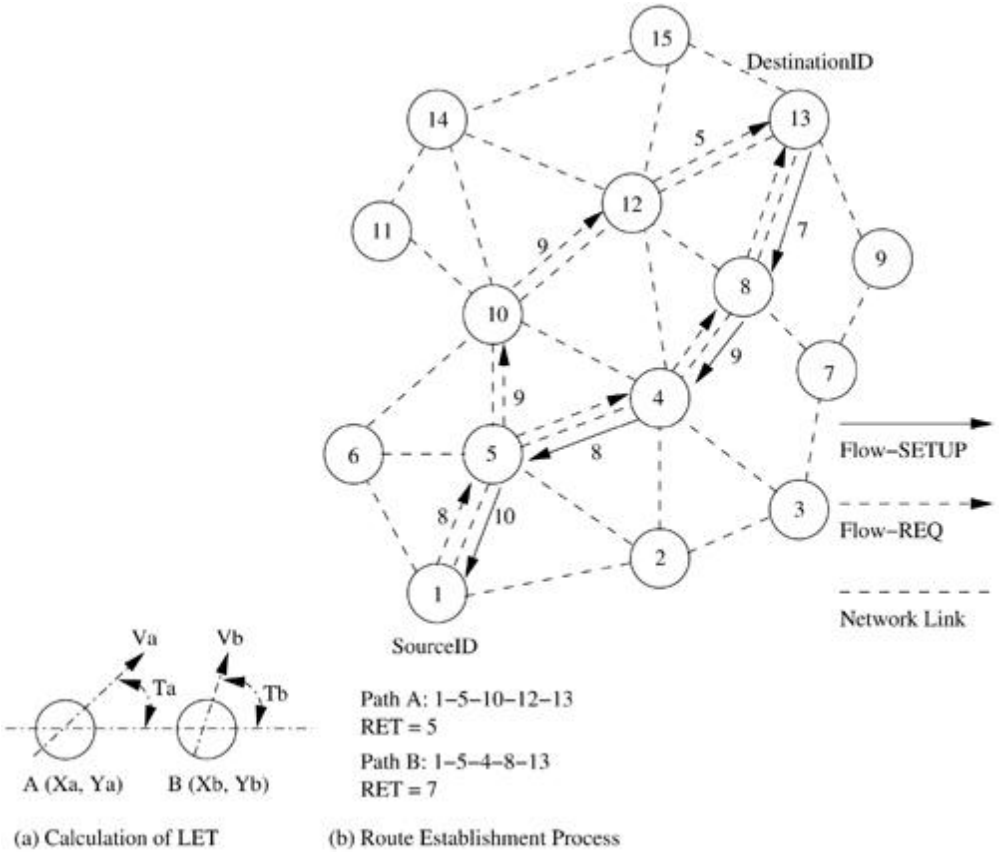
that it puts a strong *RouteRequest* forwarding condition which results in *RouteRequest* failures. A failed *RouteRequest* attempt initiates a similar path-finding process for a new path without considering the stability criterion. Such multiple flooding of *RouteRequest* packets consumes a significant amount of bandwidth in the already bandwidth-constrained network, and also increases the path setup time. Another disadvantage is that the strong links criterion increases the path length, as shorter paths may be ignored for more stable paths.

3.5.7 Flow-Oriented Routing Protocol

Flow-oriented routing protocol (FORP) is an on-demand routing protocol that employs a prediction-based *multi-hop-handoff* mechanism for supporting time-sensitive traffic in ad hoc wireless networks. This protocol has been proposed for IPv6-based ad hoc wireless networks where quality of service (QoS) needs to be provided. The *multi-hop-handoff* is aimed at alleviating the effects of path breaks on the real-time packet flows. A sender or an intermediate node initiates the route maintenance process only after detecting a link break. This reactive route maintenance procedure may result in high packet loss, leading to a low quality of service provided to the user. FORP uses a unique prediction-based mechanism that utilizes the mobility and location information of nodes to estimate the link expiration time (LET). LET is the approximate lifetime of a given wireless link. The minimum of the LET values of all wireless links on a path is termed as the route expiry time (RET). Every node is assumed to be able to predict the LET of each of its links with its neighbors. The LET between two nodes can be estimated using information such as current position of the nodes, their direction of movement, and their transmission ranges. FORP requires the availability of GPS information in order to identify the location of nodes. When a sender node needs to set up a real-time flow to a particular destination, it checks its routing table for the availability of a route to that destination. If a route is available, then that is used to send packets to the destination. Otherwise, the sender broadcasts a *Flow-REQ* packet carrying information regarding the source and the destination nodes. The *Flow-REQ* packet also carries a flow identification number/sequence number which is unique for every session. A neighbor node, on receiving this packet, first checks if the sequence number of

the received *Flow-REQ* is higher than the sequence number corresponding to a packet belonging to the same session that had been previously forwarded by the node. If so, then it updates its address on the packet and extracts the necessary state information out of the packet. If the sequence number on the packet is less than that of the previously forwarded packet, then the packet is discarded. This is done to avoid looping of *Flow-REQ* packets. A *Flow-REQ* with the same sequence number as that of a *Flow-REQ* belonging to the same session which had been forwarded already by the node, would be broadcast further only if it has arrived through a shorter (and therefore better) path. Before forwarding a *Flow-REQ,* the intermediate node appends its node address and the LET of the last link the packet had traversed onto the packet. The *Flow-REQ* packet, when received at the destination node, contains the list of nodes on the path it had traversed, along with the LET values of every wireless link on that path. FORP assumes all the nodes in the network to be synchronized to a common time by means of GPS information. If the calculated value of RET, corresponding to the new *Flow-REQ* packet arrived at the destination, is better than the RET value of the path currently being used, then the destination originates a *Flow-SETUP* packet. The LET of a link can be estimated given the information about the location, velocity, and transmission range of the nodes concerned. The LET of the wireless link between two nodes *a* and *b* with transmission range T_x, which are moving at velocity V_a and V_b at angles T_a and T_b, respectively (refer to Figure 3.22 (a)), can be estimated as described below: **(3.5.1)** where **(3.5.2) (3.5.3) (3.5.4) (3.5.5)**

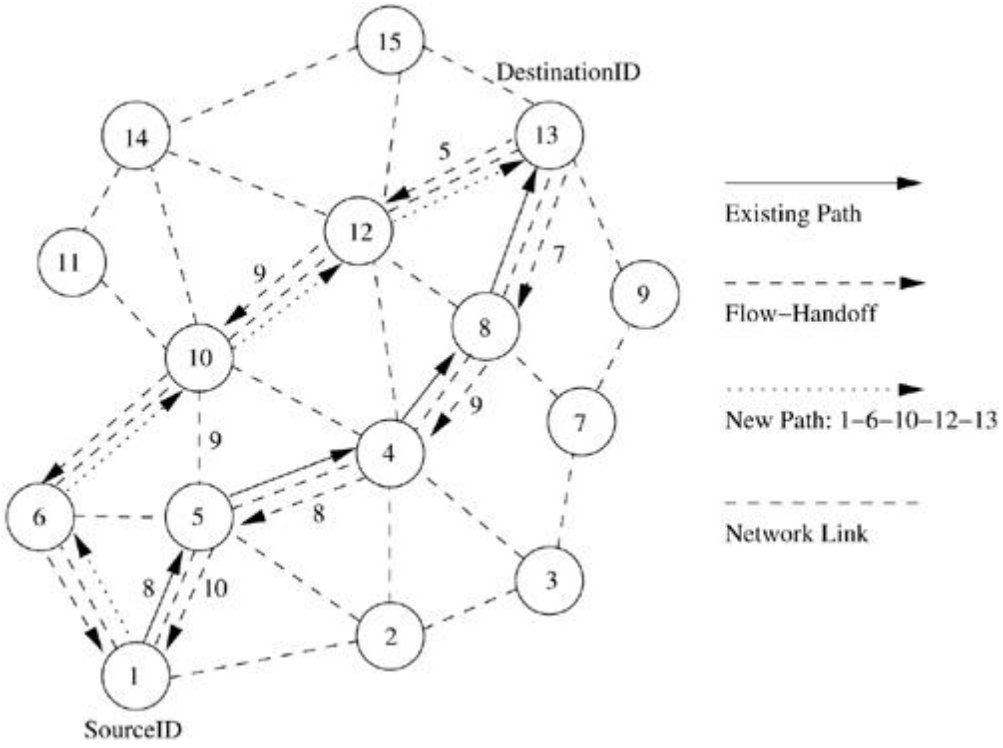
Figure 3.22. Route establishment in FORP.



The route establishment procedure is shown in Figure 3.22 (b). In this case, the path 1-5-4-8-13 (path 1) has a RET value of 7, whereas the path 1-5-10-12-13 (path 2) has a RET value of 5. This indicates that path 1 may last longer than path 2. Hence the sender node originates a *Flow-SETUP* through the reverse path 13-8-4-5-1. FORP employs a proactive route maintenance mechanism which makes use of the expected RET of the current path available at the destination. Route maintenance is illustrated in Figure 3.23. When the destination node determines (using the RET of the current path) that a route break is about to occur within a critical time period (t_c), it originates a *Flow-HANDOFF* packet to the source node, which is forwarded by the intermediate nodes. The mechanism by which *Flow-HANDOFF* packets are forwarded is similar to the *Flow- REQ* forwarding mechanism. When many *Flow-HANDOFF* packets arrive at the source node, the source node calculates the RET values of paths taken by each of them, selects the best path, and uses this new path for sending packets to the destination. In the example shown in Figure 3.23, the *Flow- HANDOFF* packets are forwarded by every intermediate node after

appending the LET information of the previous link traversed onto the packet. The existing path 1-5-4-8-13 is erased and a new path is selected by the source node based on the RETs corresponding to different paths traversed by the *Flow-HANDOFF* packets. In this case, the path 1-6-10-12-13 is chosen. The critical time (t_c) is taken as the difference between the RET and delay encountered by the latest packet which has traversed through the existing path from the source to the destination.

Figure 3.23. Route maintenance in FORP.



Advantages and Disadvantages

The use of LET and RET estimates reduces path breaks and their associated ill effects such as reduction in packet delivery, increase in the number of out-oforder packets, and non-optimal paths resulting from local reconfiguration attempts. The proactive route reconfiguration mechanism adopted here works well when the topology is highly dynamic. The requirements of time synchronization increases the control overhead. Dependency on the GPS infrastructure affects the operability of this protocol in environments where such infrastructure may not be available.

3.6 HYBRID ROUTING PROTOCOLS In hybrid routing protocols, each node maintains the network topology information up to *m* hops. The different existing hybrid protocols are presented below.

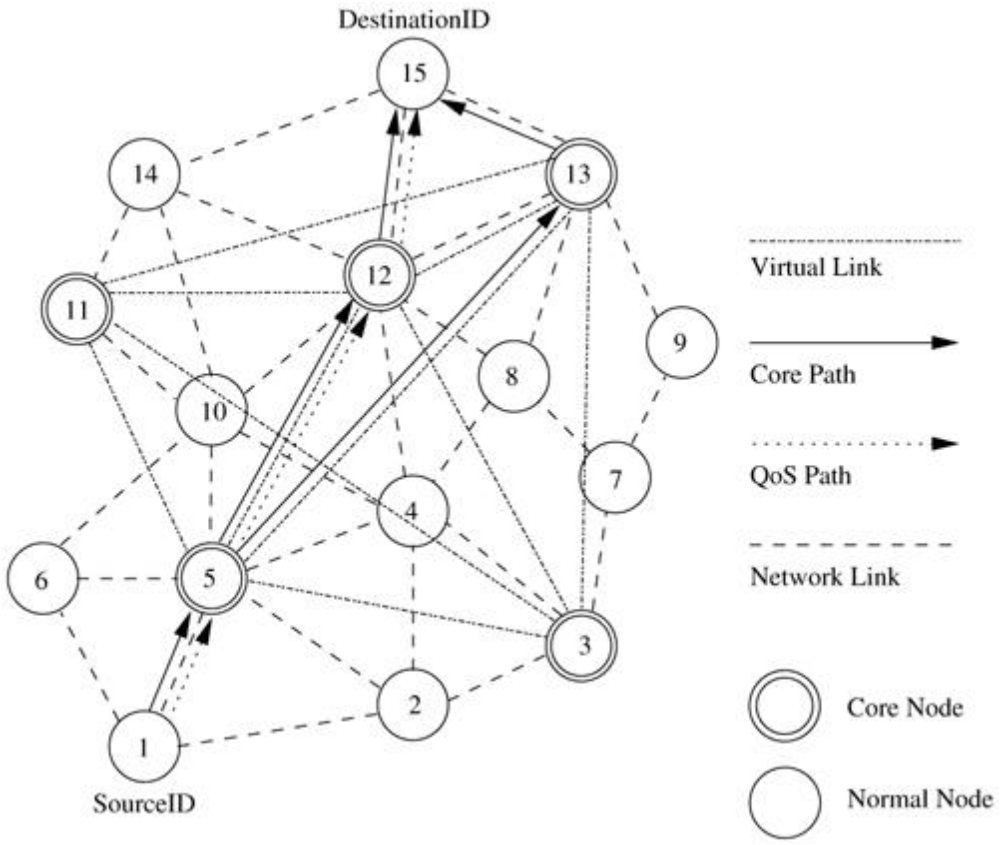
3.6.1 Core Extraction Distributed Ad Hoc Routing Protocol

Core extraction distributed ad hoc routing (CEDAR) integrates routing and support for QoS. It is based on extracting core nodes (also called as dominator nodes) in the network, which together approximate the minimum dominating set. A dominating set (DS) of a graph is defined as a set of nodes in the graph such that every node in the graph is either present in the DS or is a neighbor of some node present in the DS. There exists at least one core node within three hops. The DS of the least cardinality in a graph is called the minimum dominating set. Nodes that choose a core node as their dominating node are called core member nodes of the core node concerned. The path between two core nodes is termed a virtual link. CEDAR employs a distributed algorithm to select core nodes. The selection of core nodes represents the core extraction phase. CEDAR uses the core broadcast mechanism to transmit any packet throughout the network in the unicast mode, involving as minimum number of nodes as possible. These nodes that take part in the core broadcast process are called core nodes. In order to carry out a core broadcast efficiently, each core node must know about its neighboring core nodes. The transfer of information about neighboring core nodes results in significant control overhead at high mobility. When a core node to which many nodes are attached moves away from them, each node has to reselect a new core node. The selection of core nodes, which is similar to the distributed leader election process, involves substantial control overhead. Each core node maintains its neighborhood local topology information. CEDAR employs an efficient link state propagation mechanism in which information regarding the presence of high bandwidth and stable links is propagated to several more nodes, compared to the propagation of information regarding low bandwidth and unstable links, which is suppressed and kept local. To propagate link information, slow-moving increase-waves and fast-moving decrease-waves are used. An increase-wave carrying update information is originated when the bandwidth on the link concerned increases

above a certain threshold. The fast-moving decrease-waves are propagated in order to quickly notify the nearby nodes (core nodes which are at most separated by three hops) about the reduction in available bandwidth. As bandwidth increase information moves slowly, only stable high-bandwidth link state information traverses long distances. If the high-bandwidth link is unstable, then the corresponding increase-wave is overtaken by fast-moving decrease-waves which represent the decrease in available bandwidth on the corresponding link. These waves are very adaptive to the dynamic topology of ad hoc wireless networks. Increase and decrease-waves are initiated only when changes in link capacity cross certain thresholds, that is, only when there is a significant change in link capacity. Fast-moving decrease-waves are prevented from moving across the entire network, thereby suppressing low bandwidth unstable information to the local nodes only. Route establishment in CEDAR is carried out in two phases. The first phase finds a core path from the source to the destination. The core path is defined as a path from the dominator of the source node (source core) to the dominator of the destination node (destination core). In the second phase, a QoS feasible path is found over the core path. A node initiates a *RouteRequest* if the destination is not in the local topology table of its core node; otherwise, the path is immediately established. For establishing a route, the source core initiates a core broadcast in which the *RouteRequest* is sent to all neighboring core nodes as unicast data. Each of these recipient core nodes in turn forwards the *RouteRequest* to its neighboring core nodes if the destination is not its core member. A core node which has the destination node as its core member replies to the source core. Once the core path is established, a path with the required QoS support is then chosen. To find a path that can provide the required QoS, the source core first finds a path to the domain of the farthest possible core node in the core path, which can provide the bandwidth required. Among the available paths to this domain, the source core chooses the shortest-widest path (shortest path with highest bandwidth). Assume *MidCore* is the farthest possible core node found by the source core. In the next iteration, *MidCore* becomes the new source core and finds another *MidCore* node that satisfies the QoS support requirements. This iterative process repeats until a path to the destination with the required bandwidth is found. If no feasible path

is available, the source node is informed about the non-availability of a QoS path. Consider Figure 3.24 where the source is node 1 and the destination is node 15. The core nodes in the network are nodes 3, 5, 11, 12, and 13. In this figure, node 5 is the dominator of nodes 1 and 6. Similarly, node 12 is the dominator of node 15. When node 1 initiates a *RouteRequest* to be flooded throughout the network, it intimates its core node the <source id, destination id> pair information. If the core node 5 does not have any information about the dominator of node 15, which is the destination node, it initiates a core broadcast. Due to this, all nearby core nodes receive the request in the unicast transmission mode. This unicast transmission is done on the virtual links. For core node 5, the virtual link with core node 3 comprises of the links 5-2 and 2-3, while the virtual link between core nodes 5 and 13 is represented by path 5-4-8-13. When a core node receives the core broadcast message, it checks whether the destination is its core member. A core node having the destination as one of its core members replies to the source core node. In our case, core node 12 replies to core node 5. The path between core nodes 12 and 5 constitutes a core path. Once a core path is established, the feasibility of the path in terms of the availability of the required bandwidth is checked. If the required bandwidth is available on the path, the connection is established; otherwise, the core path is rejected.

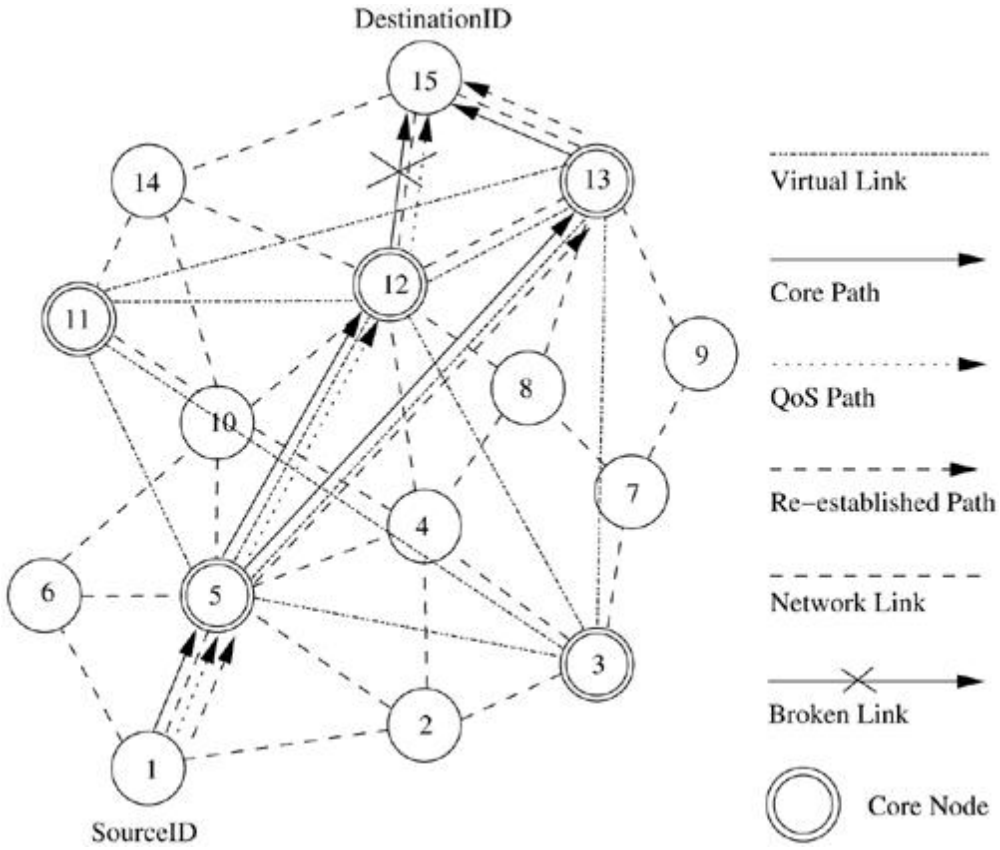
Figure 3.24. Route establishment in CEDAR.



CEDAR attempts to repair a broken route locally when a path break occurs. When a link u-v on the path from source to the destination fails, node u sends back a notification to the source and starts recomputation of a route from itself to the destination node. Until the route recomputation gets completed, node u drops every subsequent packet it receives. Once the source node receives the notification sent by node u, it immediately stops transmitting packets belonging to the corresponding flow, and starts computing a new route to the destination. If the link break occurs near the source, route recomputation at node u may take a long time, but the notification sent by node u reaches the source node very rapidly and prevents large packet losses. If the broken link is very close to the destination, the notification sent by node u might take a longer time to reach the source, but the route recomputation time at node u is small and hence large packet losses are again prevented. If the link break occurs somewhere near the middle of the path, then both the local route recomputation mechanism and the route break notification mechanism are not fast enough, and hence a considerable amount of packet loss occurs in this case. Consider the network

topology shown in Figure 3.25. When the link between nodes 12 and 15 breaks, node 12 tries to reconnect to the destination using an alternate path that satisfies the bandwidth requirement. It also notifies the source node about the link break. The source node tries to reconnect to the destination by reinitiating the route establishment process. In case node 12 does not have any other feasible path, then the alternate path 1-5-4-8-13-15 found by the source node is used for the further routing of packets.

Figure 3.25. Route maintenance in CEDAR.



Advantages and Disadvantages

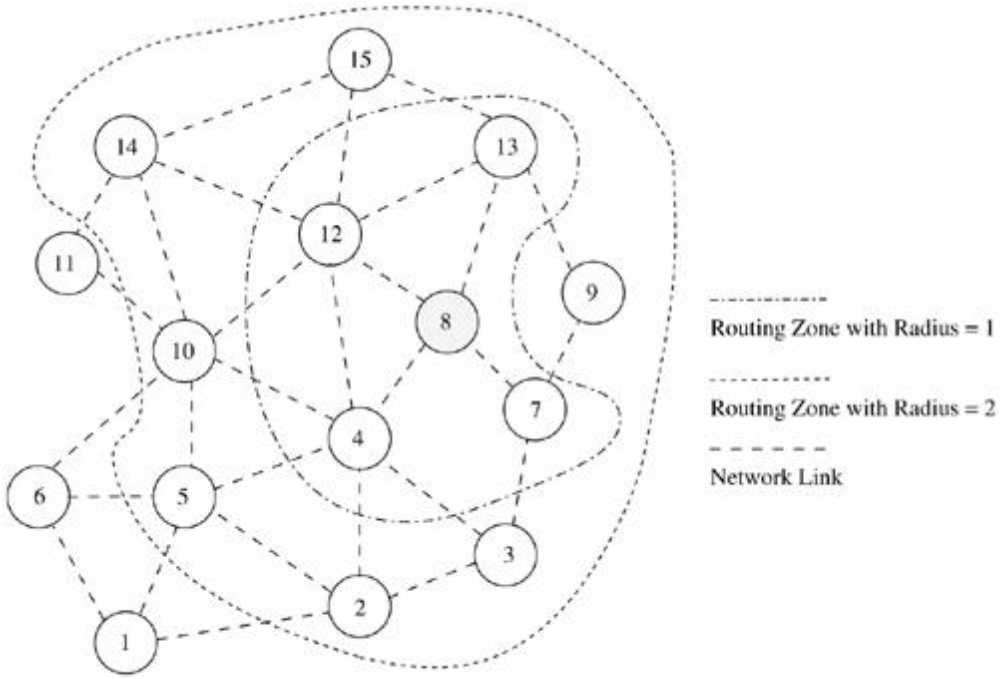
The main advantage of CEDAR is that it performs both routing and QoS path computation very efficiently with the help of core nodes. The increase- and decrease-waves help in appropriate propagation of the stable high-bandwidth link information and the unstable low-bandwidth link information, respectively. Core broadcasts provide a reliable mechanism for establishing paths with QoS support. A disadvantage of this protocol is that since route computation is

carried out at the core nodes only, the movement of the core nodes adversely affects the performance of the protocol. Also, the core node update information could cause a significant amount of control overhead.

