Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Ad Hoc Networks 4 (2006) 83–124

www.elsevier.com/locate/adhoc

Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions[☆]

T. Bheemarjuna Reddy, I. Karthigeyan, B.S. Manoj, C. Siva Ram Murthy [*]

Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600036, India

Received 15 February 2004; accepted 14 April 2004
Available online 5 June 2004

Abstract

An ad hoc wireless network (AWN) is a collection of mobile hosts forming a temporary network on the fly, without using any fixed infrastructure. Characteristics of AWNs such as lack of central coordination, mobility of hosts, dynamically varying network topology, and limited availability of resources make QoS provisioning very challenging in such networks. In this paper, we describe the issues and challenges in providing QoS for AWNs and review some of the QoS solutions proposed. We first provide a layer-wise classification of the existing QoS solutions, and then discuss each of these solutions.

Keywords: Ad hoc wireless networks; Quality of service; Real-time traffic; QoS routing; QoS frameworks

1. Introduction

Ad hoc wireless networks (AWNs) are zero configuration, self organizing, and highly dynamic networks formed by a set of mobile hosts connected through wireless links. These networks can be formed on the fly, without requiring any fixed infrastructure. As these are infrastructureless networks, each node should act also as a router. Throughout this paper, the terms "mobile host", "node", and "station" are used interchangeably. As a router, the mobile host represents an intermediate node which forwards traffic on behalf of other nodes. If the destination node is not within the transmission range of the source node, the source node takes help of the intermediate nodes to communicate with the destination node. Tactical communication required on battlefields, among a fleet of ships, or among a group

[☆] This work was supported by the iNautix Technologies India Private Limited, Chennai, India and the Department of Science and Technology, New Delhi, India.

[*] Corresponding author. Tel.: +91 44 2257 8340; fax: +91 44 2257 8352.
E-mail addresses: arjun@cs.iitm.ernet.in (T.B. Reddy), ikarthik@cs.iitm.ernet.in (I. Karthigeyan), bsmanoj@cs.iitm.ernet.in (B.S. Manoj), murthy@iitm.ernet.in (C. Siva Ram Murthy).

1570-8705/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2004.04.008

of armored vehicles are some of the military applications of these networks. Civilian applications include peer-to-peer computing and file sharing, collaborated computing in a conference hall, and search and rescue operations.

2. Quality of service

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

After receiving a service request from the user, the first task is to find a suitable loop-free path from the source to the destination that will have the necessary resources available to meet the QoS requirements of the desired service. This process is known as QoS routing. After finding a suitable path, a resource reservation protocol is employed to reserve necessary resources along that path. QoS guarantees can be provided only with appropriate resource reservation techniques. For example, consider the network shown in Fig. 1. The attributes of each link are shown in a tuple $\langle BW, D\rangle$, where BW and D represent available bandwidth in Mbps and delay [1] in milliseconds. Suppose a packet-flow from node B to node G requires a bandwidth guarantee of 4 Mbps. QoS routing searches for a path that has sufficient bandwidth to meet the bandwidth requirement of the flow. Here, 6 paths are available between nodes B and G as shown in Table 1. QoS routing selects path 3 (i.e., $B \to C \to F \to G$) because, out of the available paths, path 3 alone meets the bandwidth constraint of 4 Mbps for the flow. The end-to-end bandwidth of a path is equal to the bandwidth of the bottleneck link (i.e., link having minimum bandwidth among all the links of a path). The end-to-end delay of a path is equal to the sum of delays of all the links of a path. Clearly path 3 is not optimal in terms of hop count and/or end-to-end delay parameters, while path 1 is optimal in terms of both hop count and end-to-end delay parameters. Hence, QoS routing has to select a suitable path that meets the QoS constraints specified in the service request made by the user. QoS routing has been described in detail later in this paper.

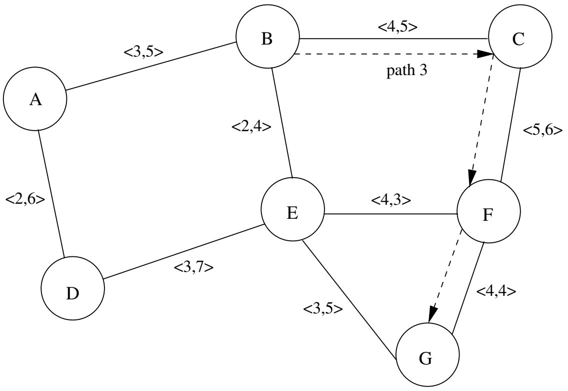


Fig. 1. An example of QoS routing in ad hoc wireless network.

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

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

Table 1
Available paths from node B to node G

No.	Path	Hop count	BW (Mbps)	Delay (ms)
1	$B \rightarrow E \rightarrow G$	2	2	9
2	$B \rightarrow E \rightarrow F \rightarrow G$	3	2	11
3	$B \rightarrow C \rightarrow F \rightarrow G$	3	4	15
4	$B \rightarrow C \rightarrow F \rightarrow E \rightarrow G$	4	3	19
5	$B \rightarrow A \rightarrow D \rightarrow E \rightarrow G$	4	2	23
6	$B \rightarrow A \rightarrow D \rightarrow E \rightarrow F \rightarrow G$	5	2	25

coordination, mobility of hosts, and limited availability of resources make QoS provisioning very challenging.