3.6.2 Zone Routing Protocol

Zone routing protocol (ZRP) is a hybrid routing protocol which effectively combines the best features of both proactive and reactive routing protocols. The key concept employed in this protocol is to use a proactive routing scheme within a limited zone in the *r*-hop neighborhood of every node, and use a reactive routing scheme for nodes beyond this zone. An *intra-zone routing protocol* (IARP) is used in the zone where a particular node employs proactive routing. The reactive routing protocol used beyond this zone is referred to as *inter-zone routing protocol* (IERP). The *routing zone* of a given node is a subset of the network, within which all nodes are reachable within less than or equal to *zone radius* hops. Figure 3.26 illustrates *routing zones* of node 8, with $r = 1$ hop and $r = 2$ hops. With *zone radius = 2*, the nodes 7, 4, 12, and 13 are interior nodes, whereas nodes 2, 3, 5, 9, 10, 13, and 15 are peripheral nodes (nodes with the shortest distance equal to the *zone radius*). Each node maintains the information about routes to all nodes within its *routing zone* by exchanging periodic route update packets (part of IARP). Hence the larger the *routing zone*, the higher the update control traffic.

Figure 3.26. Routing zone for node 8 in ZRP.

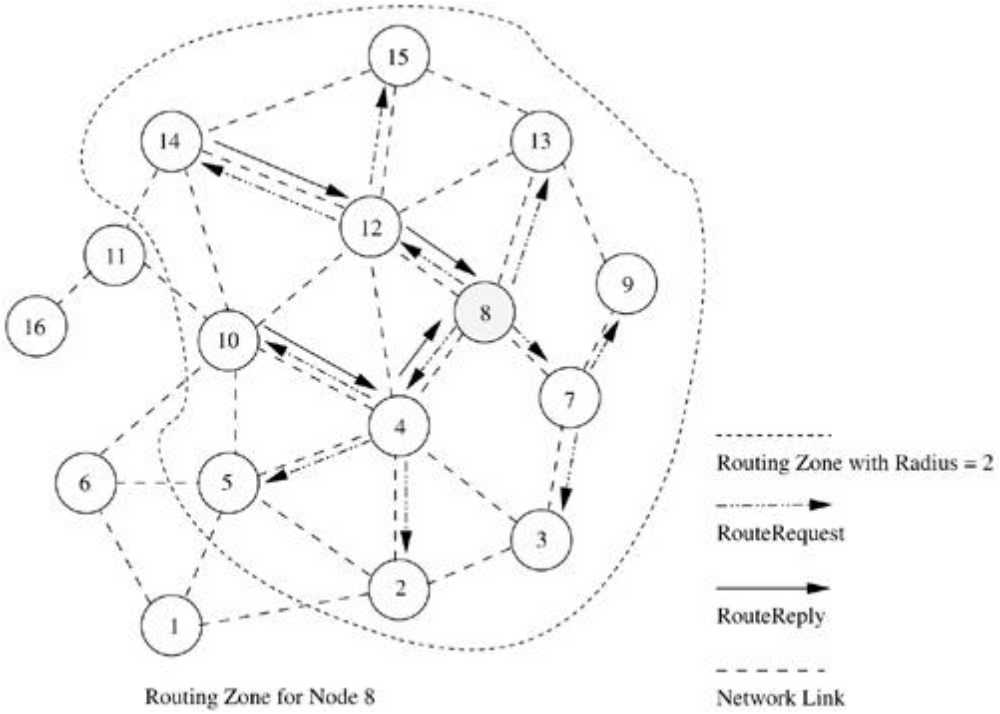


Routing Zone for Node 8

The IERP is responsible for finding paths to the nodes which are not within the *routing zone.* IERP effectively uses the information available at every node's *routing zone.* When a node *s* (node 8 in Figure 3.27) has packets to be sent to a destination node *d* (node 15 in Figure 3.27), it checks whether node *d* is within its zone. If the destination belongs to its own zone, then it delivers the packet directly. Otherwise, node *s* bordercasts (uses unicast routing to deliver packets directly to the border nodes) the *RouteRequest* to its peripheral nodes. In Figure 3.27 node 8 bordercasts *RouteRequest*s to nodes 2, 3, 5, 7, 9, 10, 13, 14, and 15. If any peripheral node finds node *d* to be located within its *routing zone,* it sends a *RouteReply* back to node *s* indicating the path; otherwise, the node rebordercasts the *RouteRequest* packet to the peripheral nodes. This process continues until node *d* is located. Nodes 10 and 14 find the information about node 16 to be available in their intra-zone routing tables, and hence they originate *RouteReply* packets back to node 8. During *RouteRequest* propagation, every node that forwards the *RouteRequest* appends its address to it. This information is used for delivering the *RouteReply* packet back to the source. The path-finding process may result in multiple *RouteReply* packets reaching the source, in which case the source node can choose the best path among them.

The criterion for selecting the best path may be the shortest path, least delay path, etc.

Figure 3.27. Path finding between node 8 and node 16.



When an intermediate node in an active path detects a broken link in the path, it performs a local path reconfiguration in which the broken link is bypassed by means of a short alternate path connecting the ends of the broken link. A path update message is then sent to the sender node to inform it about the change in path. This results in a sub-optimal path between two end points, but achieves quick reconfiguration in case of link failures. To obtain an optimal path, the sender reinitiates the global path-finding process after a number of local reconfigurations.

Advantages and Disadvantages

By combining the best features of proactive and reactive routing schemes, ZRP reduces the control overhead compared to the *RouteRequest* flooding mechanism employed in on-demand approaches and the periodic flooding of routing information packets in table-driven approaches. But in the absence of a query control, ZRP tends to produce higher control overhead than the

aforementioned schemes. This can happen due to the large overlapping of nodes' *routing zones.* The query control must ensure that redundant or duplicate *RouteRequest*s are not forwarded. Also, the decision on the zone radius has a significant impact on the performance of the protocol.

3.6.3 Zone-Based Hierarchical Link State Routing Protocol

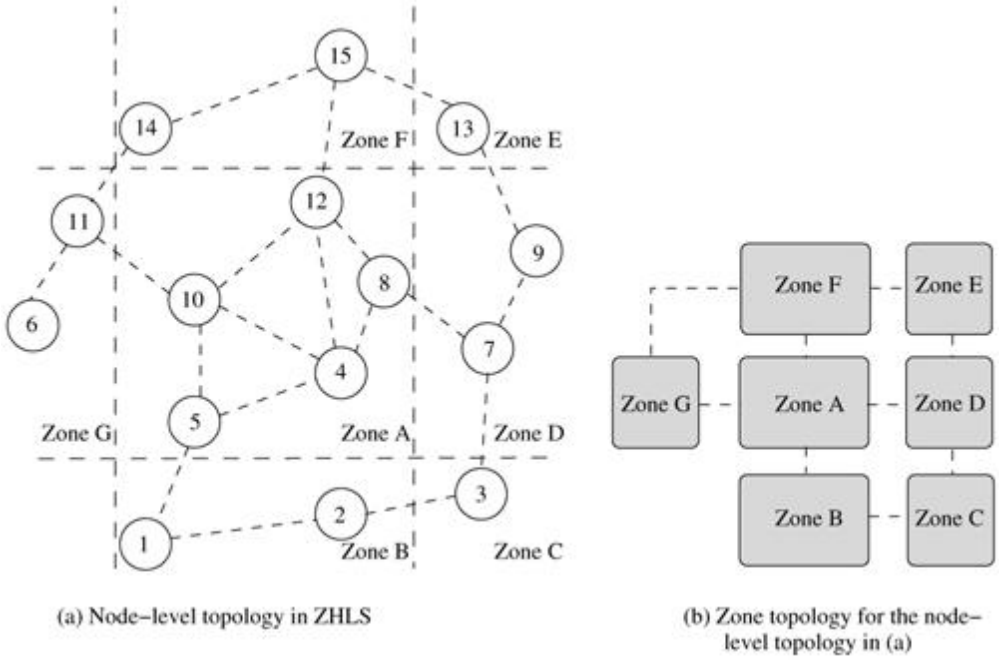
Zone-based hierarchical link state (ZHLS) routing protocol is a hybrid hierarchical routing protocol that uses the geographical location information of the nodes to form non-overlapping zones. A hierarchical addressing that consists of a zone ID and a node ID is employed. Each node requires its location information, based on which it can obtain its zone ID. The information about topology inside a zone is maintained at every node inside the zone, and for regions outside the zone, only the zone connectivity information is maintained. ZHLS maintains high-level hierarchy for inter-zone routing. Packet forwarding is aided by the hierarchical address comprising of the zone ID and node ID. Similar to ZRP, ZHLS also employs a proactive approach inside the geographical zone and a reactive approach beyond the zone. A destination node's current location is identified by the zone ID of the zone in which it is present and is obtained by a route search mechanism. In ZHLS, every node requires GPS or similar infrastructure support for obtaining its own geographical location that is used to map itself onto the corresponding zone. The assignment of zone addresses to geographical areas is important and is done during a phase called the network design phase or network deployment phase. The area of the zone is determined by several factors such as the coverage of a single node, application scenario, mobility of the nodes, and size of the network. For example, the ad hoc network formed by a set of hand-held devices with a limited mobility may require a zone radius of a few hundred meters, whereas the zone area required in the network formed by a set of ships, airplanes, or military tanks may be much larger. The intra-zone routing table is updated by executing the shortest path algorithm on the node-level topology of the zone. The node-level topology is obtained by using the *intra-zone clustering* mechanism, which is similar to the link state updates limited to the nodes present in the zone. Each node builds a one-hop topology by means of a link request and link response

mechanism. Once the one-hop topology is available, each node prepares link state packets and propagates them to all nodes in the zone. These update packets contain the node IDs of all nodes that belong to the zone, and node IDs and zone IDs of all nodes that belong to other zones. The nodes that receive link responses from nodes belonging to other zones are called *gateway* nodes. Data traffic between two zones will be relayed through these gateway nodes. For example, nodes 5, 8, 10, and 12 are the gateway nodes for zone A in Figure 3.28 (a). Every node in a zone is aware of the neighboring zones connected to its zone and the gateway nodes to be used for reaching them. Once the node-level link state packets are exchanged and the node-level topology is updated, every node in a zone generates a zone link state packet. The zone link state packet contains information about the zone-level connectivity. These zone link state packets are propagated throughout the network by the gateway nodes. The zone-level topology is shown in Figure 3.28 (b). The zone link state packets originated by every zone are shown in Table 7.1. Using the information obtained from zone link state packets, a node can build the zone topology. The zone routing table can be formed for any destination zone by executing the shortest path algorithm on the zone-level topology. The zone link state packets are source sequence numbered and a time-stamp field is included to avoid stale link state packets. The association of the nodes to the respective zones helps in reducing routing overhead as in ZRP, but it includes the additional requirement of determining a given destination node's present location. If a source node Src wants to communicate with a destination node Dest, Src checks whether Dest resides in its own zone. If Dest belongs to the same zone, then packets are delivered to Dest as per the intra-zone routing table. If the destination Dest does not belong to the zone, then the node Src originates a location request packet containing the sender's and destination's information. This location request packet is forwarded to every other zone. The gateway node of a zone at which the location request packet is received verifies its routing table for the destination node for which the location request was originated. The gateway node that finds the destination node required by a location request packet originates a location response packet containing the zone information to the sender.

Table 7.1. Zone link state packets

Source Zone	Zone Link State Packet
A	B, D, F, and G
B	C and A
C	B and D
D	A, C, and E
E	A, D, and F
F	A, E, and G
G	A and F

Figure 3.28. Zone-based hierarchical link state routing protocol.



(a) Node–level topology in ZHLS

(b) Zone topology for the node–level topology in (a)

Route maintenance is easier with the presence of multiple gateway nodes between zones. If a given gateway node moves away, causing a zone-level connection failure, routing can still take place with the help of the other gateway nodes. This is due to the hierarchical addressing that makes use of zone ID and node ID. At any intermediate zone, with the most updated inter-zonal routing table, it forwards the data packets.

Advantages and Disadvantages

The hierarchical approach used in this protocol significantly reduces the storage requirements and the communication overhead created because of mobility. The zone-level topology is robust and resilient to path breaks due to mobility of

nodes. Intra-zonal topology changes do not generate network-wide control packet transmissions. A main disadvantage of this protocol is the additional overhead incurred in the creation of the zone-level topology. Also the path to the destination is sub-optimal. The geographical information required for the creation of the zone level topology may not be available in all environments.

3.7 ROUTING PROTOCOLS WITH EFFICIENT FLOODING MECHANISMS

Many of the existing on-demand routing protocols employ flooding of *RouteRequest* packets in order to obtain a feasible path with the required packet-forwarding constraints. Flooding of control packets results in a significant amount of redundancy, wastage of bandwidth, increase in number of collisions, and broadcast storms[1] at times of frequent topological changes. Existing routing protocols that employ efficient flooding mechanisms to counter the requirement of flooding include Preferred Link-Based Routing (PLBR) Protocol and Optimized Link State Routing (OLSR) Protocol. The former belongs to the on-demand routing protocols category and the latter belongs to the table-driven routing protocols category. These protocols utilize algorithms that require a minimum number of transmissions in order to flood the entire network.

3.7.1 Preferred Link-Based Routing Protocols

SSA uses the preferred link approach in an implicit manner by processing a *RouteRequest* packet only if it is received through a strong link. Wired networks also employ preferred links mechanisms, but restrict themselves by selecting a single preferred link, based on heuristics that satisfy multiple constraints, for example, minimum cost and least delay required by the route. In ad hoc networks, the single preferred link model is not suitable due to reasons such as lack of topology information, continuously changing link characteristics, broadcast nature of the radio channel, and mobility of nodes. Sisodia *et al.* proposed two algorithms in known as preferred link-based routing (PLBR) protocols that employ a different approach in which a node selects a subset of nodes from its neighbors list (*NL*). This subset is referred to as the *preferred list* (*PL*). Selection of this subset may be based on link or node characteristics.

Every *RouteRequest* packet carries the list of a selected subset of neighbors. All neighbors receive *RouteRequest* packets because of the broadcast radio channel, but only neighbors present in the PL forward them further. The packet is forwarded by *K* neighbors, where *K* is the maximum number of neighbors allowed in the PL. PLBR aims to minimize control overhead in the ad hoc wireless network. All nodes operate in the promiscuous mode. Each node maintains information about its neighbors and their neighbors in a table called neighbor's neighbor table (*NNT*). It periodically transmits a *beacon* containing the changed neighbor's information. PLBR has three main phases: route establishment, route selection, and route maintenance. The route establishment phase starts when a source node (Src) receives packets from the application layer, meant for a destination node (Dest) for which no route is currently available. If Dest is in Src's*NNT*, the route is established directly. Otherwise, Src transmits a *RouteRequest* packet containing the source node's address (SrcID), destination node's address (DestID), a unique sequence number (SeqNum) (which prevents formation of loops and forwarding of multiple copies of the same*RouteRequest* packet received from different neighbors), a traversed path (*TP*) (containing the list of nodes through which the packet has traversed so far and the weight assigned to the associated links), and a PL. It also contains a time to live (*TTL*) field that is used to avoid packets being present in the network forever, and a *NoDelay* flag, the use of which will be described later in this section. Before forwarding a *RouteRequest* packet, each eligible node recomputes the preferred list table (*PLT*) that contains the list of neighbors in the order of preference. The node inserts the first *K* entries of the *PLT*onto the *PL* field of the *RouteRequest* packet (*K* is a global parameter that indicates the maximum size of *PL*). The old *PL* of a received packet is replaced every time with a new *PL* by the forwarding node. A node is eligible for forwarding a *RouteRequest* only if it satisfies all the following criteria: the node ID must be present in the received *RouteRequest* packet's *PL*, the *RouteRequest* packet must not have been already forwarded by the node, and the *TTL* on the packet must be greater than zero. If Dest is in the eligible node's *NNT*, the *RouteRequest* is forwarded as a unicast packet to the neighbor, which might either be Dest or whose *NL* contains Dest. If there are multiple neighbors whose *NL* have Dest,

the *RouteRequest* is forwarded to only one randomly selected neighbor. Otherwise, the packet is broadcast with a new *PL* computed from the node's *NNT*. *PLT* is computed by means of one of the two algorithms discussed later in this section. If the computed *PLT* is empty, that is, there are no eligible neighbors, the *RouteRequest* packet is discarded and marked as *sent.* If the *RouteRequest* packet reaches the destination, the route is selected by the route selection procedure given below. When multiple *RouteRequest* packets reach Dest, the route selection procedure selects the best route among them. The criterion for selecting the best route can be the shortest path, or the least delay path, or the most stable path. Dest starts a timer after receiving the first *RouteRequest* packet. The timer expires after a certain *RouteSelectWait* period, after which no more *RouteRequest* packets would be accepted. From the received *RouteRequest* packets, a route is selected as follows. For every *RouteRequest i* that reached Dest during the *RouteSelectWait* period, Max( ) is selected, where is the minimum weight of a link in the path followed by *i.* If two or more paths have the same value for *Max* ( ), the shortest path is selected. After selecting a route, all subsequent*RouteRequest* packets from the same Src with a SeqNum less than or equal to the SeqNum of the selected *RouteRequest* are discarded. If the *NoDelay* flag is set, the route selection procedure is omitted and *TP* of the first *RouteRequest* reaching the Dest is selected as the route. The *NoDelay* flag can be set if a fast connection setup is needed. Mobility of nodes causes frequent path breaks that should be locally repaired to reduce broadcast of the*RouteRequest.* The local route repair broadcast mechanism used in ABR has a high failure rate due to the use of restricted *TTL*, which increases the average delay in route reestablishment. PLBR uses a quick route repair mechanism which bypasses the down link (Dest side) node from the broken path, using information about the next two hops from *NNT*. Algorithms for Preferred Links Computation

Two different algorithms have been proposed by Sisodia *et al.* in , for finding preferred links. The first algorithm selects the route based on degree of neighbor nodes (degree of a node is the number of neighbors). Preference is given to nodes whose neighbor degree is higher. As higher degree neighbors cover more nodes, only a few of them are required to cover all the nodes of the *NNT*. This

reduces the number of broadcasts. The second algorithm gives preference to stable links. Links are not explicitly classified as stable or unstable. The notion of stability is based on the weight given to links.

Neighbor Degree-Based Preferred Link Algorithm (NDPL)

Let *d* be the node that calculates the preferred list table *PLT*. *TP* is the traversed path and OLD_{PL} is the preferred list of the received *RouteRequest* packet. The *NNT* of node *d* is denoted by NNT_d. *N*(*i*) denotes the neighbors of node *i* and itself. Include list (*INL*) is a set containing all neighbors reachable by transmitting the *RouteRequest* packet after execution of the algorithm, and the exclude list (*EXL*) is a set containing all neighbors that are unreachable by transmitting the *RouteRequest* packet after execution of the algorithm. 1. In this step, node *d* marks the nodes that are not eligible for further forwarding the *RouteRequest*packet. 1. If a node *i* of *TP* is a neighbor of node *d*, mark all neighbors of *i* as reachable, that is, add *N*(*i*) to*INL*. 2. If a node *i* of OLD_{PL} is a neighbor of node *d*, and $i < d$, mark all neighbors of node *i* as reachable, that is, include *N*(*i*) in *INL*. 3. If neighbor *i* of node *d* has a neighbor *n* present in *TP*, mark all neighbors of *i* as reachable by adding *N*(*i*) to *INL*. 4. If neighbor *i* of node *d* has a neighbor *n* present in OLD_{PL}, and $n < d$, here again add *N*(*i*) to *INL*, thereby marking all neighbors of node *i* as reachable. 2. If neighbor *i* of node *d* is not in *INL*, put *i* in preferred list table *PLT* and mark all neighbors of *i* as reachable. If *i* is present in *INL*, mark the neighbors of *i* as unreachable by adding them to *EXL*, as*N*(*i*) may not be included in this step. Here neighbors *i* of *d* are processed in decreasing order of their degrees. After execution of this step, the *RouteRequest* is guaranteed to reach all neighbors of*d*. If *EXL* is not empty, some neighbor's neighbors *n* of node *d* are currently unreachable, and they are included in the next step. 3. If neighbor *i* of *d* has a neighbor *n* present in *EXL*, put *i* in *PLT* and mark all neighbors of *i* as reachable. Delete all neighbors of *i* from *EXL*. Neighbors are processed in decreasing order of their degrees. After execution of this step, all the nodes in NNT_d are reachable. Apply reduction steps to remove overlapping neighbors from *PLT* without compromising on reachability. 4. Reduction steps are applied here in order to remove overlapping neighbors from *PLT* without compromising on reachability. 1. Remove each neighbor *i* from *PLT* if *N*(*i*) is covered by remaining neighbors of *PLT*. Here the

minimum degree neighbor is selected every time. 2. Remove neighbor *i* from *PLT* whose *N*(*i*) is covered by node *d* itself.

Weight-Based Preferred Link Algorithm (WBPL) In this algorithm, a node finds the preferred links based on stability, which is indicated by a weight, which in turn is based on its neighbors' temporal and spatial stability.

1. Let $BCnt_i$ be the count of *beacon*s received from a neighbor *i* and TH_{bcon} is the number of beacons generated during a time period equal to that required to cover twice the transmission range . Weight given to *i* based on time stability is

2. Estimate the distance ( ) to *i* from the received power of the last few packets using appropriate propagation models. The weight based on spatial stability is .

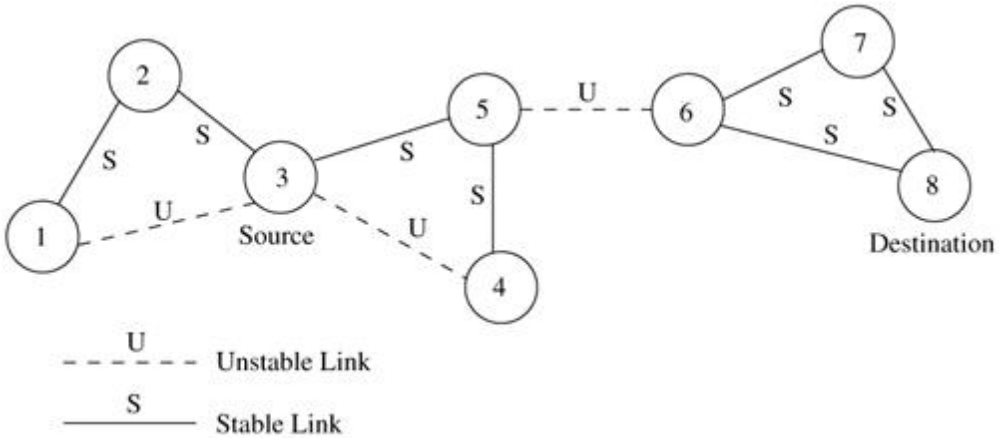
3. The weight assigned to the link *i* is the combined weight given to time stability and spatial stability.

4. Arrange the neighbors in a non-increasing order of their weights. The nodes are put into the *PLT* in this order.

5. If a link is overloaded, delete the associated neighbor from *PLT*. Execute *Step* 1 of NDPL and delete $\forall i, i \in PLT \cap i \in INL$. Also, delete those neighbors from *PLT* that satisfy *Step* 4 of NDPL. Consider, for example, Figure 3.29, where the node 3 is the source and node 8 is the destination. *S* and*U* denote stable and unstable links. In WBPL and NDPL, the source that initiates that *RouteRequest* as Dest is not present in *NNT* and computes the preferred link table (*PLT*). Let *K* = 2 be the preferred list'ssize. In NDPL, after *Step* 2 the *PLT* becomes {5, 1}, and after *Step* 3 also the *PLT* remains {5, 1}. In reduction *Step* 4*b*, neighbor 1 is deleted from *PLT* and hence node 3 sends the *RouteRequest* only to neighbor 5. In WBPL, the weights are assigned to all neighbors according to *Steps* 1, 2, and 3, and all neighbors are in *PLT*. In *Step* 5, neighbors 1, 4, and 2 are deleted from *PLT* due to *Step* 4*a* and 4*b* ofNDPL and hence only node 5 remains. Now the *RouteRequest* can be sent as a unicast packet to avoid broadcast. If it is broadcast, all the nodes receive the packet, but only node 5 can further forward it. As Dest 8 is present in node 5's *NNT*, it directly sends it to node 6 which forwards it to Dest. Here only three packets are transmitted for finding the route

and the path length is 3. Now consider SSA. After broadcasts by nodes 3 and 5, the *RouteRequest* packet reaches node 6, where it is rejected and hence the *RouteRequest* fails to find a route. After timeout, it sets a flag indicating processed by *all* and hence finds the same route as WBPL and NDPL.

Figure 3.29. Example network. *Reproduced with permission from* , © Korean Institute of Communication Sciences, 2002.



Advantages and Disadvantages

The efficient flooding mechanism employed in this protocol minimizes the broadcast storm problem prevalent in on-demand routing protocols. Hence this protocol has higher scalability compared to other on-demand routing protocols. The reduction in control overhead results in a decrease in the number of collisions and improvement in the efficiency of the protocol. PLBR achieves bandwidth efficiency at the cost of increased computation. Both NDPL and WBPL are computationally more complex than other *RouteRequest* forwarding schemes.

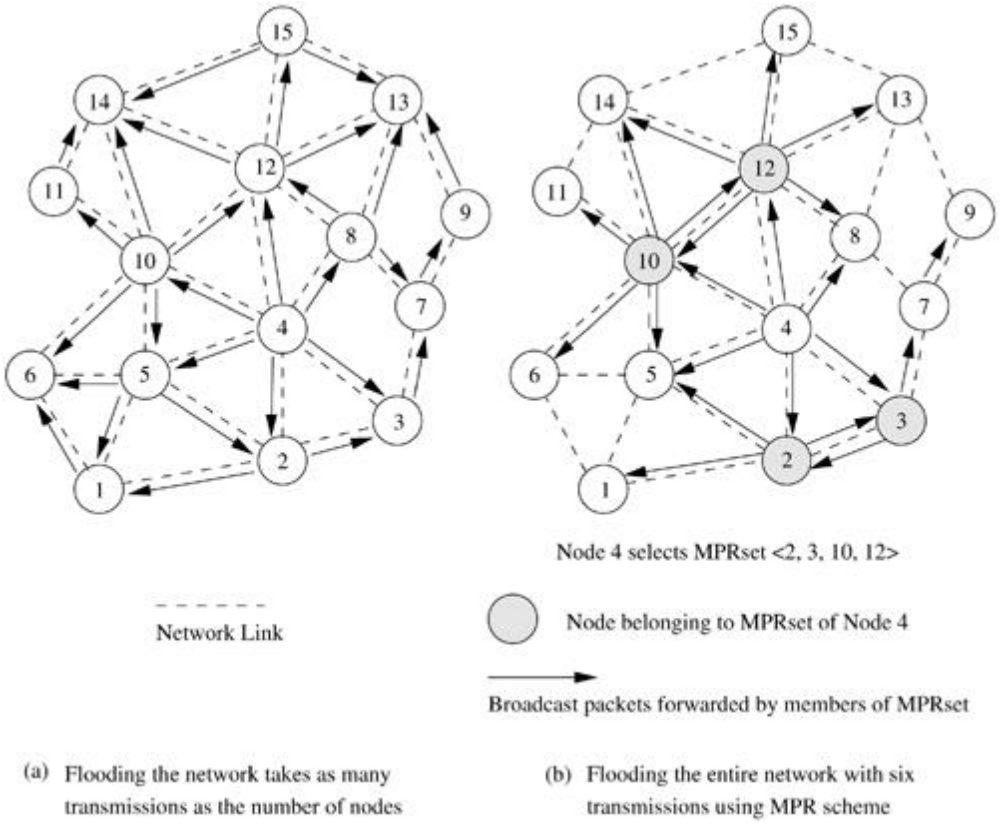
3.3.2 Optimized Link State Routing

The optimized link state routing (OLSR) protocol is a proactive routing protocol that employs an efficient link state packet forwarding mechanism called *multipoint relaying.* This protocol optimizes the pure link state routing protocol. Optimizations are done in two ways: by reducing the size of the control packets and by reducing the number of links that are used for forwarding the link state packets. The reduction in the size of link state packets is made by declaring only

a subset of the links in the link state updates. This subset of links or neighbors that are designated for link state updates and are assigned the responsibility of packet forwarding are called *multipoint relays.* The optimization by the use of *multipoint relaying* facilitates periodic link state updates. The link state update mechanism does not generate any other control packet when a link breaks or when a link is newly added. The link state update optimization achieves higher efficiency when operating in highly dense networks. Figure 3.30 (a) shows the number of message transmissions required when the typical flooding-based approach is employed. In this case, the number of message transmissions is approximately equal to the number of nodes that constitute the network. The set consisting of nodes that are *multipoint relays* is referred to as *MPRset.* Each node (say, *P*) in the network selects an *MPRset* that processes and forwards every link state packet that node *P* originates. The neighbor nodes that do not belong to the *MPRset* process the link state packets originated by node *P* but do not forward them. Similarly, each node maintains a subset of neighbors called *MPR selectors,* which is nothing but the set of neighbors that have selected the node as a *multipoint relay.* A node forwards packets that are received from nodes belonging to its *MPRSelector* set. The members of both *MPRset* and *MPRSelectors* keep changing over time. The members of the *MPRset* of a node are selected in such a manner that every node in the node's two-hop neighborhood has a bidirectional link with the node. The selection of nodes that constitute the *MPRset* significantly affects the performance of OLSR because a node calculates routes to all destinations only through the members of its *MPRset.* Every node periodically broadcasts its *MPRSelector* set to nodes in its immediate neighborhood. In order to decide on the membership of the nodes in the *MPRset,* a node periodically sends *Hello* messages that contain the list of neighbors with which the node has bidirectional links and the list of neighbors whose transmissions were received in the recent past but with whom bidirectional links have not yet been confirmed. The nodes that receive this *Hello* packet update their own two-hop topology tables. The selection of *multipoint relays* is also indicated in the *Hello* packet. A data structure called *neighbor table* is used to store the list of neighbors, the two-hop neighbors, and the status of neighbor nodes. The neighbor nodes can be in one of the three

possible link status states, that is, unidirectional, bidirectional, and *multipoint relay.* In order to remove the stale entries from the *neighbor table*, every entry has an associated timeout value, which, when expired, removes the table entry. Similarly a sequence number is attached with the *MPRset* which gets incremented with every new *MPRset.*

Figure 3.30. An example selection of MPRset in OLSR.



The *MPRset* need not be optimal, and during initialization of the network it may be same as the neighbor set. The smaller the number of nodes in the *MPRset*, the higher the efficiency of protocol compared to link state routing. Every node periodically originates *topology control* (TC) packets that contain topology information with which the routing table is updated. These TC packets contain the *MPRSelector* set of every node and are flooded throughout the network using the *multipoint relaying* mechanism. Every node in the network receives several such TC packets from different nodes, and by using the information contained in the TC packets, the *topology table* is built. A TC message may be originated by a node earlier than its regular period if there is a change in the

MPRSelector set after the previous transmission and a minimal time has elapsed after that. An entry in the *topology table* contains a destination node which is the *MPRSelector* and a last-hop node to that destination, which is the node that originates the TC packet. Hence, the routing table maintains routes for all other nodes in the network.

Selection of Multipoint Relay Nodes Figure 3.30 (b) shows the forwarding of TC packets using the *MPRset* of node 4. In this example, node 4 selects the nodes 2, 3, 10, and 12 as members of its *MPRset.* Forwarding by these nodes makes the TCpackets reach all nodes within the transmitting node's two-hop local topology. The selection of the optimal *MPRset* is NP-complete . In , a heuristic has been proposed for selecting the *MPRset.*The notations used in this heuristic are as follows: $N_i(x)$ refers to the i_{th} hop neighbor set of node x and$MPR(x)$ refers to the *MPRset* of node x. 1. $MPR(x) \leftarrow \emptyset$/* Initializing empty *MPRset* */ 2. $MPR(x) \leftarrow$ { Those nodes that belong to $N_1(x)$ and which are the only neighbors of nodes in $N_2(x)$ } 3. While there exists some node in $N_2(x)$ which is not covered by $MPR(x)$ 1. For each node in $N_1(x)$, which is not in $MPR(x)$, compute the maximum number of nodes that it covers among the uncovered nodes in the set $N_2(x)$. 2. Add to $MPR(x)$ the node belonging to $N_1(x)$, for which this number is maximum. A node updates its *MPRset* whenever it detects a new bidirectional link in its neighborhood or in its two-hop topology, or a bidirectional link gets broken in its neighborhood.

Advantages and Disadvantages

OLSR has several advantages that make it a better choice over other tabledriven protocols. It reduces the routing overhead associated with table-driven routing, in addition to reducing the number of broadcasts done. Hence OLSR has the advantages of low connection setup time and reduced control overhead.

3.8 HIERARCHICAL ROUTING PROTOCOLS

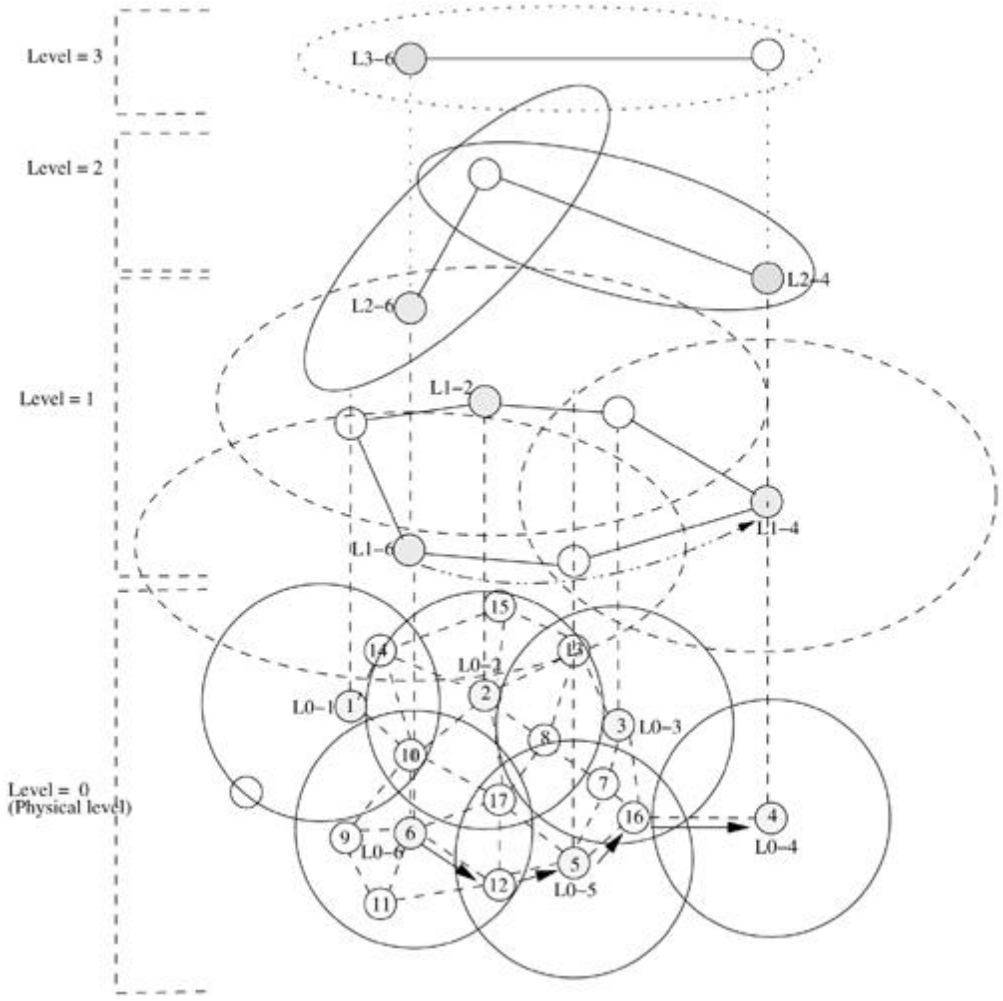
The use of routing hierarchy has several advantages, the most important ones being reduction in the size of routing tables and better scalability. This section discusses the existing hierarchical routing protocols for ad hoc wireless networks.

3.8.1 Hierarchical State Routing Protocol

The hierarchical state routing (HSR) protocol is a distributed multi-level hierarchical routing protocol that employs clustering at different levels with efficient membership management at every level of clustering. The use of clustering enhances resource allocation and management. For example, the allocation of different frequencies or spreading codes to different clusters can improve the overall spectrum reuse. HSR operates by classifying different levels of clusters. Elected leaders at every level form the members at the immediate higher level. Different clustering algorithms, such as the one proposed in , are employed for electing leaders at every level. The first level of physical clustering is done among the nodes that are reachable in a single wireless hop. The next higher level of physical clustering is done among the nodes that are elected as leaders of each of these first-level clusters. In addition to the *physical clustering*, a *logical clustering* scheme has been proposed in HSR, which is based on certain relations among the nodes rather than on their geographical positions, as in the case of *physical clustering.* Figure 3.31 illustrates the multilayer clustering defined by the HSR protocol. At the lowest level *(L* = 0), there are six cluster leaders (nodes 1, 2, 3, 4, 5, and 6). Nodes are classified as cluster leaders, or gateway nodes, or normal member nodes. A cluster leader is entrusted with responsibilities such as slot/frequency/code allocation, call admission control, scheduling of packet transmissions, exchange of routing information, and handling route breaks. In Figure 3.31, node 5 is a clusterhead marked as *L*0-5, which refers to the level of clustering (*L* = 0) and node ID (5). Similarly, each of the higher-level cluster leaders is also marked (*e.g.*, $L1-6$, $L-2-6$, and $L3-6$ refer to the same node 6, but acting as leader with the given leader IDs at levels 1, 2, and 3, respectively). The spectrum reuse schemes, including spreading code assignment, can be used among the cluster leaders of the *L* = 0 clusters. For the nodes under the leadership of node 6 at level 0, the cluster members are nodes 9, 10, 11, 12, and 17. Those nodes that belong to multiple clusters are referred to as cluster gateway nodes. For the level 0 cluster whose leader is node 6, the cluster gateways are nodes 10, 12, and 17. The second level of clustering is done among the leaders of the first level, that

is, the leaders of 0_{th} level clusters, *L*0 - 1, *L*0 - 2, *L0* - 3,*L0* - 4, *L0* - 5, and *L0* - 6, form the members of the first-level cluster.

Figure 3.31. Example of HSR multi-level clustering.



Every node maintains information about all its neighbors and the status of links to each of them. This information is broadcast within the cluster at regular intervals. The cluster leader exchanges the topology and link state routing information among its peers in the neighborhood clusters, using which the nexthigher- level clustering is performed. This exchange of link state information is done over multiple hops that consist of gateway nodes and cluster-heads. The path between two cluster-heads which is formed by multiple wireless links is called a *virtual link.* The link status for the *virtual link*(otherwise called *tunnel*) is obtained from the link status parameters of the

wireless links that constitute the *virtual link.* In Figure 3.31, the path between first-level clusters *L1* - 6 and *L1* - 4 includes the wireless links 6 - 12 - 5 - 16 - 4. The clustering is done recursively to the higher levels. At any level, the cluster leader exchanges topology information with its peers. After obtaining information from its peers, it floods the information to the lower levels, making every node obtain the hierarchical topology information. This hierarchical topology necessitates a hierarchical addressing which helps in operating with less routing information against the full topology exchange required in the link state routing. The hierarchical addressing defined in HSR includes the hierarchical ID (HID) and node ID. The HID is a sequence of IDs of cluster leaders of all levels starting from the highest level to the current node. This ID of a node in HSR is similar to the unique MAC layer address. The hierarchical addresses are stored in an HSR table at every node that indicates the node's own position in the hierarchy. The HSR table is updated whenever routing update packets are received by the node. The hierarchical address of node 11 in Figure 3.31 is < 6, 6, 6, 6, 11 >, where the last entry (11) is the node ID and the rest consists of the node IDs of the cluster leaders that represent the location of node 11 in the hierarchy. Similarly, the HID of node 4 is < 6, 4, 4, 4 >. When node 11 needs to send a packet to node 4, the packet is forwarded to the highest node in the hierarchy (node 6). Node 6 delivers the packet to node 4, which is at the top-most level of the hierarchy.

Advantages and Disadvantages

The HSR protocol reduces the routing table size by making use of hierarchy information. In HSR, the storage required is $O(n \times m)$ compared to $O(n_m)$ that is required for a flat topology link state routing protocol (n is the average number of nodes in a cluster and m is the number of levels). Though the reduction in the amount of routing information stored at nodes is appreciable, the overhead involved in exchanging packets containing information about the multiple levels of hierarchy and the leader election process make the protocol unaffordable in the ad hoc wireless networks context. Besides, the number of nodes that participate in an ad hoc wireless network does not grow to the dimensions of the number of nodes in the Internet where the hierarchical routing is better suited. In the military applications of ad hoc wireless networks, the hierarchy of routing

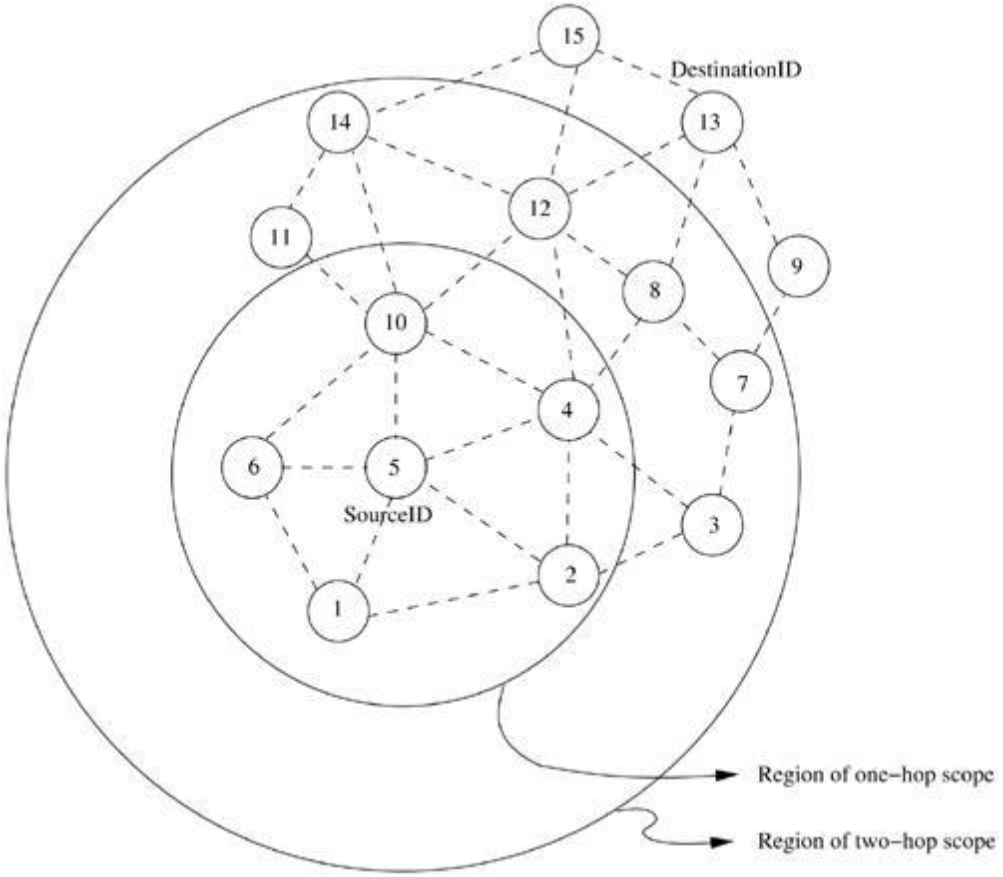
assumes significance where devices with higher capabilities of communication can act as the cluster leaders.

3.8.2 Fisheye State Routing Protocol

The table-driven routing protocols generate routing overhead that is dependent on the size of the network and mobility of the nodes, whereas the routing overhead generated by on-demand routing protocols are dependent on the number of connections present in the system in addition to the above two factors. Hence, as the number of senders in the network increases, the routing overhead also increases. ZRP uses an intra-zone proactive approach and an inter-zone reactive approach to reduce control overhead. The fisheye state routing (FSR) protocol is a generalization of the GSR protocol. FSR uses the *fisheye* technique to reduce information required to represent graphical data, to reduce routing overhead. The basic principle behind this technique is the property of a fish's eye that can capture pixel information with greater accuracy near its eye's focal point. This accuracy decreases with an increase in the distance from the center of the focal point. This property is translated to routing in ad hoc wireless networks by a node, keeping accurate information about nodes in its local topology, and not-so-accurate information about far-away nodes, the accuracy of the network information decreasing with increasing distance. FSR maintains the topology of the network at every node, but does not flood the entire network with the information, as is done in link state routing protocols. Instead of flooding, a node exchanges topology information only with its neighbors. A sequence numbering scheme is used to identify the recent topology changes. This constitutes a hybrid approach comprising of the link-level information exchange of distance vector protocols and the complete topology information exchange of link state protocols. The complete topology information of the network is maintained at every node and the desired shortest paths are computed as required. The topology information exchange takes place periodically rather than being driven by an event. This is because instability of the wireless links may cause excessive control overhead when event-driven updates are employed. FSR defines routing scope, which is the set of nodes that are reachable in a specific number of hops. The scope of a node at two hops is

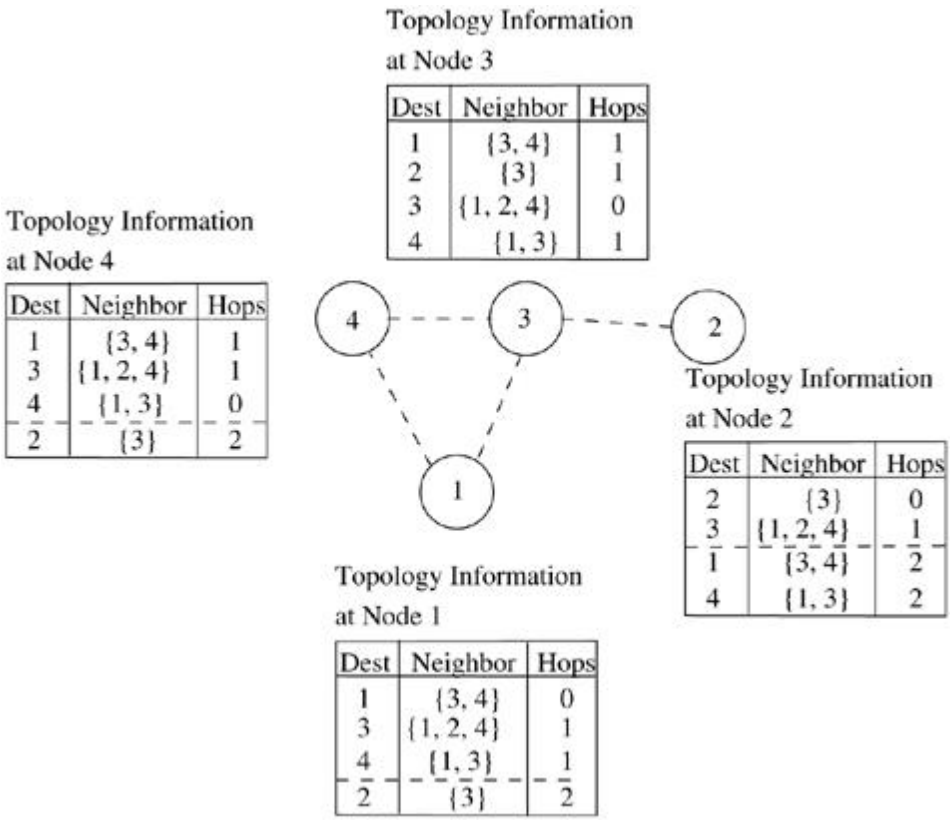
the set of nodes that can be reached in two hops. Figure 3.32shows the scope of node 5 with one hop and two hops. The routing overhead is significantly reduced by adopting different frequencies of updates for nodes belonging to different scopes.