2.1. QoS parameters in ad hoc wireless networks

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

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

3. Issues and challenges in providing QoS in ad hoc wireless networks

Providing QoS support in AWNs is an active research area. AWNs have certain unique characteristics that pose several difficulties in provisioning QoS. A detailed discussion on how the characteristics of AWNs affects QoS provisioning is given below:

- *Dynamically varying network topology:* Since the nodes in an ad hoc wireless network do not have any restriction on mobility, the network topology changes dynamically. Hence the admitted QoS sessions may suffer due to frequent path breaks, thereby requiring such sessions to be re-established over new paths. The delay incurred in re-establishing a QoS session may cause some of the packets belonging to that session to miss their delay targets/deadlines, which is not acceptable for applications that have stringent QoS requirements.
- *Imprecise state information:* In most cases, the nodes in an ad hoc wireless network maintain both the link-specific state information and flow-specific state information. The link-specific state information includes bandwidth, delay, delay jitter, loss rate, error rate, stability, cost, and distance values for each link. The flow-specific information includes session ID, source address, destination address, and QoS requirements of the flow (such as maximum bandwidth requirement, minimum bandwidth requirement, maximum delay, and maximum delay jitter). The state information is inherently imprecise due to dynamic changes in network topology and channel characteristics. Hence routing decisions may not be accurate, resulting in some of the real-time packets missing their deadlines.
- *Lack of central coordination:* Unlike wireless LANs and cellular networks, AWNs do not have central controllers to coordinate the activity of nodes. This further complicates QoS provisioning in AWNs.
- *Error prone shared radio channel:* The radio channel is a broadcast medium by nature. During propagation through the wireless medium

the radio waves suffer from several impairments such as attenuation, multi-path propagation, and interference (from other wireless devices operating in the vicinity).

- *Hidden terminal problem:* The hidden terminal problem is inherent in AWNs. This problem occurs when packets originating from two or more sender nodes, which are not within the direct transmission range of each other, collide at a common receiver node. It necessitates retransmission of packets, which may not be acceptable for flows that have stringent QoS requirements. The RTS/CTS control packet exchange mechanism, proposed in [1] and adopted later in the IEEE 802.11 standard [2], reduces the hidden terminal problem only to a certain extent. BTMA [3] and DBTMA [4] provide two important solutions for this problem.
- *Limited resource availability:* Resources such as bandwidth, battery life, storage space, and processing capability are limited in AWNs. Out of these, bandwidth and battery life are very critical resources, the availability of which significantly affects the performance of the QoS provisioning mechanism. Hence efficient resource management mechanisms are required for optimal utilization of these scarce resources.
- *Insecure medium:* Due to the broadcast nature of the wireless medium, communication through a wireless channel is highly insecure. Hence security is an important issue in AWNs, especially for military and tactical applications. AWNs are susceptible to attacks such as eavesdropping, spoofing, denial of service, message distortion, and impersonation. Without sophisticated security mechanisms, it is very difficult to provide secure communication guarantees.

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

- *Hard state vs soft state resource reservation:* QoS resource reservation is one of the very important components of any QoS framework (a QoS framework is a complete system that provides required/promised services to each user or application). It is responsible for reserving resources at all intermediate nodes along the path from the source to the destination as requested by the QoS session. QoS resource reservation mechanisms can be broadly classified into two categories, *hard state* and *soft state* reservation mechanisms. In hard state resource reservation schemes, resources are reserved at all intermediate nodes along the path from the source to the destination throughout the duration of the QoS session. If such a path is broken due to network dynamics, these reserved resources have to be explicitly released by a deallocation mechanism. Such a mechanism not only introduces additional control overhead, but may also fail to release resources completely in case a node previously belonging to the session becomes unreachable. Due to these problems soft state resource reservation mechanisms, which maintain reservations only for small time intervals, are used. These reservations get refreshed if packets belonging to the same flow are received before the timeout period. The soft state reservation timeout period can be equal to packet inter-arrival time or a multiple of the packet inter-arrival time. If no data packets are received for the specified time interval, the resources are deallocated in a decentralized manner without incurring any additional control overhead. Thus no explicit tear down is required for a flow. The hard state schemes reserve resources explicitly and hence at high network loads, the call-blocking ratio will be high, where as soft state schemes provide high call acceptance at a gracefully degraded fashion.
- *Stateful vs stateless approach:* In the stateful approach, each node maintains either *global state* information or only *local state* information, while in the case of stateless approach no such information is maintained at the nodes. State information includes both the topology information and the flow-specific information. If global state information is available, the source node can use a centralized routing algorithm to route packets to the destination. The performance of the routing protocol depends on the accuracy of the global state information

maintained at the nodes. Significant control overhead is incurred in gathering and maintaining global state information. On the other hand, if mobile nodes maintain only local state information (which is more accurate), distributed routing algorithms can be used. Even though control overhead incurred in maintaining local state information is low, care must be taken to obtain loop-free routes. In the case of stateless approach, neither flow-specific nor link-specific state information is maintained at the nodes. Though the stateless approach solves the scalability problem permanently and reduces the burden (storage and computation) on nodes, providing QoS guarantees becomes extremely difficult.