Figure 3.32. Fisheye state routing.



The link state information for the nodes belonging to the smallest scope is exchanged at the highest frequency. The frequency of exchanges decreases with an increase in scope. This keeps the immediate neighborhood topology information maintained at a node more precise compared to the information about nodes farther away from it. Thus the message size for a typical topology information update packet is significantly reduced due to the removal of topology information regarding the far-away nodes. The path information for a distant node may be inaccurate as there can be staleness in the information. But this is compensated by the fact that the route gets more and more accurate as the packet nears its destination. FSR scales well for large ad hoc wireless networks

because of the reduction in routing overhead due to the use of the abovedescribed mechanism, where varying frequencies of updates are used. Figure 3.33 illustrates an example depicting the network topology information maintained at nodes in a network. The routing information for the nodes that are one hop away from a node are exchanged more frequently than the routing information about nodes that are more than one hop away. Information regarding nodes that are more than one hop away from the current node are listed below the dotted line in the topology table. **Figure 3.33. An illustration of routing tables in FSR.**



Advantages and Disadvantages

The notion of multi-level scopes employed by FSR significantly reduces the bandwidth consumed by link state update packets. Hence, FSR is suitable for large and highly mobile ad hoc wireless networks. The choice of the number of hops associated with each scope level has a significant influence on the performance of the protocol at different mobility values, and hence must be carefully chosen.

3.9 POWER-AWARE ROUTING PROTOCOLS

In a deviation from the traditional wired network routing and cellular wireless network routing, power consumption by the nodes is a serious factor to be taken into consideration by routing protocols for ad hoc wireless networks. This is because, in ad hoc wireless networks, the routers are also equally power constrained just as the nodes are. This section discusses some of the important routing metrics that take into consideration this energy factor.

3.9.1 Power-Aware Routing Metrics

The limitation on the availability of power for operation is a significant bottleneck, given the requirements of portability, weight, and size of commercial hand-held devices. Hence, the use of routing metrics that consider the capabilities of the power sources of the network nodes contributes to the efficient utilization of energy and increases the lifetime of the network. Singh *et al.* proposed a set of routing metrics in that supports conservation of battery power. The routing protocols that select paths so as to conserve power must be aware of the states of the batteries at the given node as well as at the other intermediate nodes in the path.

Minimal Energy Consumption per Packet

This metric aims at minimizing the power consumed by a packet in traversing from source node to the destination node. The energy consumed by a packet when traversing through a path is the sum of the energies required at every intermediate hop in that path. The energy consumed at an intermediate hop is a function of the distance between the nodes that form the link and the load on that link. This metric does not balance the load so that uniform consumption of power is maintained throughout the network. The disadvantages of this metric include selection of paths with large hop length, inability to measure the power consumption at a link in advance when the load varies, and the inability to prevent the fast discharging of batteries at some nodes.

Maximize Network Connectivity

This metric attempts to balance the routing load among the *cut-set* (the subset of the nodes in the network, the removal of which results in network partitions). This assumes significance in environments where network connectivity is to be ensured by uniformly distributing the routing load among the *cut-set.* With a variable traffic origination rate and unbounded contention in the network, it is difficult to achieve a uniform battery draining rate for the *cut-set.*

Minimum Variance in Node Power Levels

This metric proposes to distribute the load among all nodes in the network so that the power consumption pattern remains uniform across them. This problem is very complex when the rate and size of data packets vary. A nearly optimal performance can be achieved by routing packets to the least-loaded next-hop node.

Minimum Cost per Packet

In order to maximize the life of every node in the network, this routing metric is made as a function of the state of the node's battery. A node's cost decreases with an increase in its battery charge and vice versa. Translation of the remaining battery charge to a cost factor is used for routing. With the availability of a battery discharge pattern, the cost of a node can be computed. This metric has the advantage of ease in the calculation of the cost of a node and at the same time congestion handling is done.

Minimize Maximum Node Cost

This metric minimizes the maximum cost per node for a packet after routing a number of packets or after a specific period. This delays the failure of a node, occurring due to higher discharge because of packet forwarding.

TRANSPORT LAYER AND SECURITY PROTOCOLS FOR AD HOC WIRELESS NETWORKS

3.10 INTRODUCTION

The objectives of a transport layer protocol include the setting up of an end-to end connection, end-to-end delivery of data packets, flow control, and congestion control. There exist simple, unreliable, and connection-less transport layer protocols such as UDP, and reliable, byte-stream-based, and connection oriented transport layer protocols such as TCP for wired networks. These traditional wired transport layer protocols are not suitable for ad hoc wireless networks due to the inherent problems associated with the latter. The first half of this chapter discusses the issues and challenges in designing a transport layer protocol for ad hoc wireless networks, the reasons for performance degradation when TCP is employed in ad hoc wireless networks, and it also discusses some of the existing TCP extensions and other transport layer protocols for ad hoc wireless networks. The previous chapters discussed various networking protocols for ad hoc wireless networks. However, almost all of them did not take into consideration one very important aspect of communication: security. Due to the unique characteristics of ad hoc wireless networks, which have been mentioned in the previous chapters, such networks are highly vulnerable to security attacks compared to wired networks or infrastructure-based wireless networks (such as cellular networks). Therefore, security protocols being used

in the other networks (wired networks and infrastructure-based wireless networks) cannot be directly applied to ad hoc wireless networks. The second half of this chapter focuses on the security aspect of communication in ad hoc wireless networks. Some of the recently proposed protocols for achieving secure communication are discussed.

3.11 ISSUES IN DESIGNING A TRANSPORT LAYER PROTOCOL FOR AD HOC WIRELESS NETWORKS

In this section, some of the issues to be considered while designing a transport layer protocol for ad hoc wireless networks are discussed.

• **Induced traffic:** Unlike wired networks, ad hoc wireless networks utilize multi-hop radio relaying. A link-level transmission affects the neighbor nodes of both the sender and receiver of the link. In a path having multiple links, transmission at a particular link affects one upstream link and one downstream link. This traffic at any given link (or path) due to the traffic through neighboring links (or paths) is referred to as induced traffic. This is due to the broadcast nature of the channel and the location-dependent contention on the channel. This induced traffic affects the throughput achieved by the transport layer protocol.

• **Induced throughput unfairness:** This refers to the throughput unfairness at the transport layer due to the throughput/delay unfairness existing at the lower layers such as the network and MAClayers. For example, an ad hoc wireless network that uses IEEE 802.11 DCF as the MAC protocol may experience throughput unfairness at the transport layer as well. A transport layer protocol should consider these in order to provide a fair share of throughput across contending flows.

• **Separation of congestion control, reliability, and flow control:** A transport layer protocol can provide better performance if end-to-end reliability, flow control, and congestion control are handled separately. Reliability and flow control are end-to-end activities, whereas congestion can at times be a local activity. The transport layer flow can experience congestion with just one intermediate link under congestion. Hence, in networks such as ad hoc wireless

networks, the performance of the transport layer may be improved if these are separately handled. While separating these, the most important objective to be considered is the minimization of the additional control overhead generated by them.

• **Power and bandwidth constraints:** Nodes in ad hoc wireless networks face resource constraints including the two most important resources: (i) power source and (ii) bandwidth. The performance of a transport layer protocol is significantly affected by these constraints.

• **Misinterpretation of congestion:** Traditional mechanisms of detecting congestion in networks, such as packet loss and retransmission timeout, are not suitable for detecting the network congestion in ad hoc wireless networks. This is because the high error rates of wireless channel, location-dependent contention, hidden terminal problem, packet collisions in the network, path breaks due to the mobility of nodes, and node failure due to a drained battery can also lead to packet loss in ad hoc wireless networks. Hence, interpretation of network congestion as used in traditional networks is not appropriate in ad hoc wireless networks.

• **Completely decoupled transport layer:** Another challenge faced by a transport layer protocol is the interaction with the lower layers. Wired network transport layer protocols are almost completely decoupled from the lower layers. In ad hoc wireless networks, the cross-layer interaction between the transport layer and lower layers such as the network layer and the MAC layer is important for the transport layer to adapt to the changing network environment.

• **Dynamic topology:** Some of the deployment scenarios of ad hoc wireless networks experience rapidly changing network topology due to the mobility of nodes. This can lead to frequent path breaks, partitioning and remerging of networks, and high delay in reestablishment of paths. Hence, the performance of a transport layer protocol is significantly affected by the rapid changes in the network topology.

3.12 DESIGN GOALS OF A TRANSPORT LAYER PROTOCOL FOR AD HOC WIRELESS NETWORKS

The following are the important goals to be met while designing a transport layer protocol for ad hoc wireless networks:

• The protocol should maximize the throughput per connection.

• It should provide throughput fairness across contending flows.

• The protocol should incur minimum connection setup and connection maintenance overheads. It should minimize the resource requirements for setting up and maintaining the connection in order to make the protocol scalable in large networks.

• The transport layer protocol should have mechanisms for congestion control and flow control in the network.

• It should be able to provide both reliable and unreliable connections as per the requirements of the application layer.

• The protocol should be able to adapt to the dynamics of the network such as the rapid change in topology and changes in the nature of wireless links from uni-directional to bidirectional or vice versa.

• One of the important resources, the available bandwidth, must be used efficiently.

• The protocol should be aware of resource constraints such as battery power and buffer sizes and make efficient use of them.

• The transport layer protocol should make use of information from the lower layers in the protocol stack for improving the network throughput.

• It should have a well-defined cross-layer interaction framework for effective, scalable, and protocol-independent interaction with lower layers.

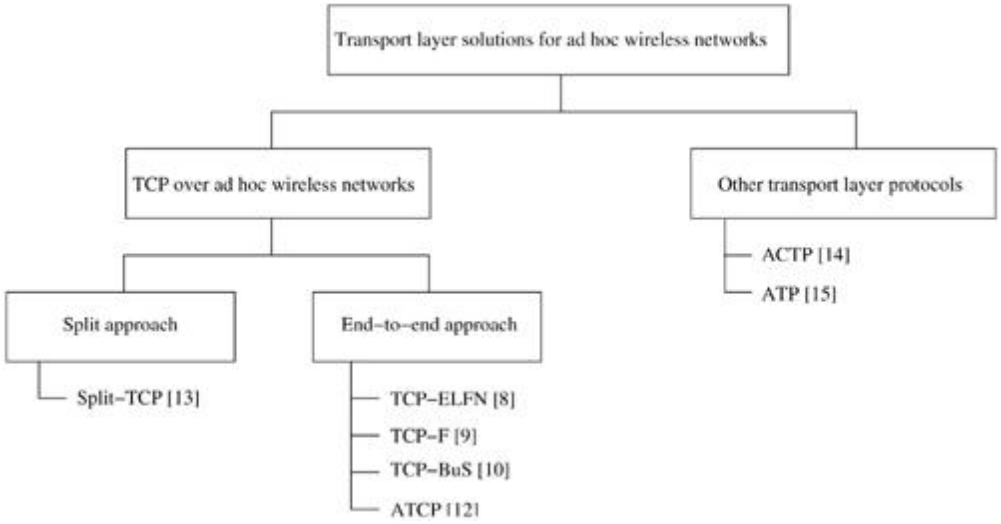
• The protocol should maintain end-to-end semantics.

3.13 CLASSIFICATION OF TRANSPORT LAYER SOLUTIONS

Figure 3.34 shows a classification tree for some of the transport layer protocols discussed in this chapter. The top-level classification divides the protocols as

extensions of TCP for ad hoc wireless networks and other transport layer protocols which are not based on TCP. The solutions for TCP over ad hoc wireless networks can further be classified into split approaches and end-to-end approaches.

Figure 3.34. Classification of transport layer solutions.



3.14 TCP OVER AD HOC WIRELESS NETWORKS The transmission control protocol (TCP) is the most predominant transport layer protocol in the Internet today. It transports more than 90% percent of the traffic on the Internet. Its reliability, end-to-end congestion control mechanism, byte stream transport mechanism, and, above all, its elegant and simple design have not only contributed to the success of the Internet, but also have made TCP an influencing protocol in the design of many of the other protocols and applications. Its adaptability to the congestion in the network has been an important feature leading to graceful degradation of the services offered by the network at times of extreme congestion. TCP in its traditional form was designed and optimized only for wired networks. Since TCP is widely used today and the efficient integration of an ad hoc wireless network with the Internet is paramount wherever possible, it is essential to have mechanisms that can improve TCP's performance in ad hoc wireless networks. This would enable the seamless operation of application-level protocols such as FTP, SMTP, and HTTP across the integrated ad hoc wireless networks and the Internet. This section discusses the issues and challenges that TCP experiences when used in

ad hoc wireless networks as well as some of the existing solutions for overcoming them.

3.14.1 A Brief Revisit to TraditionalTCP

TCP is a reliable, end-to-end, connection-oriented transport layer protocol that provides a byte-stream-based service [the stream of bytes from the application layer is split into TCP segments,[1] the length of each segment limited by a maximum segment size (MSS)]. The major responsibilities of TCP include congestion control, flow control, in-order delivery of packets, and reliable transportation of packets. Congestion control deals with excess traffic in the network which may lead to degradation in the performance of the network, whereas flow control controls the per-flow traffic such that the receiver capacity is not exceeded. TCP regulates the number of packets sent to the network by expanding and shrinking the congestion window. The TCP sender starts the session with a congestion window value of one MSS. It sends out one MSS and waits for the ACK. Once the ACK is received within the retransmission timeout (RTO) period, the congestion window is doubled and two MSSs are originated. This doubling of the congestion window with every successful acknowledgment of all the segments in the current congestion window, is called *slow-start* (a more appropriate name would be *exponential start*, as it actually grows exponentially) and it continues until the congestion window reaches the *slowstart threshold* (the slow-start threshold has an initial value of 64 KB).

Figure 3.34shows the variation of the congestion window in TCP; the slow start phase is between points A-B. Once it reaches the slow-start threshold (in Figure 3.34, the slow-start threshold is initially taken as 16 for illustration), it grows linearly, adding one MSS to the congestion window on every ACK received. This linear growth, which continues until the congestion window reaches the receiver window (which is advertised by the TCP receiver and carries the information about the receiver's buffer size), is called *congestion avoidance*, as it tries to avoid increasing the congestion window exponentially, which will surely worsen the congestion in the network. TCP updates the RTO period with the current round-trip delay calculated on the arrival of every ACK packet. If the ACK packet does not arrive within the RTO period, then it assumes that the

packet is lost. TCP assumes that the packet loss is due to the congestion in the network and it invokes the congestion control mechanism. The TCP sender does the following during congestion control:

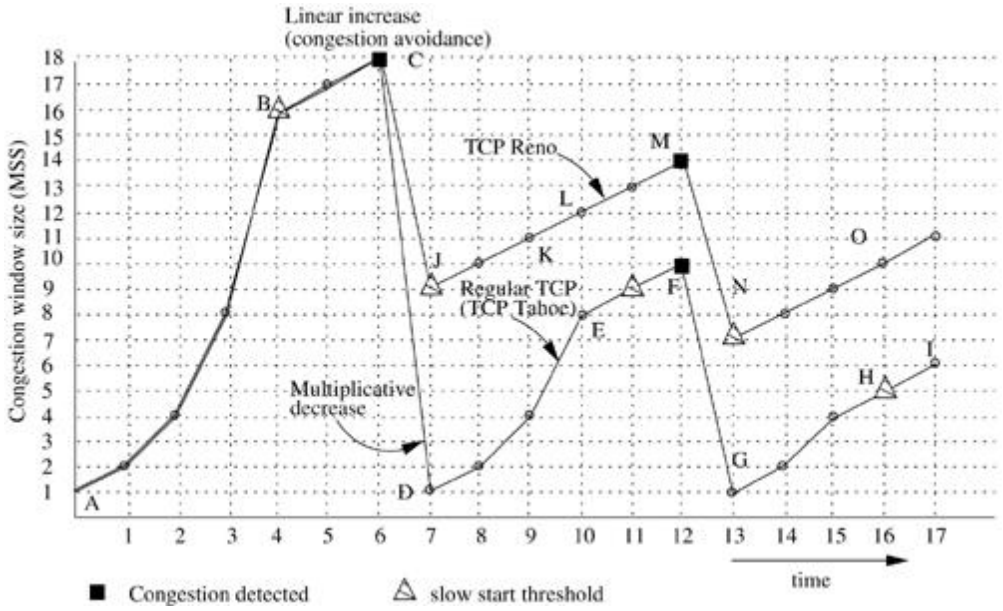
(i) reduces the slow-start threshold to half the current congestion window or two MSSs whichever is larger,

(ii) resets the congestion window size to one MSS,

(ii) activates the slow-start algorithm, and

(iii) resets the RTO with an exponential back-off value which doubles with every subsequent retransmission. The slow-start process further doubles the congestion window with every successfully acknowledged window and, upon reaching the slow-start threshold, it enters into the congestion avoidance phase.

Figure 3.34. Illustration of TCP congestion window.



The TCP sender also assumes a packet loss if it receives three consecutive duplicate ACKs (DUPACKs) [repeated acknowledgments for the same TCP segment that was successfully received in-order at the receiver]. Upon reception of three DUPACKs, the TCP sender retransmits the oldest unacknowledged segment. This is called the *fast retransmit* scheme. When the TCP receiver receives out-of-order packets, it generates DUPACKs to indicate to the TCP

sender about the sequence number of the last in-order segment received successfully. Among the several extensions of TCP, some of the important schemes are discussed below. The regularTCP which was discussed above is also called as TCP Tahoe (in most of the existing literature).TCP Reno is similar to TCP Tahoe with *fast recovery*. On timeout or arrival of three DUPACKs, the TCP Reno sender enters the fast recovery during which (refer to points C-JK in Figure 3.34) theTCP Reno sender retransmits the lost packet, reduces the slow-start threshold and congestion window size to half the size of the current congestion window, and increments the congestion window linearly (one MSS per DUPACK) with every subsequent DUPACK. On reception of a new ACK (not a DUPACK,*i.e.*, an ACK with a sequence number higher than the highest seen sequence number so far), the TCPReno resets the congestion window with the slow-start threshold and enters the congestion avoidance phase similar to TCP Tahoe (points K-L-M in Figure 3.34). J. C. Hoe proposed TCP-New Reno extending the TCP Reno in which the TCP sender does not exit the fast-recovery state, when a new ACK is received. Instead it continues to remain in the fast-recovery state until all the packets originated are acknowledged. For every intermediate ACK packet, TCP-New Reno assumes the next packet after the last acknowledged one is lost and is retransmitted. TCP with selective ACK (SACK) , improves the performance of TCP by using the selective ACKs provided by the receiver. The receiver sends a SACK instead of an ACK, which contains a set of SACK blocks. These SACK blocks contain information about the recently received packets which is used by the TCP sender while retransmitting the lost packets.

3.14.2 Why Does TCP Not Perform Well in Ad Hoc Wireless Networks? The major reasons behind throughput degradation that TCP faces when used in ad hoc wireless networks are the following:

• **Misinterpretation of packet loss:** Traditional TCP was designed for wired networks where the packet loss is mainly attributed to network congestion. Network congestion is detected by the sender's packet RTO period. Once a packet loss is detected, the sender node assumes congestion in the network and invokes a congestion control algorithm. Ad hoc wireless networks experience a

much higher packet loss due to factors such as high bit error rate (BER) in the wireless channel, increased collisions due to the presence of hidden terminals, presence of interference, location-dependent contention, uni-directional links, frequent path breaks due to mobility of nodes, and the inherent fading properties of the wireless channel.

• **Frequent path breaks:** Ad hoc wireless networks experience dynamic changes in network topology because of the unrestricted mobility of the nodes in the network. The topology changes lead to frequent changes in the connectivity of wireless links and hence the route to a particular destination may need to be recomputed very often. The responsibility of finding a route and reestablishing it once it gets broken is attached to the network layer Once a path is broken, the routing protocol initiates a route reestablishment process. This route reestablishment process takes a significant amount of time to obtain a new route to the destination. The route reestablishment time is a function of the number of nodes in the network, transmission ranges of nodes, current topology of the network, bandwidth of the channel, traffic load in the network, and the nature of the routing protocol. If the route reestablishment time is greater than the RTO period of the TCP sender, then the TCP sender assumes congestion in the network, retransmits the lost packets, and initiates the congestion control algorithm. These retransmissions can lead to wastage of bandwidth and battery power. Eventually, when a new route is found, the TCP throughput continues to be low for some time, as it has to build up the congestion window since the traditional TCP undergoes a slow start.