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

4. Classifications of QoS solutions

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

4.1. Classifications of QoS approaches

As shown in Fig. 2 several criteria are used for classifying QoS approaches. The QoS approaches can be classified based on the interaction between the routing protocol and the QoS provisioning mechanism, based on the interaction between the network and the MAC layers, or based on the routing information update mechanism. Based

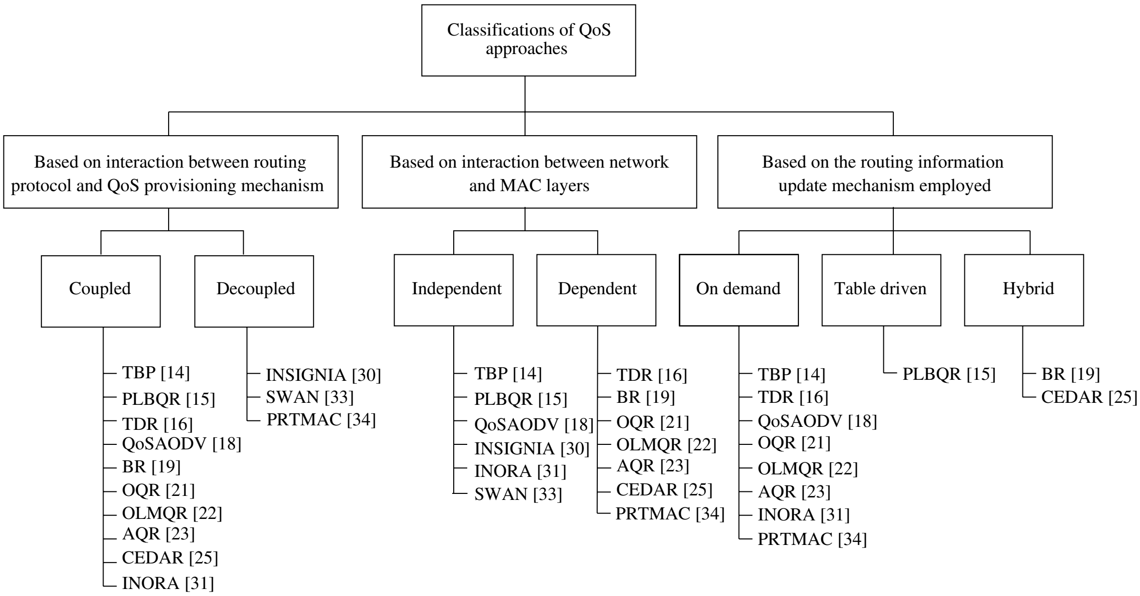


Fig. 2. Classifications of QoS approaches.

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

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

4.2. Layer-wise classification of existing QoS solutions

The existing QoS solutions can also be classified based on which layer in the network protocol stack they operate in. Fig. 3 gives a layer-wise classification of QoS solutions. The figure also shows some of the cross-layer QoS solutions proposed for AWNs. The following sections describe the various QoS solutions listed in Fig. 3.

5. MAC layer solutions

The MAC protocol determines which node should transmit next on the broadcast channel when several nodes are competing for transmission on that channel. Some of the MAC protocols that provide QoS support for applications in AWNs are described below.

5.1. Cluster TDMA

Gerla and Tsai proposed cluster TDMA [5] for supporting real-time traffic in AWNs. In bandwidth-constrained AWNs, the limited resources available need to be managed efficiently. To

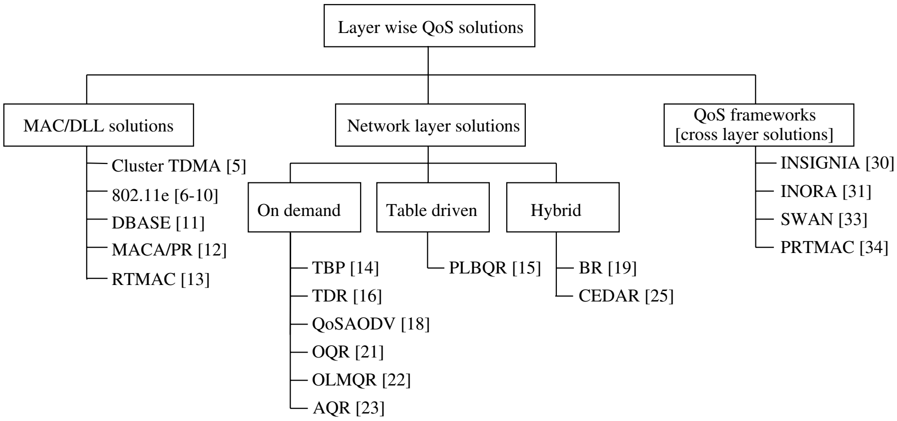


Fig. 3. Layer-wise classification of QoS solutions.

achieve this goal, a dynamic clustering scheme is used in cluster TDMA. In this clustering approach, nodes are split into different groups. Each group has a cluster-head (elected by members of that group), which acts as a regional broadcast node and as a local coordinator to enhance the channel throughput. Every node within a cluster is one hop away from the cluster-head. Formation of clusters and selection of cluster-heads is done in a distributed manner. Clustering algorithms split the nodes into clusters such that they are interconnected and cover all the nodes. Three such algorithms used are, lowest-ID algorithm, highest-degree (degree refers to number of neighbors which are within transmission range of a node) algorithm, and least cluster change (LCC) algorithm. In lowest-ID algorithm, a node becomes a cluster-head if it has the lowest ID among all its neighbors. In the highest-degree algorithm, a node with a degree greater than the degrees of all its neighbors becomes the cluster-head. In LCC algorithm, cluster-head change occurs only if a change in network causes two cluster-heads to come into one cluster or one of the nodes moves out of the range of all the cluster-heads. In each cluster, the corresponding cluster-head maintains a power gain [2] matrix. It contains the power gain lists of all the nodes that belong to a particular cluster. It is useful for controlling the transmission power and the code division within a cluster.