• **Effect of path length:** It is found that the TCP throughput degrades rapidly with an increase in path length in string (linear chain) topology ad hoc wireless networks, . This is shown in Figure 3.35. The possibility of a path break increases with path length. Given that the probability of a link break is p_l, the probability of a path break (p_b) for a path of length k can be obtained as $p_b = 1 - (1 - p_l)_k$. Figure 3.36 shows the variation of p_b with path length for $p_l = 0.1$. Hence as the path length increases, the probability of a path break increases, resulting in the degradation of the throughput in the network. **Figure 3.35. Variation of TCP throughput with path length.**

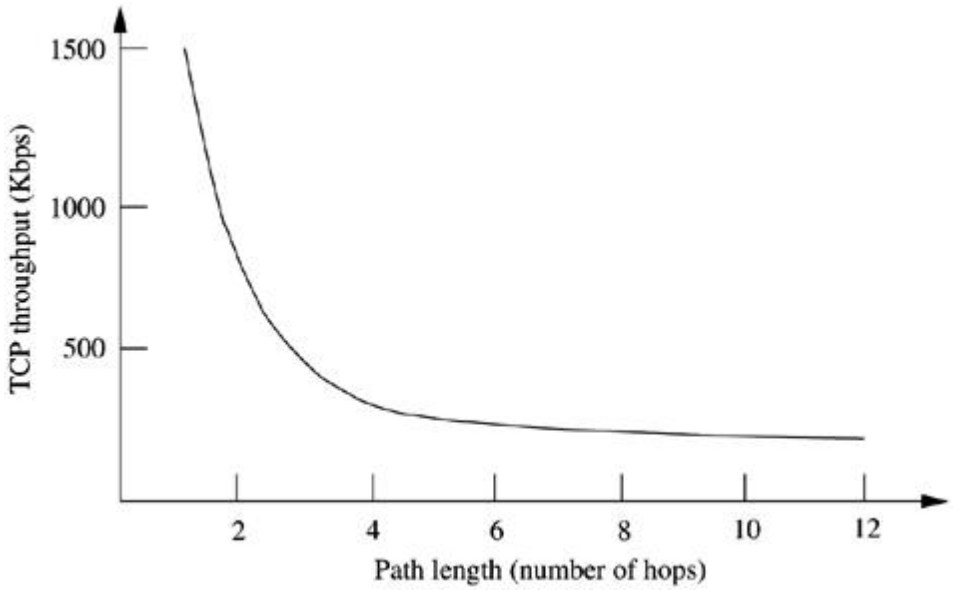
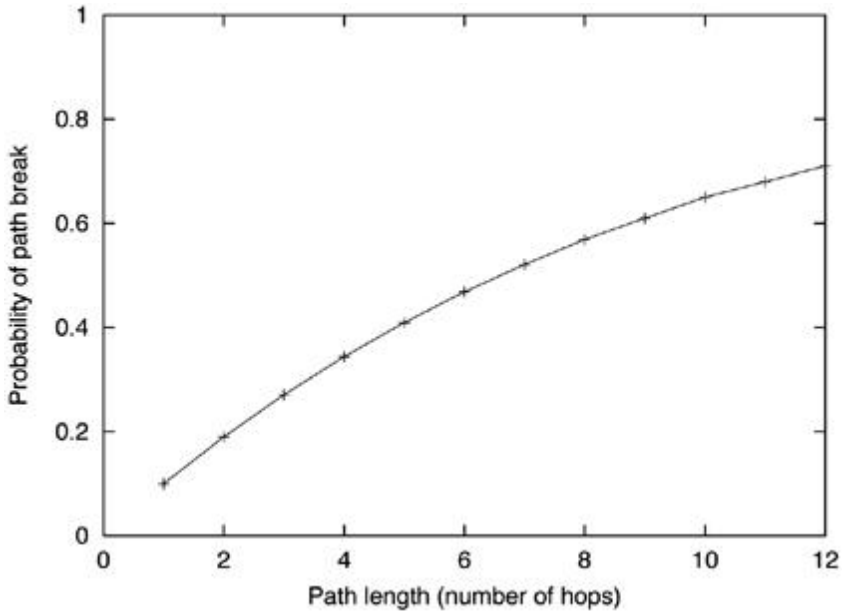


Figure 3.36. Variation of p_b with path length (p_l = 0.1). • Misinterpretation of congestion window:



TCP considers the congestion window as a measure of the rate of transmission that is acceptable to the network and the receiver. In ad hoc wireless networks, the congestion control mechanism is invoked when the network gets partitioned or when a path break occurs. This reduces the congestion window and increases the RTO period. When the route is reconfigured, the congestion window may not reflect the transmission rate acceptable to the new route, as the new route

may actually accept a much higher transmission rate. Hence, when there are frequent path breaks, the congestion window may not reflect the maximum transmission rate acceptable to the network and the receiver.

• **Asymmetric link behavior:** The radio channel used in ad hoc wireless networks has different properties such as location-dependent contention, environmental effects on propagation, and directional properties leading to asymmetric links. The directional links can result in delivery of a packet to a node, but failure in the delivery of the acknowledgment back to the sender. It is possible for a bidirectional link to become uni-directional for a while. This can also lead to TCP invoking the congestion control algorithm and several retransmissions.

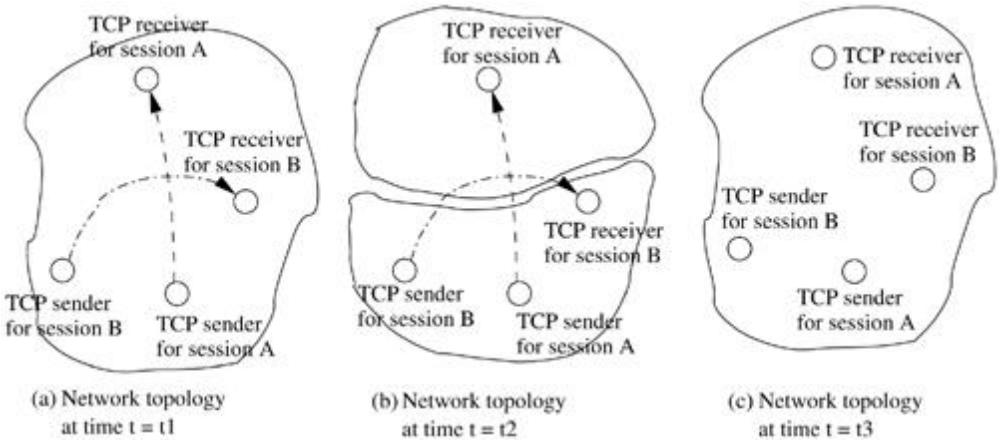
• **Uni-directional path:** Traditional TCP relies on end-to-end ACK for ensuring reliability. Since the ACK packet is very short compared to a data segment, ACKs consume much less bandwidth in wired networks. In ad hoc wireless networks, every TCP ACK packet requires RTS-CTS-Data-ACK exchange in case IEEE 802.11 is used as the underlying MAC protocol. This can lead to an additional overhead of more than 70 bytes if there are no retransmissions. This can lead to significant bandwidth consumption on the reverse path, which may or may not contend with the forward path. If the reverse path contends with the forward path, it can lead to the reduction in the throughput of the forward path. Some routing protocols select the forward path to be also used as the reverse path, whereas certain other routing protocols may use an entirely different or partially different path for the ACKs. A path break on an entirely different reverse path can affect the performance of the network as much as a path break in the forward path.

• **Multipath routing:** There exists a set of QoS routing and best-effort routing protocols that use multiple paths between a source-destination pair. There are several advantages in using multipath routing. Some of these advantages include the reduction in route computing time, the high resilience to path breaks, high call acceptance ratio, and better security. For TCP, these advantages may add to throughput degradation. These can lead to a significant amount of out-of-order packets, which in turn generates a set of duplicate acknowledgments

(DUPACKs) which cause additional power consumption and invocation of congestion control.

• **Network partitioning and remerging:** The randomly moving nodes in an ad hoc wireless network can lead to network partitions. As long as the TCP sender, the TCP receiver, and all the intermediate nodes in the path between the TCP sender and the TCP receiver remain in the same partition, the TCP connection will remain intact. It is likely that the sender and receiver of the TCPsession will remain in different partitions and, in certain cases, that only the intermediate nodes are affected by the network partitioning. Figure 3.37 illustrates the effect of network partitions in ad hoc wireless networks. A network with two TCP sessions A and B is shown in Figure 3.37 (a) at time instant t1. Due to dynamic topological changes, the network gets partitioned into two as in Figure 3.37 (b) at time t2. Now the TCP session A's sender and receiver belong to two different partitions and the TCPsession B experiences a path break. These partitions could merge back into a single network at time t3 (refer to Figure 3.37 (c)).

Figure 3.37. Effect of partitioning and merging of network.



• **The use of sliding-window-based transmission:** TCP uses a sliding window for flow control. The transmission of packets is decided by the size of the window, and when the ACKs arrive from a destination, further packets are transmitted. This avoids the use of individual fine-grained timers for transmission of each TCP flow. Such a design is preferred in order to improve scalability of the protocol in high-bandwidth networks such as the Internet where millions of TCP connections may be established with some heavily

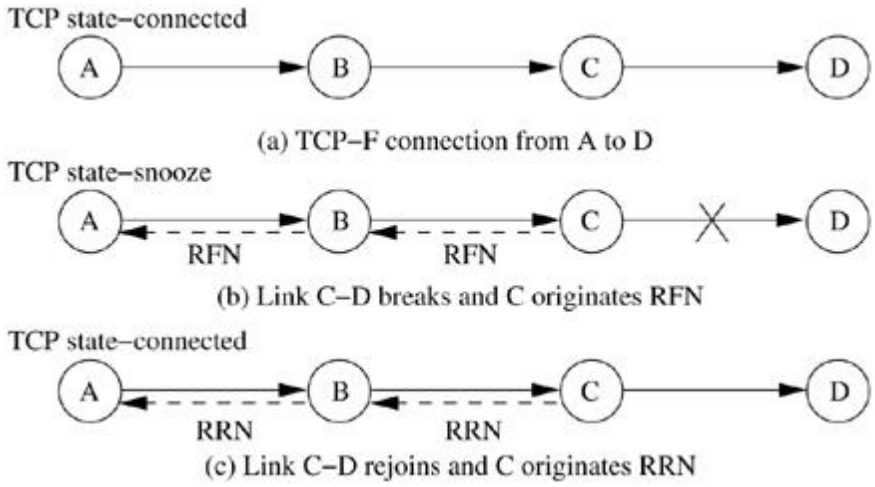
loaded servers. The use of a sliding window can also contribute to degraded performance in bandwidth-constrained ad hoc wireless networks where the MAC layer protocol may not exhibit short-term and long-term fairness. For example, the popular MAC protocols such as CSMA/CA protocol show shortterm unfairness, where a node that has captured the channel has a higher probability of capturing the channel again. This unfairness can lead to a number of TCPACK packets being delivered to the TCP sender in succession, leading to a burstiness in traffic due to the subsequent transmission of TCP segments. The enhancements to TCP that improve the performance of TCP in ad hoc wireless networks are discussed in the following sections.

3.14.3 Feedback-Based TCP

Feedback-based TCP [also referred to as TCP feedback (TCP-F)] proposes modifications to the traditional TCP for improving performance in ad hoc wireless networks. It uses a feedback-based approach. TCP-F requires the support of a reliable link layer and a routing protocol that can provide feedback to the TCP sender about the path breaks. The routing protocol is expected to repair the broken path within a reasonable time period. TCP-F aims to minimize the throughput degradation resulting from the frequent path breaks that occur in ad hoc wireless networks. During a TCP session, there could be several path breaks resulting in considerable packet loss and path reestablishment delay. Upon detection of packet loss, the sender in a TCP session invokes the congestion control algorithm leading to the exponential back-off of retransmission timers and a decrease in congestion window size. This was discussed earlier in this chapter. In TCP-F, an intermediate node, upon detection of a path break, originates a route failure notification (RFN) packet. This RFN packet is routed toward the sender of the TCP session. The TCP sender's information is expected to be obtained from the TCP packets being forwarded by the node. The intermediate node that originates the RFN packet is called the failure point (FP). The FP maintains information about all the RFNs it has originated so far. Every intermediate node that forwards the RFNpacket understands the route failure, updates its routing table accordingly, and avoids forwarding any more packets on that route. If any of the intermediate nodes that

receive RFN has an alternate route to the same destination, then it discards the RFN packet and uses the alternate path for forwarding further data packets, thus reducing the control overhead involved in the route reconfiguration process. Otherwise, it forwards the RFN toward the source node. When a TCP sender receives an RFN packet, it goes into a state called *snooze*. In the snooze state, a sender stops sending any more packets to the destination, cancels all the timers, freezes its congestion window, freezes the retransmission timer, and sets up a route failure timer. This route failure timer is dependent on the routing protocol, network size, and the network dynamics and is to be taken as the worst-case route reconfiguration time. When the route failure timer expires, the TCP sender changes from the snooze state to the *connected* state.Figure 3.38 shows the operation of the TCP-F protocol. In the figure, a TCP session is set up between node A and node D over the path A-B-C-D [refer to Figure 3.38 (a)]. When the intermediate link between node C and node D fails, node C originates an RFN packet and forwards it on the reverse path to the source node [see Figure 3.38 (b)]. The sender's TCP state is changed to the snooze state upon receipt of an RFN packet. If the link CD rejoins, or if any of the intermediate nodes obtains a path to destination node D, a route reestablishment notification (RRN) packet is sent to node A and the TCPstate is updated back to the connected state [Figure 3.38 (c)].

Figure 3.38. Operation of TCP-F.



As soon as a node receives an RRN packet, it transmits all the packets in its buffer, assuming that the network is back to its original state. This can also take care of all the packets that were not acknowledged or lost during transit due to the path break. In fact, such a step avoids going through the slow-start process that would otherwise have occurred immediately after a period of congestion. The route failure timer set after receiving the RFN packet ensures that the sender does not remain in the snooze state indefinitely. Once the route failure timer expires, the sender goes back to the connected state in which it reactivates the frozen timers and starts sending the buffered and unacknowledged packets. This can also take care of the loss of the RRN packet due to any possible subsequent congestion. TCP-F permits the TCP congestion control algorithm to be in effect when the sender is not in the snooze state, thus making it sensitive to congestion in the network.

Advantages and Disadvantages

TCP-F provides a simple feedback-based solution to minimize the problems arising out of frequent path breaks in ad hoc wireless networks. At the same time, it also permits the TCP congestion control mechanism to respond to congestion in the network. TCP-F depends on the intermediate nodes' ability to detect route failures and the routing protocols' capability to reestablish a broken path within a reasonably short duration. Also, the FP should be able to obtain the correct path (the path which the packet traversed) to the TCP-F sender for sending the RFN packet. This is simple with a routing protocol that uses source routing [*i.e.*, dynamic source routing (DSR)]. If a route to the sender is not available at the FP, then additional control packets may need to be generated for routing the RFNpacket. TCP-F has an additional state compared to the traditional TCP state machine, and hence its implementation requires modifications to the existing TCP libraries. Another disadvantage of TCP-F is that the congestion window used after a new route is obtained may not reflect the achievable transmission rate acceptable to the network and the TCP-F receiver.

3.14.4 TCP with Explicit Link Failure Notification

Holland and Vaidya proposed the use of TCP with explicit link failure notification (TCP-ELFN) for improving TCP performance in ad hoc wireless networks. This is similar to TCP-F, except for the handling of explicit link failure notification (ELFN) and the use of TCP probe packets for detecting the route reestablishment. The ELFN is originated by the node detecting a path break upon detection of a link failure to the TCP sender. This can be implemented in two ways:

(i) by sending an ICMP[2] destination unreachable (DUR) message to the sender, or (ii) by piggy-backing this information on the *RouteError*[3] message that is sent to the sender. [2] Internet control message protocol (IETF RFC 792) is used for defining control messages for aiding routing in the Internet. [3] Certain routing protocols for ad hoc wireless networks have explicit *RouteError* messages to inform the sender about path breaks so that the sender can recompute a fresh route to the destination. This is especially used in on-demand routing protocols such as DSR. Once the TCP sender receives the ELFN packet, it disables its retransmission timers and enters a *standby* state. In this state, it periodically originates probe packets to see if a new route is reestablished. Upon reception of an ACK by the TCP receiver for the probe packets, it leaves the standby state, restores the retransmission timers, and continues to function as normal. Advantages and Disadvantages TCP-ELFN improves the TCP performance by decoupling the path break information from the congestion information by the use of ELFN. It is less dependent on the routing protocol and requires only link failure notification about the path break. The disadvantages of TCP-ELFN include the following: (i) when the network is temporarily partitioned, the path failure may last longer and this can lead to the origination of periodic probe packets consuming bandwidth and power and (ii) the congestion window used after a new route is obtained may not reflect the achievable transmission rate acceptable to the network and the TCP receiver.

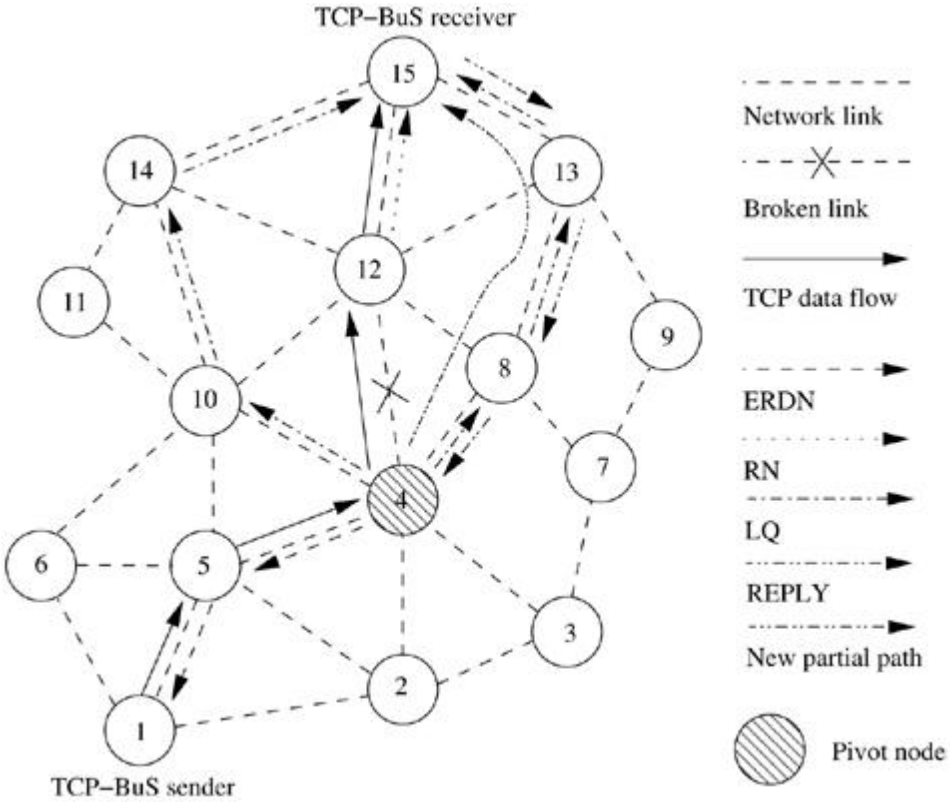
3.14.5 TCP-BuS

TCP with buffering capability and sequence information (TCP-BuS) is similar to the TCP-F andTCP-ELFN in its use of feedback information from an intermediate node on detection of a path break. But TCP-BuS is more dependent on the routing protocol compared to TCP-F and TCP-ELFN. TCP-BuS was

proposed, with associativity-based routing (ABR) protocol as the routing scheme. Hence, it makes use of some of the special messages such as localized query (LQ) and REPLY, defined as part ofABR for finding a partial path. These messages are modified to carry TCP connection and segment information. Upon detection of a path break, an upstream intermediate node [called pivot node (PN)] originates an explicit route disconnection notification (ERDN) message. This ERDN packet is propagated to the TCP-BuS sender and, upon reception of it, the TCP-BuS sender stops transmission and freezes all timers and windows as in TCP-F. The packets in transit at the intermediate nodes from the TCP-BuS sender to the PN are buffered until a new partial path from the PN to the TCPBuS receiver is obtained by the PN. In order to avoid unnecessary retransmissions, the timers for the buffered packets at the TCP-BuS sender and at the intermediate nodes up to PN use timeout values proportional to the roundtrip time (RTT). The intermediate nodes between the TCP-BuS sender and the PN can request the TCP-BuS sender to selectively retransmit any of the lost packets. Upon detection of a path break, the downstream node originates a route notification (RN) packet to the TCP-BuS receiver, which is forwarded by all the downstream nodes in the path. An intermediate node that receives an RN packet discards all packets belonging to that flow. The ERDN packet is propagated to the TCP-BuS sender in a reliable way by using an implicit acknowledgment and retransmission mechanism. The PN includes the sequence number of the TCP segment belonging to the flow that is currently at the head of its queue in the ERDN packet. The PN also attempts to find a new partial route to the TCP-BuS receiver, and the availability of such a partial path to destination is intimated to theTCP-BuS sender through an explicit route successful notification (ERSN) packet. TCP-BuS utilizes the route reconfiguration mechanism of ABR to obtain the partial route to the destination. Due to this, other routing protocols may require changes to support TCP-BuS. The LQ and REPLY messages are modified to carry TCP segment information, including the last successfully received segment at the destination. The LQ packet carries the sequence number of the segment at the head of the queue buffered at the PN and the REPLY carries the sequence number of the last successful segment the TCPBuS receiver received. This enables the TCP-BuS receiver to understand the packets lost in

transition and those buffered at the intermediate nodes. This is used to avoid fast retransmission requests usually generated by the TCP-BuS receiver when it notices an out-of-order packet delivery. Upon a successfulLQREPLY process to obtain a new route to the TCP-BuS receiver, PN informs the TCP-BuS sender of the new partial path using the ERSN packet. When the TCP-BuS sender receives an ERSN packet, it resumes the data transmission. Since there is a chance for ERSN packet loss due to congestion in the network, it needs to be sent reliably. The TCP-BuS sender also periodically originates probe packets to check the availability of a path to the destination. Figure 3.39 shows an illustration of the propagation of ERDN and RN messages when a link between nodes 4 and 12 fails.

Figure 3.39. Operation of TCP-BuS.



When a TCP-BuS sender receives the ERSN message, it understands, from the sequence number of the last successfully received packet at the destination and the sequence number of the packet at the head of the queue at PN, the packets lost in transition. The TCP-BuS receiver understands that the lost packets will

be delayed further and hence uses a selective acknowledgment strategy instead of fast retransmission. These lost packets are retransmitted by the TCP-BuS sender. During the retransmission of these lost packets, the network congestion between the TCP-BuS sender and PN is handled in a way similar to that in traditional TCP.

Advantages and Disadvantages

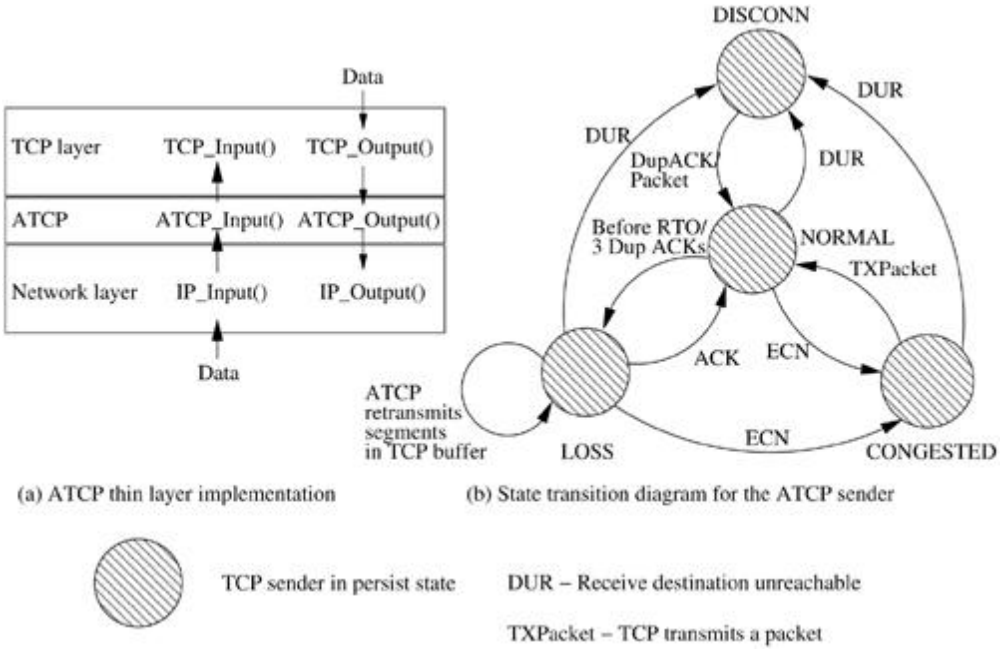
The advantages of TCP-BuS include performance improvement and avoidance of fast retransmission due to the use of buffering, sequence numbering, and selective acknowledgment. TCP-BuS also takes advantage of the underlying routing protocols, especially the on-demand routing protocols such asABR. The disadvantages of TCP-BuS include the increased dependency on the routing protocol and the buffering at the intermediate nodes. The failure of intermediate nodes that buffer the packets may lead to loss of packets and performance degradation. The dependency of TCP-BuS on the routing protocol may degrade its performance with other routing protocols that do not have similar control messages as in ABR.