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

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

5.2. IEEE 802.11e

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

5.2.1. IEEE 802.11 MAC protocol

The 802.11 MAC protocol [2] supports two modes of operation, namely distributed coordination function (DCF) and point coordination function (PCF). The DCF mode provides best-effort service, while the PCF mode has been designed to provide real-time traffic support in infrastructure-based wireless network configurations. The DCF mode does not use any kind of centralized control, all stations are allowed to contend for the shared medium simultaneously. CSMA/CA mechanism and random backoff scheme are used to reduce frame collisions.

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

The PCF mode requires an access point (AP i.e., central controller) to coordinate the activity of all nodes in its coverage area. The stations requesting the PCF mode of operation get associated with the PC during the contention period (CP). With PCF, the channel access alternates between the contention free period (CFP) and the contention period (CP) for the PCF and DCF modes of operation, respectively. A CFP and the following CP form a super-frame. The PC generates a beacon frame at regular beacon frame intervals called target beacon transmission time (TBTT). The value of TBTT is announced in the beacon frame. Each super-frame starts with a beacon frame, which is used to maintain synchronization among local timers in the stations and to deliver protocol related parameters. Fig. 4 shows the operation of the network in the combined PCF and DCF modes. The channel access switches alternately between PCF mode and DCF mode, but the CFP may shrink due to stretching when DCF takes more time than expected. This happens when an MSDU is fragmented into several MPDUs, hence giving priority to these fragments over PCF mode of operation.

PCF has certain shortcomings which make it unsuitable for supporting real-time traffic [6]. At TBTT, the PC has to sense the medium idle for at least PIFS before transmitting the beacon frame. If the medium is busy around TBTT, the beacon is delayed, thereby delaying the transmission of real-time traffic that has to be delivered in the following CFP. Further, polled stations' transmission durations are unknown to the PC. The MAC frame (i.e., MSDU) of the polled station may have to be fragmented and may be of arbitrary length. Further, the transmission time of an MSDU is not under the control of the PC because of different modulation and coding schemes specified in the IEEE 802.11 standard. QoS provisioning requires giving some traffic higher priority over other traffic. Such service differentiation is not provided in the DCF mode. Further, the backoff mechanism is uniform for all kinds of traffic. Due to these reasons, several mechanisms have been proposed to enhance the IEEE 802.11 standard to provide QoS support. The QoS mechanisms that are proposed as part of the IEEE 802.11e draft are described below.

5.2.2. QoS support mechanisms of IEEE 802.11e

The IEEE 802.11 Task Group e (TGe) has been setup to enhance the current 802.11 MAC protocol such that it is able to support multimedia applications. The TGe has chosen the virtual DCF (VDCF) [7] proposal as the enhanced DCF (EDCF) access mechanism. EDCF supports real-time traffic by providing differentiated DCF access to the wireless medium. The TGe has also specified a hybrid coordination function (HCF) [8] that combines EDCF with the features of PCF to simplify the QoS provisioning. HCF operates during both the CFP and the CP.

Enhanced distributed coordination function: Enhanced distributed coordination function (EDCF) [7] provides differentiated and distributed access to the wireless medium. Each frame from the higher layer carries its user priority (UP). After receiving each frame, the MAC layer maps it into an access category (AC). Each AC has a different priority of access to the wireless medium. One or more UPs can be assigned to each AC. EDCF channel access

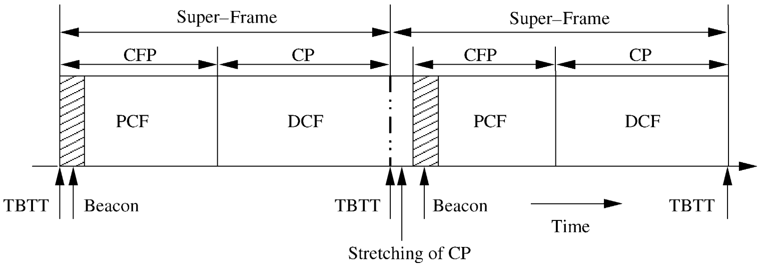


Fig. 4. PCF and DCF frame sharing.

has up to eight ACs [9], to support UPs. EDCF supports eight UPs. Similar to the DCF, each AC has a set of access parameters, such as $CW_{\min}$, $CW_{\max}$, *AIFS*, and transmission opportunity (TXOP) limit. Hence, each AC is an enhanced variant of the DCF. Flows that fall under the same AC are effectively given identical priority to access the channel. A station accesses the channel based on the AC of the frame to be transmitted. An access point that provides QoS is called QoS access point (QAP). Each QAP shall provide at least four ACs. Each station contends for transmission opportunities (TXOPs) using a set of EDCF channel access parameters that are unique to the AC of the packet to be transmitted. The TXOP is defined as an interval of time during which a station has the right to initiate transmissions. It is characterized by a starting time and a maximum duration called TXOPLimit. Depending on the duration of TXOP, a station may transmit one or more MSDUs. Priority of an AC refers to the lowest UP assigned to that AC.