3.14.6 Ad Hoc TCP

Similar to TCP-F and TCP-ELFN, ad hoc TCP (ATCP) also uses a network layer feedback mechanism to make the TCP sender aware of the status of the network path over which the TCPpackets are propagated. Based on the feedback information received from the intermediate nodes, theTCP sender changes its state to the *persist* state, *congestion control* state, or the *retransmit* state. When an intermediate node finds that the network is partitioned, then the TCP sender state is changed to the persist state where it avoids unnecessary retransmissions. When ATCP puts TCP in the persist state, it sets TCP's congestion window size to one in order to ensure that TCP does not continue using the old congestion window value. This forces TCP to probe the correct value of the congestion window to be used for the new route. If an intermediate node loses a packet due to error, then the ATCP at theTCP sender immediately retransmits it without invoking the congestion control algorithm. In order to be compatible with widely deployed TCP-based networks, ATCP provides this feature without modifying the traditional TCP. ATCP is implemented as a thin layer residing

between the IP and TCPprotocols. The ATCP layer essentially makes use of the explicit congestion notification (ECN) for maintenance of the states. Figure 3.40 (a) shows the thin layer implementation of ATCP between the traditional TCP layer and theIP layer. This does not require changes in the existing TCP protocol. This layer is active only at the TCPsender. The major function of the ATCP layer is to monitor the packets sent and received by the TCPsender, the state of the TCP sender, and the state of the network. Figure 3.40 (b) shows the state transition diagram for the ATCP at the TCP sender. The four states in the ATCP are (i) NORMAL, (ii) CONGESTED, (iii) LOSS, and (iv) DISCONN. When a TCP connection is established, the ATCP sender state is in NORMAL. In this state, ATCP does not interfere with the operation of TCP and it remains invisible.

Figure 3.40. An illustration of ATCP thin layer and ATCP state diagram.



When packets are lost or arrive out-of-order at the destination, it generates duplicate ACKs. In traditional TCP, upon reception of duplicate ACKs, the TCP sender retransmits the segment under consideration and shrinks the contention window. But the ATCP sender counts the number of duplicate ACKs received and if it reaches three, instead of forwarding the duplicate ACKs to TCP, it puts TCP in the persist state and ATCP in the LOSS state. Hence, the TCP sender

avoids invoking congestion control. In the LOSS state, ATCP retransmits the unacknowledged segments from the TCPbuffer. When a new ACK comes from the TCP receiver, it is forwarded to TCP and the TCP sender is removed from the persist state and then the ATCP sender changes to the NORMAL state. When the ATCP sender is in the LOSS state, the receipt of an ECN message or an ICMP *source quench* message changes it to the CONGESTED state. Along with this state transition, the ATCP sender removes the TCP sender from the persist state. When the network gets congested, the ECN[4] flag is set in the data and the ACK packets. When the ATCP sender receives this ECN message in the normal state, it changes to the CONGESTED state and just remains invisible, permitting TCP to invoke normal congestion control mechanisms. When a route failure or a transient network partition occurs in the network, ATCP expects the network layer to detect these and inform the ATCP sender through an ICMP destination unreachable (DUR) message. Upon reception of the DUR message, ATCP puts the TCP sender into the persist state and enters into the DISCONN state. It remains in the DISCONN state until it is connected and receives any data or duplicate ACKs. On the occurrence of any of these events, ATCP changes to the NORMAL state. The connected status of the path can be detected by the acknowledgments for the periodic probe packets generated by the TCP sender. The receipt of an ICMPDUR message in the LOSS state or the CONGESTED state causes a transition to the DISCONN state. When ATCP puts TCP into the persist state, it sets the congestion window to one segment in order to make TCP probe for the new congestion window when the new route is available. In summary, ATCP tries to perform the activities listed in Table 3.10.

[4] ECN is currently under consideration by IETF and is now a standard (IETF RFC 3168).

Table 3.10. The actions taken by ATCP

Event	Action
Packet loss due to high BER	Retransmits the lost packets without reducing congestion window
Route recomputation delay	Makes the TCP sender go to persist state and stop transmission until new route has been found
Transient partitions	Makes the TCP sender go to persist state and stop transmission until new route has been found
Out-of-order packet delivery due to multipath routing	Maintains TCP sender unaware of this and retransmits the packets from TCP buffer
Change in route	Recomputes the congestion window

Advantages and Disadvantages

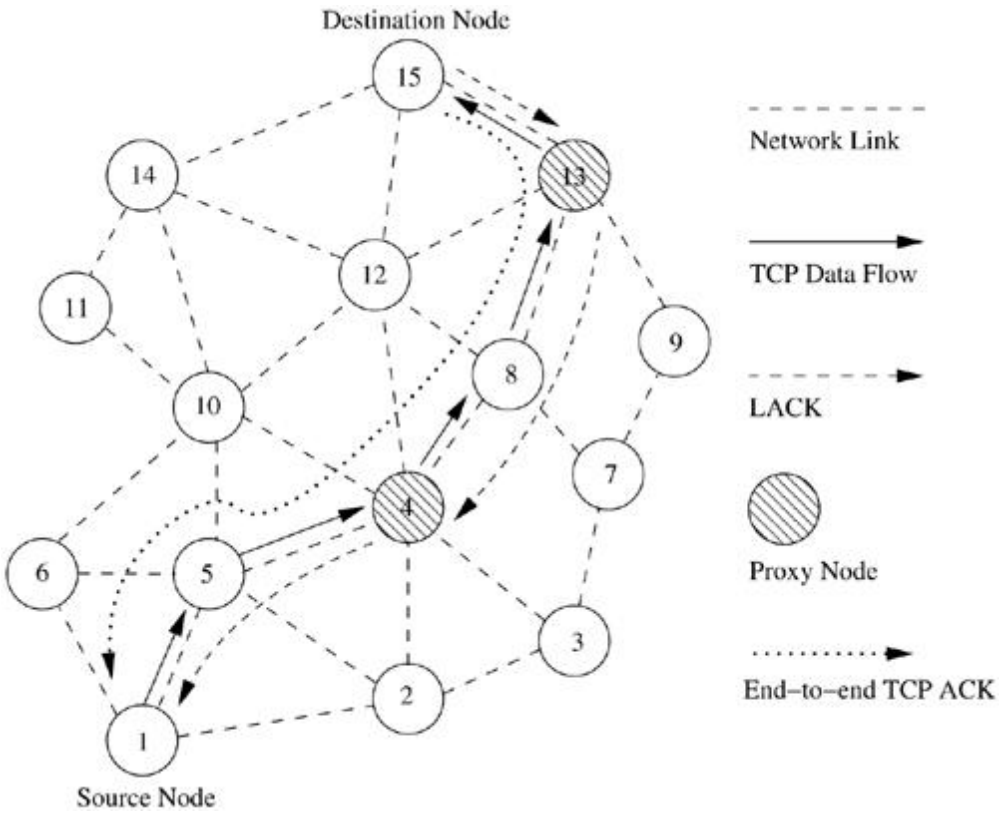
Two major advantages of ATCP are (i) it maintains the end-to-end semantics of TCP and (ii) it is compatible with traditional TCP. These advantages permit ATCP to work seamlessly with the Internet. In addition, ATCP provides a feasible and efficient solution to improve throughput of TCP in ad hoc wireless networks. The disadvantages of ATCP include (i) the dependency on the network layer protocol to detect the route changes and partitions, which not all routing protocols may implement and (ii) the addition of a thin ATCP layer to the TCP/IP protocol stack that requires changes in the interface functions currently being used.

3.14.7 SplitTCP

One of the major issues that affects the performance of TCP over ad hoc wireless networks is the degradation of throughput with increasing path length, as discussed early in this chapter. The short (*i.e.,* in terms of path length) connections generally obtain much higher throughput than long connections. This can also lead to unfairness among TCP sessions, where one session may obtain much higher throughput than other sessions. This unfairness problem is further worsened by the use of MAC protocols such as IEEE 802.11, which are found to give a higher throughput for certain link-level sessions, leading to an effect known as *channel capture* effect. This effect leads to certain flows capturing the channel for longer time durations, thereby reducing throughput for other flows. The channel capture effect can also lead to low overall system throughput. The reader can refer to Chapter 6for more details on MAC protocols and throughput fairness. Split-TCP provides a unique solution to this problem

by splitting the transport layer objectives into congestion control and end-to-end reliability. The congestion control is mostly a local phenomenon due to the result of high contention and high traffic load in a local region. In the ad hoc wireless network environment, this demands local solutions. At the same time, reliability is an end-to-end requirement and needs end-to-end acknowledgments. In addition to splitting the congestion control and reliability objectives, split-TCP splits a long TCP connection into a set of short concatenated TCP connections (called *segments* or *zones*) with a number of selected intermediate nodes (known as *proxy* nodes) as terminating points of these short connections. Figure 3.41 illustrates the operation of split-TCP where a three segment split-TCP connection exists between source node 1 and destination node 15. A proxy node receives the TCP packets, reads its contents, stores it in its local buffer, and sends an acknowledgment to the source (or the previous proxy). This acknowledgment called local acknowledgment (LACK) does not guarantee end-to-end delivery. The responsibility of further delivery of packets is assigned to the proxy node. A proxy node clears a buffered packet once it receives LACK from the immediate successor proxy node for that packet. Split-TCP maintains the end-to-end acknowledgment mechanism intact, irrespective of the addition of zone-wise LACKs. The source node clears the buffered packets only after receiving the end-to-end acknowledgment for those packets.

Figure 3.41. An illustration of Split-TCP.



In Figure 3.41, node 1 initiates a TCP session to node 15. Node 4 and node 13 are chosen as proxy nodes. The number of proxy nodes in a TCP session is determined by the length of the path between source and destination nodes. Based on a distributed algorithm, the intermediate nodes that receive TCP packets determine whether to act as a proxy node or just as a simple forwarding node. The most simple algorithm makes the decision for acting as proxy node if the packet has already traversed more than a predetermined number of hops from the last proxy node or the sender of the TCP session. In Figure 3.41, the path between node 1 and node 4 is the first zone (segment), the path between nodes 4 and 13 is the second zone (segment), and the last zone is between node 13 and 15. The proxy node 4, upon receipt of each TCP packet from source node 1, acknowledges it with a LACKpacket, and buffers the received packets. This buffered packet is forwarded to the next proxy node (in this case, node 13) at a transmission rate proportional to the arrival of LACKs from the next proxy node or destination. The transmission control window at the TCP sender is also split into two windows, that is, the congestion window and the end-to-end window. The congestion window changes according to the rate of arrival of

LACKs from the next proxy node and the end-to-end window is updated based on the arrival of end-to-end ACKs. Both these windows are updated as per traditional TCP except that the congestion window should stay within the endto-end window. In addition to these transmission windows at the TCP sender, every proxy node maintains a congestion window that governs the segment level transmission rate.

Advantages and Disadvantages

Split-TCP has the following advantages: (i) improved throughput, (ii) improved throughput fairness, and (iii) lessened impact of mobility. Throughput improvement is due to the reduction in the effective transmission path length (number of hops in a zone or a path segment). TCP throughput degrades with increasing path length. Split-TCP has shorter concatenated path segments, each operating at its own transmission rate, and hence the throughput is increased. This also leads to improved throughput fairness in the system. Since in split-TCP, the path segment length can be shorter than the end-to-end path length, the effect of mobility on throughput is lessened. The disadvantages of split-TCP can be listed as follows: (i) It requires modifications to TCP protocol, (ii) the end-to-end connection handling of traditional TCP is violated, and (iii) the failure of proxy nodes can lead to throughput degradation. The traditional TCP has end-to-end semantics, where the intermediate nodes do not process TCP packets, whereas in split-TCP, the intermediate nodes need to process the TCP packets and hence, in addition to the loss of end-to-end semantics, certain security schemes that require IP payload encryption cannot be used. During frequent path breaks or during frequent node failures, the performance of split-TCP may be affected.

3.14.8 A Comparison of TCP Solutions for Ad Hoc Wireless Networks

Table 3.11 compares how various issues are handled in the TCP extensions discussed so far in this chapter.

Table 3.11. A comparison of TCP solutions for ad hoc wireless networks

Issue	TCP-F	TCP-ELFN	TCP-BuS	ATCP	Split-TCP
Packet loss due to BER or collision	Same as TCP	Same as TCP	Same as TCP	Retransmits the lost packets without invoking congestion control	Same as TCP
Path breaks	RFN is sent to the TCP sender and state changes to snooze	ELFN is sent to the TCP sender and state changes to standby	ERDN is sent to the TCP sender, state changes to snooze, ICMP DUR is sent to the TCP sender, and ATCP puts TCP into persist state	Same as TCP	Same as TCP
Out-of-order packets	Same as TCP	Same as TCP	Out-of-order packets reached after a path recovery are handled	ATCP reorders packets and hence TCP avoids sending duplicates	Same as TCP
Congestion	Same as TCP	Same as TCP	Explicit messages such as ICMP source quench are used	ECN is used to notify TCP sender. Congestion control is same as TCP	Since connection is split, the congestion control is handled within a zone by proxy nodes
Congestion window after path reestablishment	Same as before the path break	Same as before the path break	Same as before the path break	Recomputed for new route	Proxy nodes maintain congestion window and handle congestion
Explicit path break notification	Yes	Yes	Yes	Yes	No
Explicit path reestablishment notification	Yes	No	Yes	No	No
Dependency on routing protocol	Yes	Yes	Yes	Yes	No
End-to-end semantics	Yes	Yes	Yes	Yes	No
Packets buffered at intermediate nodes	No	No	Yes	No	Yes

3.15 OTHER TRANSPORT LAYER PROTOCOLS FOR AD HOC WIRELESS NETWORKS

The performance of a transport layer protocol can be enhanced if it takes into account the nature of the network environment in which it is applied. Especially in wireless environments, it is important to consider the properties of the physical layer and the interaction of the transport layer with the lower layers. This section discusses some of the transport layer protocols that were designed specifically for ad hoc wireless networks. Even though interworking with TCP is very important, there exist several application scenarios such as military communication where a radically new transport layer protocol can be used.

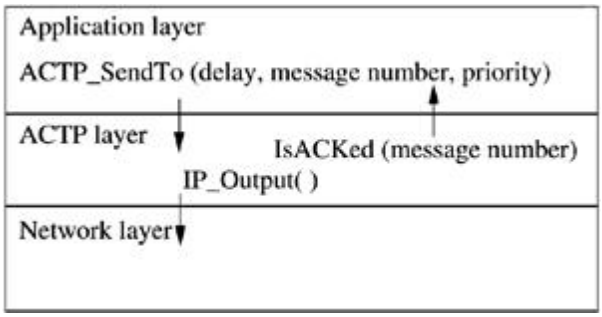
3.15.1 Application Controlled Transport Protocol

Unlike the TCP solutions discussed earlier in this chapter, application controlled transport protocol (ACTP[5]) is a light-weight transport layer protocol. It is not an extension to TCP. ACTP assigns the responsibility of ensuring reliability to the application layer. It is more like UDP with feedback of delivery and state maintenance. ACTP stands in between TCP and UDP where TCP experiences low performance with high reliability and UDP provides better performance with high packet loss in ad hoc wireless networks. [5] Originally called ATP, for differentiating with ad hoc transport protocol it is referred to as ACTP in this chapter. The key

design philosophy of ACTP is to leave the provisioning of reliability to the application layer and provide a simple feedback information about the delivery status of packets to the application layer. ACTP supports the priority of packets to be delivered, but it is the responsibility of the lower layers to actually provide a differentiated service based on this priority. Figure 3.42 shows the ACTP layer and the API functions used by the application layer to interact with the ACTP layer. Each API function call to send a packet [*SendTo()*] contains the additional information required for ACTP such as the maximum delay the packet can tolerate (delay), the message number of the packet, and the priority of the packet. The message number is assigned by the application layer, and it need not to be in sequence. The priority level is assigned for every packet by the application. It can be varied across packets in the same flow with increasing numbers referring to higher priority packets. The non-zero value in the message number field implicitly conveys that the application layer expects a delivery status information about the packet to be sent. This delivery status is maintained at the ACTP layer, and is available to the application layer for verification through another API function*IsACKed<message number>*. The delivery status returned by *IsACKed<message number>* function call can reflect (i) a successful delivery of the packet (ACK received), (b) a possible loss of the packet (no ACK received and the deadline has expired), (iii) remaining time for the packet (no ACK received but the deadline has not expired), and (iv) no state information exists at the ACTP layer regarding the message under consideration. A zero in the delay field refers to the highest priority packet, which requires immediate transmission with minimum possible delay. Any other value in the delay field refers to the delay that the message can experience. On getting the information about the delivery status, the application layer can decide on retransmission of a packet with the same old priority or with an updated priority. Well after the packet's lifetime expires, ACTP clears the packet's state information and delivery status. The packet's lifetime is calculated as 4 × retransmit timeout (RTO) and is set as the lifetime when the packet is sent to the network layer. A node estimates the RTO interval by using the round-trip time between the transmission time of a message and the time of reception of the corresponding ACK. Hence, the RTO value may not be

available if there are no existing reliable connections to a destination. A packet without any message number (*i.e.*, no delivery status required) is handled exactly the same way as in UDP without maintaining any state information.

Figure 3.42. An illustration of the interface functions used in ACTP.



Advantages and Disadvantages

One of the most important advantages of ACTP is that it provides the freedom of choosing the required reliability level to the application layer. Since ACTP is a light-weight transport layer protocol, it is scalable for large networks. Throughput is not affected by path breaks as much as in TCP as there is no congestion window for manipulation as part of the path break recovery. One disadvantage of ACTP is that it is not compatible with TCP. Use of ACTP in a very large ad hoc wireless network can lead to heavy congestion in the network as it does not have any congestion control mechanism.

3.15.2 Ad Hoc Transport Protocol

Ad hoc transport protocol (ATP) is specifically designed for ad hoc wireless networks and is not a variant of TCP. The major aspects by which ATP defers from TCP are (i) coordination among multiple layers, (ii) rate based transmissions, (iii) decoupling congestion control and reliability, and (iv) assisted congestion control. Similar to other TCP variants proposed for ad hoc wireless networks, ATP uses services from network and MAC layers for improving its performance. ATP uses information from lower layers for (i) estimation of the initial transmission rate, (ii) detection, avoidance, and control of congestion, and (iii) detection of path breaks. Unlike TCP, ATP utilizes a timer-based transmission, where the transmission rate is decided by the

granularity of the timer which is dependent on the congestion in the network. The congestion control mechanism is decoupled from the reliability and flow control mechanisms. The network congestion information is obtained from the intermediate nodes, whereas the flow control and reliability information are obtained from the ATP receiver. The intermediate nodes attach the congestion information to every ATP packet and the ATP receiver collates it before including it in the next ACK packet. The congestion information is expressed in terms of the weighted averaged[6] queuing delay (D_Q) and contention delay (D_C) experienced by the packets at every intermediate node. The field in which this delay information is included is referred to as the *rate feedback field* and the transmission rate is the inverse of the delay information contained in the rate feedback field. Intermediate nodes attach the current delay information to every ATP data packet if the already existing value is smaller than the current delay. The ATP receiver collects this delay information and the weighted average value is attached in the periodic ACK (ATP uses SACK mechanism, hence ACK refers to SACK) packet sent back to the ATP sender. During a connection startup process or when ATP recovers from a path break, the transmission rate to be used is determined by a process called *quick start.* During the quick start process, the ATP sender propagates a probe packet to which the intermediate nodes attach the transmission rate (in the form of current delay), which is received by the ATP receiver, and an ACK is sent back to the ATP sender. The ATP sender starts using the newly obtained transmission rate by setting the data transmission timers. During a connection startup, the connection request and the ACK packets are used as probe packets in order to reduce control overhead. When there is no traffic around an intermediate node, the transmission delay is approximated as $\beta \times (D_Q + D_C)$, where β is the factor that considers the induced traffic load. This is to consider the induced load (load on a particular link due to potential contention introduced by the upstream and downstream nodes in the path) when the actual transmission begins. A default value of 3 is used for β. ATP uses SACK packets periodically to ensure the selective retransmission of lost packets, which ensures the reliability of packet delivery. The SACK period is chosen such that it is more than the round-trip time and can track the network dynamics. The receiver performs a weighted average of the

delay/transmission rate information for every incoming packet to obtain the transmission rate for an ATP flow and this value is included in the subsequent SACK packet it sends. In addition to the rate feedback, the ATP receiver includes flow control information in the SACK packets. [6] Originally called "exponentially averaged," renamed here with a more appropriate term, "weighted average." An example for this is , where *a* is an appropriate weight factor and the other terms are self-explanatory. Unlike TCP, which employs either a decrease of the congestion window or an increase of the congestion window after a congestion, ATP has three phases, namely, increase, decrease, and maintain. If the new transmission rate (*R*) fed back from the network is beyond a threshold (γ) greater than the current transmission rate (*S*) [*i.e.*, $R > S(1+\gamma)$], then the current transmission rate is increased by a fraction (*k*) of the difference between the two transmission rates ( ). The fraction and threshold are taken to avoid rapid fluctuations in the transmission rate and induced load. The current transmission rate is updated to the new transmission rate if the new transmission rate is lower than the current transmission rate. In the maintain phase, if the new transmission rate is higher than the current transmission rate, but less than the above mentioned threshold, then the current transmission rate is maintained without any change. If an ATP sender has not received any ACK packets for two consecutive feedback periods, it undergoes a multiplicative decrease of the transmission rate. After a third such period without any ACK, the connection is assumed to be lost and the ATP sender goes to the connection initiation phase during which it periodically generates probe packets. When a path break occurs, the network layer detects it and originates an ELFN packet toward the ATP sender. The ATP sender freezes the sender state and goes to the connection initiation phase. In this phase also, the ATP sender periodically originates probe packets to know the status of the path. With a successful probe, the sender begins data transmission again. Advantages and Disadvantages The major advantages of ATP include improved performance, decoupling of the congestion control and reliability mechanisms, and avoidance of congestion window fluctuations. ATP does not maintain any per flow state at the intermediate nodes. The congestion information is gathered directly from the nodes that experience it. The major disadvantage of ATP is the lack of interoperability with TCP. As TCP is a widely used transport layer protocol, interoperability with TCP servers and clients in the Internet is

important in many applications. For large ad hoc wireless networks, the fine-grained per-flow timer used at the ATP sender may become a scalability bottleneck in resource-constrained mobile nodes.

3.16 SECURITY IN AD HOC WIRELESS NETWORKS

As mentioned earlier, due to the unique characteristics of ad hoc wireless networks, such networks are highly vulnerable to security attacks compared to wired networks or infrastructure-based wireless networks. The following sections discuss the various security requirements in ad hoc wireless networks, the different types of attacks possible in such networks, and some of the solutions proposed for ensuring network security.

3.17 NETWORK SECURITY REQUIREMENTS A security protocol for ad hoc wireless networks should satisfy the following requirements. The requirements listed below should in fact be met by security protocols for other types of networks also.

• **Confidentiality:** The data sent by the sender (source node) must be comprehensible only to the intended receiver (destination node). Though an intruder might get hold of the data being sent, he/she must not be able to derive any useful information out of the data. One of the popular techniques used for ensuring confidentiality is data encryption.

• **Integrity:** The data sent by the source node should reach the destination node as it was sent: unaltered. In other words, it should not be possible for any malicious node in the network to tamper with the data during transmission.

• **Availability:** The network should remain operational all the time. It must be robust enough to tolerate link failures and also be capable of surviving various attacks mounted on it. It should be able to provide the guaranteed services whenever an authorized user requires them.