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

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

where *slottime* includes the time needed for a station to detect a frame, the propagation delay, the time needed to switch from the receiving state to the transmitting state, and the time to signal to the MAC layer the state of the channel. *AIFSN*[*i*] is the *AIFS slot count* (i.e., number of time slots a station has to sense the channel as idle before initiating the backoff process) for priority class *i* and takes values greater than zero. For high priority classes, low *AIFSN* values are assigned to give higher priorities for them. After waiting for *AIFS*[*i*], each backoff counter is set to a random integer drawn from the range:

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

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

The reason for having a different range for classes with *AIFSN*[*i*] = 1 is to avoid transmissions initiated by stations that are operating in the EDCF mode from colliding with the hybrid coordinator's (HC, which is explained later in this section) poll packets. The HC operates at QAP and controls QoS basic service set (QBSS) operation under the HCF. Fig. 5 illustrates the relationship between SIFS, PIFS, DIFS, and various AIFS values. As in legacy DCF, if a station detects the channel to be busy before the backoff counter reaches zero,

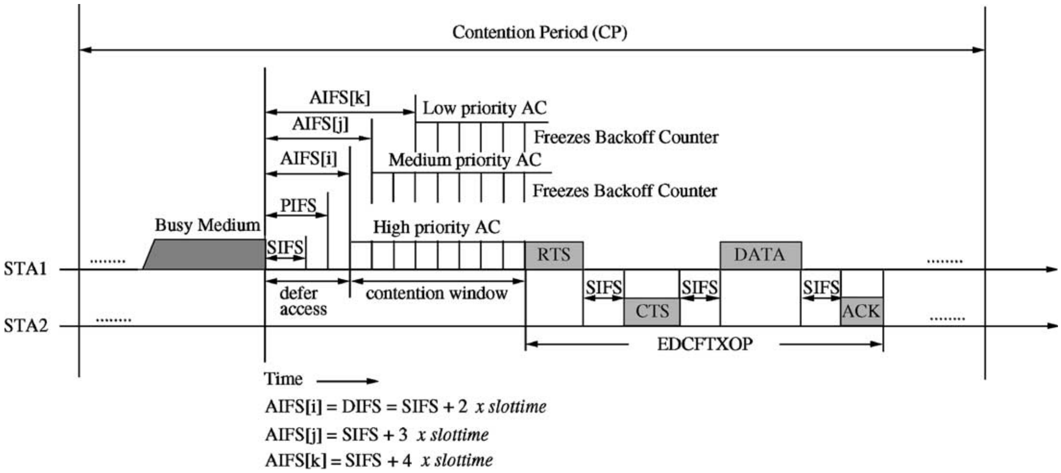


Fig. 5. An example of EDCF access mechanism.

the backoff counter is suspended. The station has to wait for the channel to become idle again for an AIFS period, before continuing to decrement the counter. In this figure, it is assumed that station *STA*1 has traffic that belongs to 3 different ACs. The backoff counter of the highest priority AC expires first, which causes the corresponding AC to seize an EDCF-TXOP for initiating data transmission. The other ACs suspend their backoff counters and wait for the channel to become idle again. When the backoff counter of a particular AC reaches zero, the corresponding station initiates a TXOP and transmits frame(s) that have the highest priority. TXOPs are allocated via contention (EDCF-TXOP) or granted through HCF (polled-TXOP) [6]. The duration of EDCF-TXOP is limited by a QBSS-wide TXOPLimit transmitted in beacons by the HC, while during the CFP the starting time and maximum duration of each polled-TXOP is specified in the corresponding *CF-Poll* frame by the HC. If the backoff counters of two or more ACs in a single station reach zero at the same time, a scheduler inside the station avoids the *virtual collision* by granting the TXOP to the highest priority AC, while low priority ACs behave as if there was an external collision on the wireless medium.

Hybrid coordination function: The Hybrid coordination function (HCF) [8] combines features of EDCF and PCF to provide the capability of selectively handling MAC service data units (MSDUs), in a manner that has upward compatibility with the both DCF and PCF. It uses a common set of frame exchange sequences during both the CP and the CFP. The HCF is usable only in infrastructure-based BSSs that provide QoS, i.e., QBSSs. The HCF uses a QoS-aware point coordinator, called HC, which is typically collocated with a QAP. The HC implements the frame exchange sequences and the MSDU handling rules defined in HCF, operating during both the CP and the CFP. It allocates TXOPs to stations and initiates controlled contention periods for the stations to send reservation requests. When the HC needs access to the wireless medium, it senses the medium. If the medium remains idle for a PIFS period, it initiates MSDU deliveries. The HC can start contention-free controlled access periods (CAPs) at any time during a CP, after the medium is determined to be idle for at least one PIFS period.

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

Fig. 6 illustrates CFP in the HCF mode of operation. During CFP, the HC grants TXOPs to stations by sending QoS *CF-Poll* frames. The polled station can transmit one or more MSDUs in the allocated TXOP. If size of an MSDU is too large, it can be divided into two or more fragments and transmitted sequentially with SIFS waiting periods

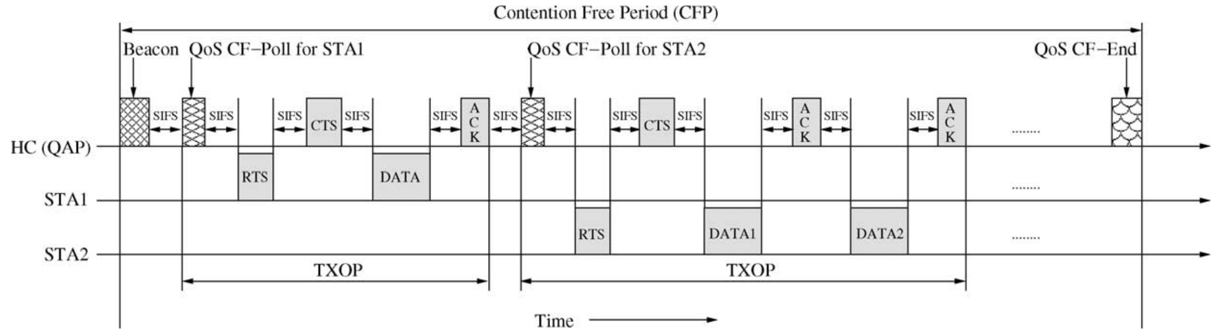


Fig. 6. An example of HCF access mechanism.

in between them. These fragments have to be acknowledged individually. The CFP ends after the time announced in the beacon frame or by a *CF-End* frame from the HC.

5.3. DBASE

The distributed bandwidth allocation/sharing/extension (DBASE) protocol [11] supports multimedia traffic [both variable bit rate (VBR) and constant bit rate (CBR)] over ad hoc WLANs. In an ad hoc WLAN, there is no fixed infrastructure (i.e., AP) to coordinate the activity of individual stations. The stations are part of a single-hop wireless network and contend for the broadcast channel in a distributed manner. For real-time traffic (*rt*-traffic), a contention-based process is used in order to gain access to the channel. Once a station gains channel access, a reservation-based process is used to transmit the subsequent frames. The non-real-time stations (*nrt*-stations) regulate their accesses to the channel according to the standard CSMA/CA protocol used in 802.11 DCF. DBASE is still compliant with the IEEE 802.11 standard.

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

5.3.1. The access procedure for real-time stations

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

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

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

$$\mathrm{RBT} = \mathrm{rand}(c, d) \times \mathit{slottime},$$

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

$$\mathrm{RBTP} = \mathrm{rand}(c + 1, d) \times \mathit{slottime}.$$

If an RF is not received during the interval $(t, t+D_{\max})$, it means that there are no active *rt*-stations. If the channel is still idle in the interval

$(t+D_{\max}+\delta, t+D_{\max}+\delta+\text{PIFS})$ and no RF is detected, the *rt*-station that wants to transmit data at time instant *t*, will execute the backoff scheme. Here δ represents the remaining transmitting time of the current frame at the time instant $t+D_{\max}$. During the backoff process, the *rt*-station should keep monitoring the channel to check whether any *rt*-station has started acting as the CFP Generator. If RBT reaches zero, *rt*-station sends an RTS to the receiver. If no collision occurs, it gets CTS from the receiver and acts as CFPG. If a collision occurs, the *P*-persistent scheme as mentioned above is used to decide on when the stations are to transmit again.

The bandwidth reservation scheme is illustrated in Fig. 7. Fig. 7(a) depicts a case in which no collision occurs, while Fig. 7(b) shows a scenario in which a collision occurs. In Fig. 7(a), stations *A* and *C* have *rt*-frames for transmission to stations *B* and *D*, respectively. Besides these, station *E* has *nrt*-frames to be transmitted to station *D*. After listening to the channel for $D_{\max}$ time period in order to detect the presence of an RF, stations *A* and *C* conclude that no CFPG exists in the network. Then, if they find the channel as being idle for a PIFS period, they initiate their backoff timers. In this case, assume that RBT_A is one slot and RBT_C is three slots. During the backoff process, once the channel becomes busy, the backoff timer of station *C* is paused as shown in Fig. 7(a). It is restarted from the same value once the channel becomes idle again. After RBT_A counts down to zero, station *A* seizes the channel and sends an RTS. If no collision occurs, station *A* receives a CTS within SIFS time duration. Then station *A* records its reservation information into the RSVT and becomes the CFPG. Since station *A* is currently playing the role of CFPG, it transmits an RF before transmitting its first *rt*-frame. Once station *A* completes its transmission, station *C* continues its backoff process. When RBT_C counts down to zero, station *C* reserves bandwidth by