• **Non-repudiation:** Non-repudiation is a mechanism to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message. Digital signatures, which

function as unique identifiers for each user, much like a written signature, are used commonly for this purpose.

3.18 ISSUES AND CHALLENGES IN SECURITY PROVISIONING

Designing a foolproof security protocol for ad hoc wireless is a very challenging task. This is mainly because of certain unique characteristics of ad hoc wireless networks, namely, shared broadcast radio channel, insecure operating environment, lack of central authority, lack of association among nodes, limited availability of resources, and physical vulnerability. A detailed discussion on how each of the above mentioned characteristics causes difficulty in providing security in ad hoc wireless networks is given below.

• **Shared broadcast radio channel:** Unlike in wired networks where a separate dedicated transmission line can be provided between a pair of end users, the radio channel used for communication in ad hoc wireless networks is broadcast in nature and is shared by all nodes in the network. Data transmitted by a node is received by all nodes within its direct transmission range. So a malicious node could easily obtain data being transmitted in the network. This problem can be minimized to a certain extent by using directional antennas.

• **Insecure operational environment:** The operating environments where ad hoc wireless networks are used may not always be secure. One important application of such networks is in battlefields. In such applications, nodes may move in and out of hostile and insecure enemy territory, where they would be highly vulnerable to security attacks.

• **Lack of central authority:** In wired networks and infrastructure-based wireless networks, it would be possible to monitor the traffic on the network through certain important central points (such as routers, base stations, and access points) and implement security mechanisms at such points. Since ad hoc wireless networks do not have any such central points, these mechanisms cannot be applied in ad hoc wireless networks.

• **Lack of association:** Since these networks are dynamic in nature, a node can join or leave the network at any point of the time. If no proper authentication

mechanism is used for associating nodes with a network, an intruder would be able to join into the network quite easily and carry out his/her attacks.

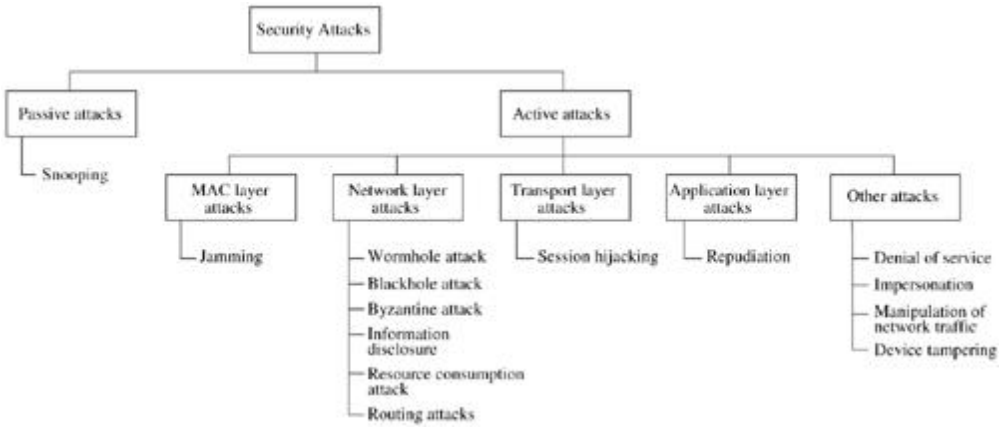
• **Limited resource availability:** Resources such as bandwidth, battery power, and computational power (to a certain extent) are scarce in ad hoc wireless networks. Hence, it is difficult to implement complex cryptography-based security mechanisms in such networks.

• **Physical vulnerability:** Nodes in these networks are usually compact and hand-held in nature. They could get damaged easily and are also vulnerable to theft.

3.19 NETWORK SECURITY ATTACKS Attacks on ad hoc wireless networks can be classified into two broad categories, namely, *passive* and *active* attacks. A passive attack does not disrupt the operation of the network; the adversary snoops the data exchanged in the network without altering it. Here, the requirement of confidentiality can be violated if an adversary is also able to interpret the data gathered through snooping. Detection of passive attacks is very difficult since the operation of the network itself does not get affected. One way of overcoming such problems is to use powerful encryption mechanisms to encrypt the data being transmitted, thereby making it impossible for eavesdroppers to obtain any useful information from the data overheard. An active attack attempts to alter or destroy the data being exchanged in the network, thereby disrupting the normal functioning of the network. Active attacks can be classified further into two categories, namely, *external* and *internal* attacks. External attacks are carried out by nodes that do not belong to the network. These attacks can be prevented by using standard security mechanisms such as encryption techniques and firewalls.[7] Internal attacks are from compromised nodes that are actually part of the network. Since the adversaries are already part of the network as authorized nodes, internal attacks are more severe and difficult to detect when compared to external attacks. [7] A firewall is used to separate a local network from the outside world. It is a software which works closely with a router program and filters all packets entering the network to determine whether or not to forward those packets toward their intended destinations. A firewall protects the resources of a private network from malicious intruders on foreign networks such as the Internet. In an ad hoc wireless network, the firewall software could be installed on each node on the network.

Figure 3.43 shows a classification of the different types of attacks possible in ad hoc wireless networks. The following sections describe the various attacks listed in the figure.

Figure 3.43. Classifications of attacks.



3.19.1 Network Layer Attacks This section lists and gives brief descriptions of the attacks pertaining to the network layer in the network protocol stack.

• **Wormhole attack:** In this attack, an attacker receives packets at one location in the network and tunnels them (possibly selectively) to another location in the network, where the packets are resent into the network . This tunnel between two colluding attackers is referred to as a wormhole. It could be established through a single long-range wireless link or even through a wired link between the two colluding attackers. Due to the broadcast nature of the radio channel, the attacker can create a wormhole even for packets not addressed to itself. Though no harm is done if the wormhole is used properly for efficient relaying of packets, it puts the attacker in a powerful position compared to other nodes in the network, which the attacker could use in a manner that could compromise the security of the network. If proper mechanisms are not employed to defend the network against wormhole attacks, most of the existing routing protocols for ad hoc wireless networks may fail to find valid routes.

• **Blackhole attack:** In this attack, a malicious node falsely advertises good paths (*e.g.*, shortest path or most stable path) to the destination node during the path-finding process (in on-demand routing protocols) or in the route update

messages (in table-driven routing protocols). The intention of the malicious node could be to hinder the path-finding process or to intercept all data packets being sent to the destination node concerned.

• **Byzantine attack:** Here, a compromised intermediate node or a set of compromised intermediate nodes works in collusion and carries out attacks such as creating routing loops, routing packets on non-optimal paths, and selectively dropping packets . Byzantine failures are hard to detect. The network would seem to be operating normally in the viewpoint of the nodes, though it may actually be exhibiting Byzantine behavior.

• **Information disclosure:** A compromised node may leak confidential or important information to unauthorized nodes in the network. Such information may include information regarding the network topology, geographic location of nodes, or optimal routes to authorized nodes in the network.

• **Resource consumption attack:** In this attack, a malicious node tries to consume/waste away resources of other nodes present in the network. The resources that are targeted are battery power, bandwidth, and computational power, which are only limitedly available in ad hoc wireless networks. The attacks could be in the form of unnecessary requests for routes, very frequent generation of beacon packets, or forwarding of stale packets to nodes. Using up the battery power of another node by keeping that node always busy by continuously pumping packets to that node is known as a sleep deprivation attack.

• **Routing attacks:** There are several types attacks mounted on the routing protocol which are aimed at disrupting the operation of the network. In what follows, the various attacks on the routing protocol are described briefly.

– **Routing table overflow:** In this type of attack, an adversary node advertises routes to non-existent nodes, to the authorized nodes present in the network. The main objective of such an attack is to cause an overflow of the routing tables, which would in turn prevent the creation of entries corresponding to new routes to authorized nodes. Proactive routing protocols are more vulnerable to this attack compared to reactive routing protocols.

– Routing table poisoning: Here, the compromised nodes in the networks send fictitious routing updates or modify genuine route update packets sent to other uncompromised nodes. Routing table poisoning may result in sub-optimal routing, congestion in portions of the network, or even make some parts of the network inaccessible.

– Packet replication: In this attack, an adversary node replicates stale packets. This consumes additional bandwidth and battery power resources available to the nodes and also causes unnecessary confusion in the routing process.

– Route cache poisoning: In the case of on-demand routing protocols (such as the AODV protocol), each node maintains a route cache which holds information regarding routes that have become known to the node in the recent past. Similar to routing table poisoning, an adversary can also poison the route cache to achieve similar objectives.

– Rushing attack: On-demand routing protocols that use duplicate suppression during the route discovery process are vulnerable to this attack . An adversary node which receives a *RouteRequest*packet from the source node floods the packet quickly throughout the network before other nodes which also receive the same *RouteRequest* packet can react. Nodes that receive the legitimate *RouteRequest* packets assume those packets to be duplicates of the packet already received through the adversary node and hence discard those packets. Any route discovered by the source node would contain the adversary node as one of the intermediate nodes. Hence, the source node would not be able to find secure routes, that is, routes that do not include the adversary node. It is extremely difficult to detect such attacks in ad hoc wireless networks.

3.19.2 Transport Layer Attacks

This section discusses an attack which is specific to the transport layer in the network protocol stack.

• **Session hijacking:** Here, an adversary takes control over a session between two nodes. Since most authentication processes are carried out only at the start of a session, once the session between two nodes gets established, the adversary

node masquerades as one of the end nodes of the session and hijacks the session.

3.19.3 Application Layer Attacks

This section briefly describes a security flaw associated with the application layer in the network protocol stack.

• **Repudiation:** In simple terms, repudiation refers to the denial or attempted denial by a node involved in a communication of having participated in all or part of the communication. non-repudiation is one of the important requirements for a security protocol in any communication network.

3.19.4 Other Attacks This section discusses security attacks that cannot strictly be associated with any specific layer in the network protocol stack. Multi-layer Attacks Multi-layer attacks are those that could occur in any layer of the network protocol stack. Denial of service and impersonation are some of the common multi-layer attacks. This section discusses some of the multi-layer attacks in ad hoc wireless networks.

• **Denial of Service:** In this type of attack, an adversary attempts to prevent legitimate and authorized users of services offered by the network from accessing those services. A denial of service (DoS) attack can be carried out in many ways. The classic way is to flood packets to any centralized resource (*e.g*., an access point) used in the network so that the resource is no longer available to nodes in the network, resulting in the network no longer operating in the manner it was designed to operate. This may lead to a failure in the delivery of guaranteed services to the end users. Due to the unique characteristics of ad hoc wireless networks, there exist many more ways to launch a DoS attack in such a network, which would not be possible in wired networks. DoS attacks can be launched against any layer in the network protocol stack . On the physical and MAC layers, an adversary could employ jamming signals which disrupt the on-going transmissions on the wireless channel. On the network layer, an adversary could take part in the routing process and exploit the routing protocol to disrupt the normal functioning of the network. For example, an adversary node could participate in a session but simply drop a certain number of packets, which may

lead to degradation in the QoS being offered by the network. On the higher layers, an adversary could bring down critical services such as the key management service (key management will be described in detail in the next section). Some of the DoS attacks are described below.

– **Jamming:** In this form of attack, the adversary initially keeps monitoring the wireless medium in order to determine the frequency at which the receiver node is receiving signals from the sender. It then transmits signals on that frequency so that error-free reception at the receiver is hindered. Frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS) (described in detail in the first chapter of this book) are two commonly used techniques that overcome jamming attacks.

– **SYN flooding:** Here, an adversary sends a large number of SYN packets[8] to a victim node, spoofing the return addresses of the SYN packets. On receiving the SYN packets, the victim node sends back acknowledgment (SYN-ACK) packets to nodes whose addresses have been specified in the received SYN packets. However, the victim node would not receive any ACK packet in return. In effect, a half-open connection gets created. The victim node builds up a table/data structure for holding information regarding all pending connections. Since the maximum possible size of the table is limited, the increasing number of half-open connections results in an overflow in the table. Hence, even if a connection request comes from a legitimate node at a later point of time, because of the table overflow, the victim node would be forced to reject the call request.

[8] SYN packets are used to establish an end-to-end session between two nodes at the transport layer.

– **Distributed DoS attack:** A more severe form of the DoS attack is the distributed DoS (DDoS) attack. In this attack, several adversaries that are distributed throughout the network collude and prevent legitimate users from accessing the services offered by the network.

• **Impersonation:** In impersonation attacks, an adversary assumes the identity and privileges of an authorized node, either to make use of network resources that may not be available to it under normal circumstances, or to disrupt the normal functioning of the network by injecting false routing information into

the network. An adversary node could masquerade as an authorized node using several methods. It could by chance guess the identity and authentication details of the authorized node (target node), or it could snoop for information regarding the identity and authentication of the target node from a previous communication, or it could circumvent or disable the authentication mechanism at the target node. A *man-in-the-middle* attack is another type of impersonation attack. Here, the adversary reads and possibly modifies, messages between two end nodes without letting either of them know that they have been attacked. Suppose two nodes *X* and *Y* are communicating with each other; the adversary impersonates node *Y* with respect to node *X* and impersonates node *X* with respect to node *Y,* exploiting the lack of third-party authentication of the communication between nodes *X* and *Y.* Device Tampering Unlike nodes in a wired network, nodes in ad hoc wireless networks are usually compact, soft, and hand-held in nature. They could get damaged or stolen easily.

3.20 KEY MANAGEMENT Having seen the various kinds of attacks possible on ad hoc wireless networks, we now look at various techniques employed to overcome the attacks. Cryptography is one of the most common and reliable means to ensure security. Cryptography is not specific to ad hoc wireless networks. It can be applied to any communication network. It is the study of the principles, techniques, and algorithms by which information is transformed into a disguised version which no unauthorized person can read, but which can be recovered in its original form by an intended recipient. In the parlance of cryptography, the original information to be sent from one person to another is called *plaintext.* This plaintext is converted into *ciphertext* by the process of encryption, that is, the application of certain algorithms or functions. An authentic receiver can decrypt/decode the ciphertext back into plaintext by the process of decryption. The processes of encryption and decryption are governed by *keys,* which are small amounts of information used by the cryptographic algorithms. When the key is to be kept secret to ensure the security of the system, it is called a secret key. The secure administration of cryptographic keys is called key management. The four main goals of cryptography are confidentiality, integrity, authentication (the receiver should be able to identify the sender and verify that the message actually came from

that sender), and non-repudiation. A detailed study of cryptography is presented in . There are two major kinds of cryptographic algorithms: symmetric key algorithms, which use the same key for encryption and decryption, and asymmetric key algorithms, which use two different keys for encryption and decryption. Symmetric key algorithms are usually faster to execute electronically, but require a secret key to be shared between the sender and receiver. When communication needs to be established among a group of nodes, each sender-receiver pair should share a key, which makes the system nonscalable. If the same key is used among more than two parties, a breach of security at any one point makes the whole system vulnerable. The asymmetric key algorithms are based on some mathematical principles which make it infeasible or impossible to obtain one key from another; therefore, one of the keys can be made public while the other is kept secret (private). This is called public key cryptography. Such systems are used extensively in practice, but are not provably secure. They rely upon the difficulty of solving certain mathematical problems, and the network would be open to attacks once the underlying mathematical problem is solved.

3.20.1 Symmetric Key Algorithms Symmetric key algorithms rely on the presence of the shared key at both the sender and receiver, which has been exchanged by some previous arrangement. There are two kinds of symmetric key algorithms, one involving block ciphers and the other stream ciphers. A block cipher is an encryption scheme in which the plaintext is broken into fixed-length segments called blocks, and the blocks are encrypted one at a time. The simplest examples include substitution and transposition. In substitution, each alphabet of the plaintext is substituted by another in the ciphertext, and this table mapping the original and the substituted alphabet is available at both the sender and receiver. A transposition cipher permutes the alphabet in the plaintext to produce the ciphertext. Figure 3.44 (a) illustrates the encryption using substitution, and Figure 3.44 (b) shows a transposition cipher. The block length used is five.

Figure 3.44. Substitution and transposition.

Original Alphabet	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Substitution	E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
Plaintext	EVERYDAY CREATES A HISTORY EVERY DAYCR EATES AHIST ORY
Ciphertext	IZIVC HECGV IEXIW ELMWX SVC

(a)

Transposition	1 2 3 4 5 ↓ 3 5 1 4 2
Plaintext	EVERYDAY CREATES A HISTORY EVERY DAYCR EATES AHIST ORY
Ciphertext	EYERV YRDCA TSEEA ITASH YOR

(b)

A stream cipher is, in effect, a block cipher of block length one. One of the simplest stream ciphers is the Vernam cipher, which uses a key of the same length as the plaintext for encryption. For example, if the plaintext is the binary string 10010100, and the key is 01011001, then the encrypted string is given by the XOR of the plaintext and key, to be 11001101. The plaintext is again recovered by XORing the ciphertext with the same key. If the key is randomly chosen, transported securely to the receiver, and used for only one communication, this forms the one-time pad which has proven to be the most secure of all cryptographic systems. The only bottleneck here is to be able to securely send the key to the receiver.

3.20.2 Asymmetric Key Algorithms Asymmetric key (or public key) algorithms use different keys at the sender and receiver ends for encryption and decryption, respectively. Let the encryption process be represented by a function *E*, and

decryption by *D*. Then the plaintext *m* is transformed into the ciphertext *c* as $c = E(m)$. The receiver then decodes *c* by applying *D*. Hence, *D* is such that $m = D(c) = D(E(m))$. When this asymmetric key concept is used in public key algorithms, the key *E* is made public, while *D* is private, known only to the intended receiver. Anyone who wishes to send a message to this receiver encrypts it using *E*. Though *c* can be overheard by adversaries, the function *E* is based on a computationally difficult mathematical problem, such as the factorization of large prime numbers. Hence, it is not possible for adversaries to derive *D* given *E*. Only the receiver can decrypt *c* using the private key *D*. A very popular example of public key cryptography is the RSA system developed by Rivest, Shamir, and Adleman, which is based on the integer factorization problem. Digital signatures schemes are also based on public key encryption. In these schemes, the functions *E* and *D* are chosen such that $D(E(m)) = E(D(m)) = m$ for any message *m*. These are called reversible public key systems. In this case, the person who wishes to sign a document encrypts it using his/her private key *D*, which is known only to him/her. Anybody who has his/her public key *E* can decrypt it and obtain the original document, if it has been signed by the corresponding sender. In practice, a trusted third party (TTP) is agreed upon in advance, who is responsible for issuing these digital signatures (*D* and *E* pairs) and for resolving any disputes regarding the signatures. This is usually a governmental or business organization.

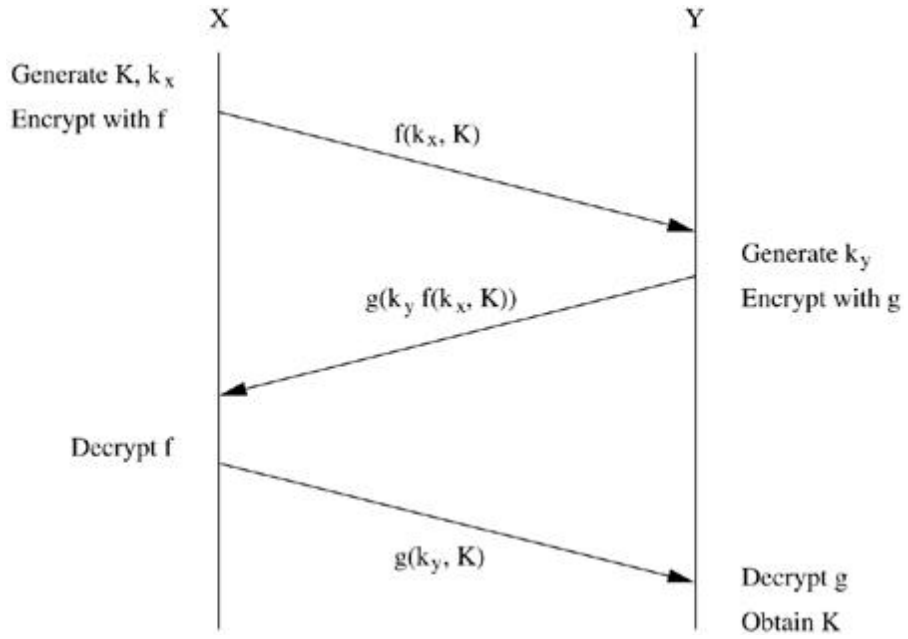
3.20.3 Key Management Approaches The primary goal of key management is to share a secret (some information) among a specified set of participants. There are several methods that can be employed to perform this operation, all of them requiring varying amounts of initial configuration, communication, and computation. The main approaches to key management are key predistribution, key transport, key arbitration, and key agreement .

Key Predistribution

Key predistribution, as the name suggests, involves distributing keys to all interested parties before the start of communication. This method involves much less communication and computation, but all participants must be known *a priori*, during the initial configuration. Once deployed, there is no mechanism to

include new members in the group or to change the key. As an improvement over the basic predistribution scheme, sub-groups may be formed within the group, and some communication can be restricted to a subgroup. However, the formation of sub-groups is also an *a priori* decision with no flexibility during the operation. Key Transport In key transport systems, one of the communicating entities generates keys and transports them to the other members. The simplest scheme assumes that a shared key already exists among the participating members. This prior shared key is used to encrypt a new key and is transmitted to all corresponding nodes. Only those nodes which have the prior shared key can decrypt it. This is called the key encrypting key (KEK) method. However, the existence of a prior key cannot always be assumed. If the public key infrastructure (PKI) is present, the key can be encrypted with each participant's public key and transported to it. This assumes the existence of a TTP, which may not be available for ad hoc wireless networks. An interesting method for key transport without prior shared keys is the Shamir's three-pass protocol . The scheme is based on a special type of encryption called commutative encryption schemes [which are reversible and composable (composition of two functions f and g is defined as $f(g(x))$)]. Consider two nodes X and Y which wish to communicate. Node X selects a key K which it wants to use in its communication with node Y. It then generates another random key k_x, using which it encrypts $K$with f, and sends to node Y. Node Y encrypts this with a random key k_y using g, and sends it back to node X. Now, node X decrypts this message with its key k_x, and after applying the inverse function f^{-1}, sends it to node Y. Finally, node Y decrypts the message using k_y and g^{-1} to obtain the key K. The message exchanges of the protocol are illustrated in Figure 3.45.

Figure 3.45. Shamir's three-pass protocol.



Key Arbitration

Key arbitration schemes use a central arbitrator to create and distribute keys among all participants. Hence, they are a class of key transport schemes. Networks which have a fixed infrastructure use the AP as an arbitrator, since it does not have stringent power or computation constraints. In ad hoc wireless networks, the problem with implementation of arbitrated protocols is that the arbitrator has to be powered on at all times to be accessible to all nodes. This leads to a power drain on that particular node. An alternative would be to make the keying service distributed, but simple replication of the arbitration at different nodes would be expensive for resource-constrained devices and would offer many points of vulnerability to attacks. If any one of the replicated arbitrators is attacked, the security of the whole system breaks down. Key Agreement Most key agreement schemes are based on asymmetric key algorithms. They are used when two or more people want to agree upon a secret key, which will then be used for further communication. Key agreement protocols are used to establish a secure context over which a session can be run, starting with many parties who wish to communicate and an insecure channel. In group key agreement schemes, each participant contributes a part to the secret key. These need the least amount of preconfiguration, but such schemes have high computational complexity. The most popular key agreement schemes use the

Diffie-Hellman exchange , an asymmetric key algorithm based on discrete logarithms.