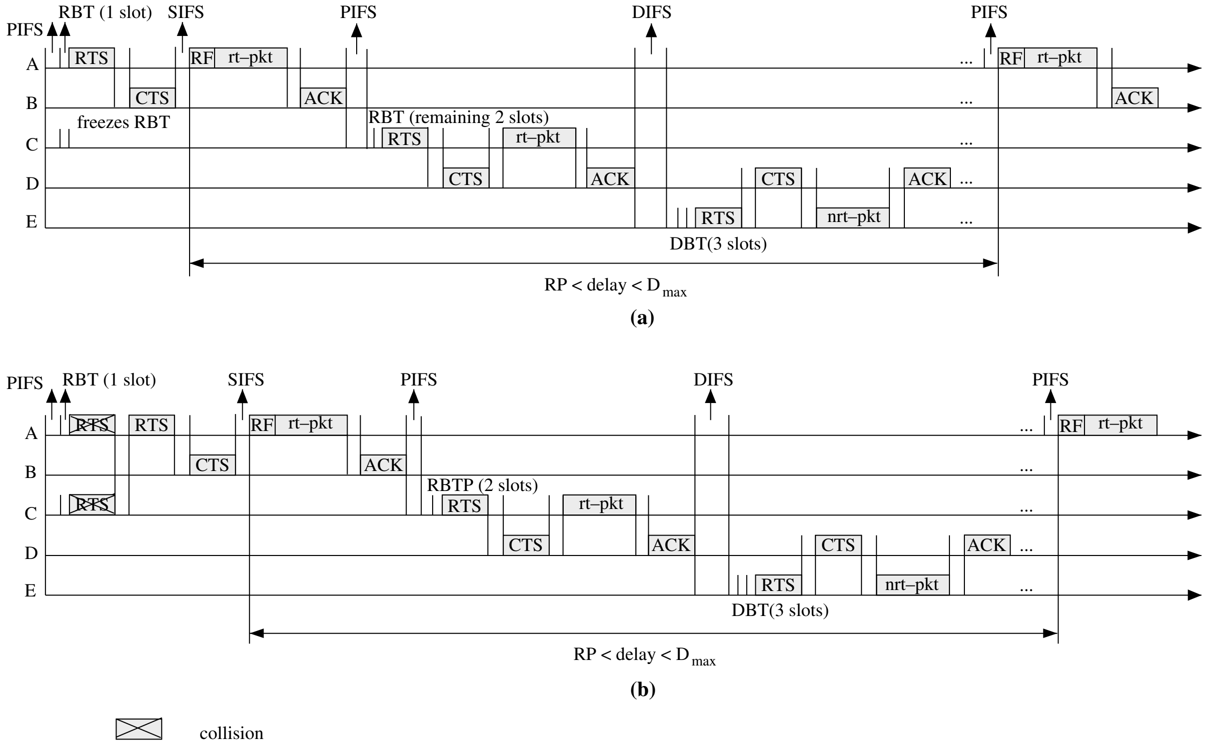


Fig. 7. An example of new *rt*-stations joining the network: (a) without collision and (b) with collision.

adding a corresponding entry into the RSVT and transmits its first *rt*-frame. When station E detects the channel as being idle for DIFS, it implies that no other *rt*-station wants to transmit currently, and hence station E sends its RTS as soon as DBT_E counts down to zero. By the end of a contention period whose length is limited by a parameter $\mathrm{RP}_{\max}$ (maximum repetition period), bandwidth would be reserved for the *rt*-stations, and thereafter they need not exchange RTS/CTS control frames before transmitting their *rt*-frames. The delay between two RFs varies from real-time period (RP) to $D_{\max}$, where RP is sum of the CFP (*rt*-stations reserved period) and the CP for new *rt*-stations.

In Fig. 7(b), assume that both station A and station C generate RBT as one slot. After waiting for one time slot, both transmit their RTS frames, which results in a collision. Then the P-persistent scheme is applied. Assume that station A gets access to the channel during the next slot itself, but station C does not. Then, station A will retransmit its RTS in the following slot, while station C initiates a new backoff time RBTP_C. If no collision occurs, station A gets a CTS within SIFS, and sends out an RF and its *rt*-frame. When RBTP_C counts down to zero, station C seizes the channel to send an RTS. If any collision occurs, the *rt*-station uses the P-persistent scheme to resolve the collision. The collision resolution process is restricted from crossing the $\mathrm{RP}_{\max}$ boundary.

The MAC layer solutions such as MACA/PR [12] and RTMAC [13] provide real-time traffic support in asynchronous AWNs. One advantage of these solutions is their asynchronous mode of operation where nodes do not require any global time synchronization. Another advantage 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 MACA/PR, which requires the free slot to accommodate 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 MACA/PR.

6. Network layer solutions

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

6.1. QoS routing protocols

QoS routing protocols search for routes with sufficient resources in order to satisfy the QoS requirements of a flow. The information regarding the availability of resources is managed by a resource management module which assists the QoS routing protocol in its search for QoS feasible paths. The QoS routing protocol should find paths that consume minimum resources. The QoS metrics can be classified as additive metrics, concave metrics, and multiplicative metrics.

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

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

6.2. Ticket-based QoS routing protocol

Ticket-based QoS routing [14] is a distributed QoS routing protocol for AWNs. It can tolerate imprecise state information during QoS route computation and exhibits good performance even when the degree of imprecision is high.

6.2.1. Protocol overview

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

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

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

6.2.2. Optimizing cost of a feasible path

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

6.2.3. Advantages and disadvantages

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

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

6.3. Predictive location based QoS routing protocol

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

6.3.1. Location and delay predictions

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

Location prediction: Let (x_1, y_1) at t_1 and (x_2, y_2) at t_2 $(t_2 > t_1)$ be the latest two updates from the destination D to the source node S. Assume that the second update message also indicates v, which is the velocity of D at (x_2, y_2). Assume that node S wishes to predict the location (x_f, y_f) of node D at some instant t_f in the future. This situation is depicted in Fig. 8. The value of t_f has to be estimated first using the delay prediction scheme, which would be explained later in this section. From Fig. 8, using similarity of triangles, the following equation is obtained:

$$\frac{y_2 - y_1}{y_f - y_1} = \frac{x_2 - x_1}{x_f - x_1}. \tag{1}$$

By solving the above equation for y_f,

$$y_f = y_1 + \frac{(x_f - x_1)(y_2 - y_1)}{x_2 - x_1}. \tag{2}$$

Using the above Eq. (2), source S can calculate y_f if it knows x_f, which in turn can be calculated as follows. Using similarity of triangles again, the following equation is obtained:

$$y_f - y_2 = \frac{(y_2 - y_1)(x_f - x_2)}{x_2 - x_1}. \tag{3}$$

By using Pythagoras theorem,

$$(x_f - x_2)^2 + (y_f - y_2)^2 = v^2(t_f - t_2)^2. \tag{4}$$

Substituting for $y_f - y_2$ from Eq. (3) in the above Eq. (4) and solving for x_f, the following equation is obtained:

$$x_f = x_2 + \frac{v(t_f - t_1)(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}. \tag{5}$$