3.20.4 Key Management in Ad Hoc Wireless Networks

Ad hoc wireless networks pose certain specific challenges in key management due to the lack of infrastructure in such networks. Three types of infrastructure have been identified in , which are absent in ad hoc wireless networks. The first is the network infrastructure, such as dedicated routers and stable links, which ensure communication with all nodes. The second missing infrastructure is services such as name resolution, directory, and TTPs. The third missing infrastructure in ad hoc wireless networks is the administrative support of certifying authorities. Password-Based Group Systems Several solutions for group keying in ad hoc wireless networks have been suggested in. The example scenario for implementation is a meeting room, where different mobile devices want to start a secure session. Here, the parties involved in the session are to be identified based on their location, that is, all devices in the room can be part of the session. Hence, relative location is used as the criterion for access control. If a TTP which knows the location of the participants exists, then it can implement location-based access control. A prior shared secret can be obtained by a physically more secure medium such as a wired network. This secret can be obtained by plugging onto a wired network first, before switching to the wireless mode. A password-based system has been explored where, in the simplest case, a long string is given as the password for users for one session. However, human beings tend to favor natural language phrases as passwords, over randomly generated strings. Such passwords, if used as keys directly during a session, are very weak and open to attack because of high redundancy, and the possibility of reuse over different sessions. Hence, protocols have been proposed to derive a strong key (not vulnerable to attacks) from the weak passwords given by the participants. This password-based system could be two-party, with a separate exchange between any two participants, or it could be for the whole group, with a leader being elected to preside over the session. Leader election is a special case of establishing an order among all participants. The protocol used is as follows. Each participant generates a random number, and

sends it to all others. When every node has received the random number of every other node, a common predecided function is applied on all the numbers to calculate a *reference value.* The nodes are ordered based on the difference between their random number and the reference value.

Threshold Cryptography

Public key infrastructure (PKI) enables the easy distribution of keys and is a scalable method. Each node has a public/private key pair, and a certifying authority (CA) can bind the keys to the particular node. But the CA has to be present at all times, which may not be feasible in ad hoc wireless networks. It is also not advisable to simply replicate the CA at different nodes. In , a scheme based on threshold cryptography has been proposed by which n servers exist in the ad hoc wireless network, out of which any (t+1) servers can jointly perform any arbitration or authorization successfully, but t servers cannot perform the same. Hence, up to t compromised servers can be tolerated. This is called an (n, t + 1) configuration, where $n \geq 3t + 1$. To sign a certificate, each server generates a partial signature using its private key and submits it to a combiner. The combiner can be any one of the servers. In order to ensure that the key is combined correctly, t + 1 combiners can be used to account for at most t malicious servers. Using t + 1 partial signatures (obtained from itself and t other servers), the combiner computes a signature and verifies its validity using a public key. If the verification fails, it means that at least one of the t + 1 keys is not valid, so another subset of t + 1 partial signatures is tried. If the combiner itself is malicious, it cannot get a valid key, because the partial signature of itself is always invalid. The scheme can be applied to asynchronous networks, with no bound on message delivery or processing times. This is one of the strengths of the scheme, as the requirement of synchronization makes the system vulnerable to DoS attacks. An adversary can delay a node long enough to violate the synchrony assumption, thereby disrupting the system. Sharing a secret in a secure manner alone does not completely fortify a system. Mobile adversaries can move from one server to another, attack them, and get hold of their private keys. Over a period of time, an adversary can have more than t private keys. To counter this, *share refreshing* has been proposed, by which

servers create a new independent set of shares (the partial signatures which are used by the servers) periodically. Hence, to break the system, an adversary has to attack and capture more than *t*servers within the period between two successive refreshes; otherwise, the earlier share information will no longer be valid. This improves protection against mobile adversaries. Self-Organized Public Key Management for Mobile Ad Hoc Networks The authors of have proposed a completely self-organized public key system for ad hoc wireless networks. This makes use of absolutely no infrastructure – TTP, CA, or server – even during initial configuration. The users in the ad hoc wireless network issue certificates to each other based on personal acquaintance. A certificate is a binding between a node and its public key. These certificates are also stored and distributed by the users themselves. Certificates are issued only for a specified period of time and contain their time of expiry along with them. Before it expires, the certificate is updated by the user who had issued the certificate. Initially, each user has a local repository consisting of the certificates issued by him and the certificates issued by other users to him. Hence, each certificate is initially stored twice, by the issuer and by the person for whom it is issued. Periodically, certificates from neighbors are requested and the repository is updated by adding any new certificates. If any of the certificates are conflicting (*e.g.*, the same public key to different users, or the same user having different public keys), it is possible that a malicious node has issued a false certificate. A node then labels such certificates as *conflicting* and tries to resolve the conflict. Various methods exist to compare the confidence in one certificate over another. For instance, another set of certificates obtained from another neighbor can be used to take a majority decision. This can be used to evaluate the trust in other users and detect malicious nodes. If the certificates issued by some node are found to be wrong, then that node may be assumed to be malicious. The authors of define a certificate graph as a graph whose vertices are public keys of some nodes and whose edges are public-key certificates issued by users. When a user *X* wants to obtain the public key of another user *Y*, he/she finds a chain of valid public key certificates leading to *Y*. The chain is such that the first hop uses an edge from *X*, that is, a certificate issued by *X*, the last hop leads into *Y*(this is a certificate issued to *Y*), and all intermediate nodes are trusted through the

previous certificate in the path. The protocol assumes that trust is transitive, which may not always be valid. Having seen the various key management techniques employed in ad hoc wireless networks, we now move on to discuss some of the security-aware routing schemes for ad hoc wireless networks.

3.21 SECURE ROUTING IN AD HOC WIRELESS NETWORKS Unlike the traditional wired Internet, where dedicated routers controlled by the Internet service providers (ISPs) exist, in ad hoc wireless networks, nodes act both as regular terminals (source or destination) and also as routers for other nodes. In the absence of dedicated routers, providing security becomes a challenging task in these networks. Various other factors which make the task of ensuring secure communication in ad hoc wireless networks difficult include the mobility of nodes, a promiscuous mode of operation, limited processing power, and limited availability of resources such as battery power, bandwidth, and memory. In the following sections, we show how some of the well known traditional routing protocols for ad hoc networks fail to provide security. Some of the mechanisms proposed for secure routing are also discussed.

3.21.1 Requirements of a Secure Routing Protocol for Ad Hoc Wireless Networks The fundamental requisites of a secure routing protocol for ad hoc wireless networks are listed as follows:

• **Detection of malicious nodes:** A secure routing protocol should be able to detect the presence of malicious nodes in the network and should avoid the participation of such nodes in the routing process. Even if such malicious nodes participate in the route discovery process, the routing protocol should choose paths that do not include such nodes.

• **Guarantee of correct route discovery:** If a route between the source and the destination nodes exists, the routing protocol should be able to find the route, and should also ensure the correctness of the selected route.

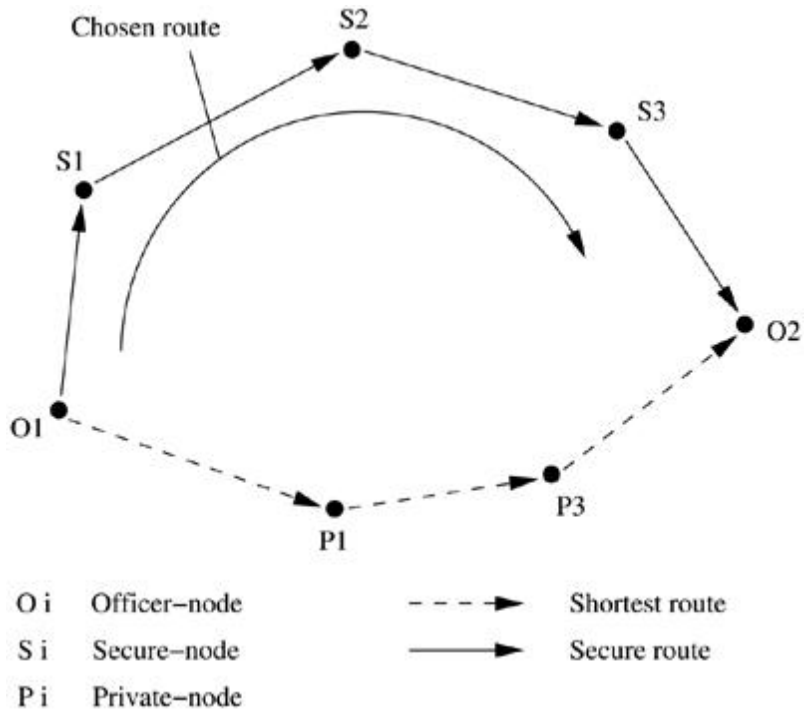
• **Confidentiality of network topology:** As explained, an information disclosure attack may lead to the discovery of the network topology by the malicious nodes. Once the network topology is known, the attacker may try to study the traffic pattern in the network. If some of the nodes are found to be

more active compared to others, the attacker may try to mount (*e.g.*, DoS) attacks on such bottleneck nodes. This may ultimately affect the on-going routing process. Hence, the confidentiality of the network topology is an important requirement to be met by the secure routing protocols.

• **Stability against attacks:** The routing protocol must be self-stable in the sense that it must be able to revert to its normal operating state within a finite amount of time after a passive or an active attack. The routing protocol should take care that these attacks do not permanently disrupt the routing process. The protocol must also ensure Byzantine robustness, that is, the protocol should work properly even if some of the nodes, which were earlier participating in the routing process, turn out to become malicious at a later point of time or are intentionally damaged. In the following sections, some of the security-aware routing protocols proposed for ad hoc wireless networks are discussed.

3.21.2 Security-Aware Ad Hoc Routing Protocol The security-aware ad hoc routing (SAR) protocol uses security as one of the key metrics in path finding. A framework for enforcing and measuring the attributes of the security metric has been provided in. This framework also enables the use of different levels of security for different applications that use SAR for routing. In ad hoc wireless networks, communication between end nodes through possibly multiple intermediate nodes is based on the fact that the two end nodes trust the intermediate nodes. SAR defines *level of trust* as a metric for routing and as one of the attributes for security to be taken into consideration while routing. The routing protocol based on the level of trust is explained using Figure 3.46. As shown in Figure 3.46, two paths exist between the two officers $O1$ and$O2$ who want to communicate with each other. One of these paths is a shorter path which runs through private nodes whose trust levels are very low. Hence, the protocol chooses a longer but secure path which passes through other secure (officer) nodes.

Figure 3.46. Illustration of the level of trust metric.



The SAR protocol can be explained using any one of the traditional routing protocols. This section explains SAR using the AODV protocol . In the AODV protocol, the source node broadcasts a *RouteRequest* packet to its neighbors. An intermediate node, on receiving a*RouteRequest* packet, forwards it further if it does not have a route to the destination. Otherwise, it initiates a *RouteReply* packet back to the source node using the reverse path traversed by the*RouteRequest* packet. In SAR, a certain level of security is incorporated into the packet-forwarding mechanism. Here, each packet is associated with a security level which is determined by a number calculation method (explained later in this section). Each intermediate node is also associated with a certain level of security. On receiving a packet, the intermediate node compares its level of security with that defined for the packet. If the node's security level is less than that of the packet, the *RouteRequest* is simply discarded. If it is greater, the node is considered to be a secure node and is permitted to forward the packet in addition to being able to view the packet. If the security levels of the intermediate node and the received packet are found to be equal, then the intermediate node will not be able to view the packet (which can be ensured using a proper authentication mechanism); it just forwards the packet further. Nodes of equal levels of trust distribute a common key among themselves and

with those nodes having higher levels of trust. Hence, a hierarchical level of security could be maintained. This ensures that an encrypted packet can be decrypted (using the common key) only by nodes of the same or higher levels of security compared to the level of security of the packet. Different levels of trust can be defined using a number calculated based on the level of security required. It can be calculated using many methods. Since timeliness, in-order delivery of packets, authenticity, authorization, integrity, confidentiality, and non-repudiation are some of the desired characteristics of a routing protocol, a suitable number can be defined for the trust level for nodes and packets based on the number of such characteristics taken into account. The SAR mechanism can be easily incorporated into the traditional routing protocols for ad hoc wireless networks. It could be incorporated into both on demand and table-driven routing protocols. The SAR protocol allows the application to choose the level of security it requires. But the protocol requires different keys for different levels of security. This tends to increase the number of keys required when the number of security levels used increases.

3.21.3 Secure Efficient Ad Hoc Distance Vector Routing Protocol

Secure efficient ad hoc distance vector (SEAD) routing protocol, is a secure ad hoc routing protocol based on the destination-sequenced distance vector (DSDV) routing protocol. This protocol is mainly designed to overcome security attacks such as DoS and resource consumption attacks. The operation of the routing protocol does not get affected even in the presence of multiple uncoordinated attackers corrupting the routing tables. The protocol uses a one-way hash function and does not involve any asymmetric cryptographic operation.

Distance Vector Routing

Distance vector routing protocols belong to the category of table-driven routing protocols. Each node maintains a routing table containing the list of all known routes to various destination nodes in the network. The metric used for routing is the distance measured in terms of hop-count. The routing table is updated periodically by exchanging routing information. An alternative to this approach is *triggered updates,* in which each node broadcasts routing updates only if its

routing table gets altered. The DSDV protocol for ad hoc wireless networks uses *sequence number* tags to prevent the formation of loops, to counter the count-to-infinity problem, and for faster convergence. When a new route update packet is received for a destination, the node updates the corresponding entry in its routing table only if the sequence number on the received update is greater than that recorded with the corresponding entry in the routing table. If the received sequence number and the previously recorded sequence number are both equal, but if the routing update has a new value for the routing metric (distance in number of hops), then in this case also the update is effected. Otherwise, the received update packet is discarded. DSDV uses triggered updates (for important routing changes) in addition to the regular periodic updates. A slight variation of DSDV protocol known as DSDV-SQ (DSDV for sequence numbers) initiates triggered updates on receiving a new sequence number update. One-Way Hash Function SEAD uses authentication to differentiate between updates that are received from non-malicious nodes and malicious nodes. This minimizes resource consumption attacks caused by malicious nodes. SEAD uses a one-way hash function for authenticating the updates. A one-way hash function (H) generates a one-way hash chain (h_1, h_2, ...). The function H maps an input bit-string of any length to a fixed length bit-string, that is, $H: (0, 1)^* \rightarrow (0, 1)_p$, where p is the length in bits of the output bit-string. To create a one-way hash chain, a node generates a random number with initial value x $(0, 1)_p$. h_0, the first number in the hash chain is initialized to x. The remaining values in the chain are computed using the general formula, $hi = H(h_{i-1})$ for $0 \leq i \leq n$, for some n. Now we shall see how the one-way hash function incorporates security into the existing DSDV-SQ routing protocol. The SEAD protocol assumes an upper bound on the metric used. For example, if the metric used is distance, then the upper bound value m - 1 defines the maximum diameter (maximum of lengths of all the routes between a pair of nodes) of the ad hoc wireless network. Hence, the routing protocol ensures that no route of length greater than m hops exists between any two nodes. If the sequence of values calculated by a node using the hash function H is given by (h_1, h_2, ..., h_n), where n is divisible by m, then for a routing table entry with sequence number i, let . If the metric j (distance) used for that routing table entry is $0 \leq j \leq m$ - 1,

then the value h_{km+j} is used to authenticate the routing update entry for that sequence number *i* and that metric *j*. Whenever a route update message is sent, the node appends the value used for authentication along with it. If the authentication value used is h_{km+j} , then the attacker who tries to modify this value can do so only if he/she knows h_{km+j-1}. Since it is a one-way hash chain, calculating h_{km+j-1} becomes impossible. An intermediate node, on receiving this authenticated update, calculates the new hash value based on the earlier updates (h_{km+j-1}), the value of the metric, and the sequence number. If the calculated value matches with the one present in the route update message, then the update is effected; otherwise, the received update is just discarded. SEAD avoids routing loops unless the loop contains more than one attacker. This protocol could be implemented easily with slight modifications to the existing distance vector routing protocols. The protocol is robust against multiple uncoordinated attacks. The SEAD protocol, however, would not be able to overcome attacks where the attacker uses the same metric and sequence number which were used by the recent update message, and sends a new routing update.

3.21.4 Authenticated Routing for Ad Hoc Networks Authenticated routing for ad hoc networks (ARAN) routing protocol, based on cryptographic certificates, is a secure routing protocol which successfully defeats all identified attacks in the network layer. It takes care of authentication, message integrity, and non-repudiation, but expects a small amount of prior security coordination among nodes. In, vulnerabilities and attacks specific to AODV and DSR protocols are discussed and the two protocols are compared with the ARAN protocol. During the route discovery process of ARAN, the source node broadcasts *RouteRequest* packets. The destination node, on receiving the *RouteRequest* packets, responds by unicasting back a reply packet on the selected path. The ARAN protocol uses a preliminary cryptographic certification process, followed by an end-to-end route authentication process, which ensures secure route establishment. Issue of Certificates This section discusses the certification process in which the certificates are issued to the nodes in the ad hoc wireless network. There exists an authenticated trusted server whose public key is known to all legal nodes in the network. The ARAN protocol assumes that keys are generated *a priori* by the server and distributed to all nodes in the network. The protocol does not

specify any specific key distribution algorithm. On joining the network, each node receives a certificate from the trusted server. The certificate received by a node *A* from the trusted server *T* looks like the following: **(3. 12.1)** Here, IP_A, K_{A+}, *t*, *e*, and K_{T-} represent the IP address of node *A*, the public key of node *A*, the time of creation of the certificate, the time of expiry of the certificate, and the private key of the server, respectively. End-to-End Route Authentication The main goal of this end-to-end route authentication process is to ensure that the correct intended destination is reached by the packets sent from the source node. The source node *S* broadcasts a*RouteRequest/RouteDiscovery* packet destined to the destination node *D*. The *RouteRequest* packet contains the packet identifier [route discovery process (RDP)], the IP address of the destination (IP_D), the certificate of the source node *S* ($Cert_S$), the current time (*t*), and nonce N_S. The process can be denoted as below. Here, K_{S-} is the private key of the source node *S*. **(3.21.2)** Whenever the source sends a route discovery message, it increments the value of nonce. Nonce is a counter used in conjunction with the time-stamp in order to make the nonce recycling easier. When a node receives an RDP packet from the source with a higher value of the source's nonce than that in the previously received RDP packets from the same source node, it makes a record of the neighbor from which it received the packet, encrypts the packet further with its own certificate, and broadcasts it further. The process can be denoted as follows: **(3.21.3)** An intermediate node *B*, on receiving an RDP packet from a node *A*, removes its neighbor's certificate, inserts its own certificate, and broadcasts the packet further. The destination node, on receiving anRDP packet, verifies node S's certificate and the tuple (N_S , *t*) and then replies with the *RouteReply*packet (REP). The destination unicasts the REP packet to the source node along the reverse path as follows: **(3.21.4)** where node *X* is the neighbor of the destination node *D*, which had originally forwarded the RDP packet to node *D*. The REP packet follows the same procedure on the reverse path as that followed by the route discovery packet. An error message is generated if the time-stamp or n once do not match the requirements or if the certificate fails. The error message looks similar to the other packets except that the packet identifier is replaced by the ERR message. Table 3.12 shows a comparison between the AODV, DSR, and ARAN protocols with respect to their security-

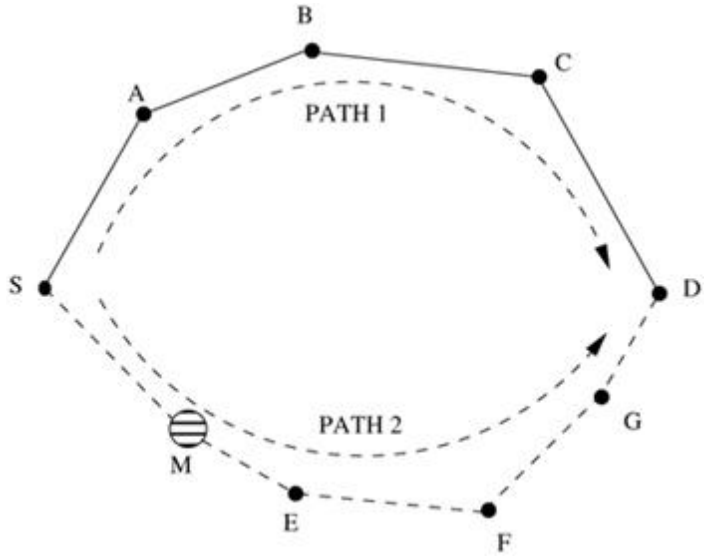
related features. ARAN remains robust in the presence of attacks such as unauthorized participation, spoofed route signaling, fabricated routing messages, alteration of routing messages, securing shortest paths, and replay attacks.

Table 3.12. Comparison of vulnerabilities of ARAN with DSR and AODV protocols

Attacks	Protocols		
	AODV	DSR	ARAN
Modifications required during remote redirection	Sequence number and hop-counts	Source routes	None
Tunneling during remote redirection	Yes	Yes	Yes
Spoofing	Yes	Yes	No
Cache poisoning	No	Yes	No

3.21.5 Security-AwareAODV Protocol This section discusses security solutions that address a particular security flaw in the AODV routing protocol . AODV is an on-demand routing protocol where the route discovery process is initiated by sending *RouteRequest* packets only when data packets arrive at a node for transmission. A malicious intermediate node could advertise that it has the shortest path to the destination, thereby redirecting all the packets through itself. This is known as a blackhole attack, as explained in Section 3.100.1. The blackhole attack is illustrated in Figure 3.47. Let node *M* be the malicious node that enters the network. It advertises that it has the shortest path to the destination node *D* when it receives the *RouteRequest*packet sent by node *S*. The attacker may not be able to succeed if node *A*, which also receives the*RouteRequest* packet from node *S*, replies earlier than node *M*. But a major advantage for the malicious node is that it does not have to search its routing table for a route to the destination. Also, the*RouteReply* packets originate directly from the malicious node and not from the destination node. Hence, the malicious node would be able to reply faster than node *A*, which would have to search its routing table for a route to the destination node. Thus, node *S* may tend to establish a route to destination *D* through the malicious node *M*, allowing node *M* to listen to all packets meant for the destination node.

Figure 3.47. Illustration of blackhole problem.



Solutions for the Blackhole Problem

One of the solutions for the blackhole problem is to restrict the intermediate nodes from originating*RouteReply* packets. Only the destination node would be permitted to initiate *RouteReply* packets. Security is still not completely assured, since the malicious node may lie in the path chosen by the destination node. Also, the delay involved in the route discovery process increases as the size of the network increases. In another solution to this problem, suggested in , as soon as the *RouteReply*packet is received from one of the intermediate nodes, another *RouteRequest* packet is sent from the source node to the neighbor node of the intermediate node in the path. This is to ensure that such a path exists from the intermediate node to the destination node. For example, let the source node send*RouteRequest* packets and receive *RouteReply* through the intermediate malicious node *M*. The*RouteReply* packet of node *M* contains information regarding its next-hop neighbor nodes. Let it contain information about the neighbor node *E*. Then, as shown in Figure 3.48, the source node *S* sends*FurtherRouteRequest* packets to this neighbor node *E*. Node *E* responds by sending a*FurtherRouteReply* packet to source node *S*. Since node *M* is a malicious node which is not present in the routing list of node *E*, the *FurtherRouteReply* packet sent by node *E* will not contain a route to the malicious node *M*. But if it contains a route to the destination node *D*, then the

new route to the destination through node *E* is selected, and the earlier selected route through node *M* is rejected. This protocol completely eliminates the blackhole attack caused by a single attacker. The major disadvantage of this scheme is that the control overhead of the routing protocol increases considerably. Also, if the malicious nodes work in a group, this protocol fails miserably.

Figure 3.48. Propagation of *FurtherRouteRequest* and *FurtherRouteReply*.