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

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

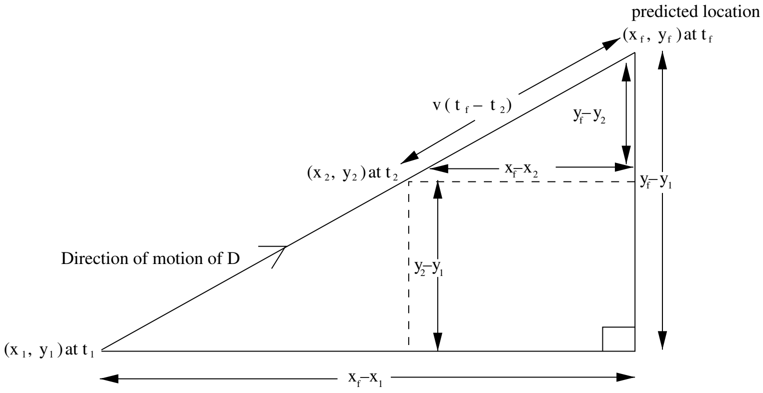


Fig. 8. Prediction of location at a future time by node S using last two updates.

6.3.2. QoS routing

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

The routing algorithm given in [15] works as follows. The source node S first runs location and delay predictions on each node in its proximity list in order to obtain a list of its neighbors at present. It determines which of these neighbors have the resources to satisfy the QoS requirements of the connection (the neighbors that satisfy the QoS requirements are called candidates). Then it performs a depth-first search for the destination starting with each of these candidate neighbors to find all candidate routes satisfying the QoS requirements of the connection request. From the resulting candidate routes, the geographically shortest route is chosen and the connection is established. Data packets are forwarded along this chosen route until the end of the connection or until the route is recomputed in anticipation of breakage. Note that node S only uses its view of the network for the entire computation.

6.3.3. Advantages and disadvantages

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

6.4. Trigger based distributed QoS routing protocol

The trigger-based (on-demand) distributed QoS routing (TDR) protocol [16] was proposed by De et al. for supporting real-time applications in AWNs. Every node maintains only the local neighborhood information in order to reduce computation overhead and storage overhead. For each neighbor, every node maintains *received power level*, current geographic coordinates, velocity, and direction of motion. To reduce control overhead, nodes maintain only the active routes.

In addition to the local neighborhood information, node N maintains a source table ST_N, a destination table DT_N, or an intermediate table

IT_N based on whether it actively participates in a session as the source (*S*), the destination (*D*), or as an intermediate node (*I*), respectively. At any time instant, a node may have to maintain one or more tables simultaneously for different ongoing sessions. Each node *N* also maintains an updated residual bandwidth ($ResiBW_N$) which indicates its ability to participate in a session. A soft state approach is used to maintain these tables.

6.4.1. Routing protocol

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

Initial route discovery: If the source *S* has enough $ResiBW_S$ to satisfy the MaxBW (maximum bandwidth) for the session, the required bandwidth is temporarily reserved for a certain duration within which it expects an acknowledgment from the destination *D*. If the source knows the location of the destination, it performs route discovery through selective forwarding. In this approach, the source node takes advantage of location information of its neighbors and forwards route requests to only selective neighbors that are lying closely towards the destination node and satisfying QoS requirements of the connection request. Otherwise, the source initiates a flooding-based initial route discovery process. Before transmitting the route discovery packet, an entry is made in the source table ST_S for this session with NodActv flag (activity flag) set to zero (i.e., idle). To ensure stability of routes and in order to reduce the control overhead, only selected neighbors, from which packets were received with power level more than a threshold level (P_{th1}), are considered during route establishment. After receiving a route discovery packet, the intermediate node (IN) increments the hop count field of that packet by one and checks for $ResiBW_{\text{IN}}$. If it can meet the MaxBW requirement and if the updated hop count field is less than MaxDelay (maximum delay), the required bandwidth is temporarily reserved and an entry is made into the activity table IT_{IN} for the session with NodActv flag set to zero. Then the packet is forwarded to its downstream neighbors. If either or both of *ResiBW* and MaxDelay criteria cannot be satisfied, the discovery packet is simply dropped. Upon receiving the first discovery packet, if the destination *D* is also able to satisfy both the *ResiBW* and the MaxDelay criteria, it builds DT_D table with the NodActv flag set to 1 (i.e., active) and sends an ACK to the source *S* along the selected route. On receiving the ACK packet, all intermediate nodes and the source *S* set the NodActv flags in their respective tables to 1 and refresh their *ResiBW* status. The packet transmission for the session follows immediately.

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

6.4.2. Advantages and disadvantages

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

6.5. QoS enabled ad hoc on-demand distance vector routing protocol

Perkins et al. have extended the basic ad hoc on-demand distance vector (AODV) routing protocol [17] to provide QoS support in AWNs [18]. To provide QoS, packet formats have been modified in order to specify the service requirements which must be met by the nodes forwarding a route request (RREQ) or a route reply (RREP).

6.5.1. QoS extensions to AODV protocol

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

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

6.5.2. Maximum delay extension field

The maximum delay extension field is interpreted differently for RREQ and RREP messages. In a RREQ message it indicates the maximum time (in seconds) allowed for a transmission from the current node to the destination node. In a RREP message, it indicates the current estimate of cumulative delay from the current intermediate node forwarding the RREP, to the destination. Using this field the source node finds a path (if it exists) to the destination node satisfying the maximum delay constraint. Before forwarding the RREQ, an intermediate node compares its NODE TRAVERSAL TIME (i.e., the time it takes for a node to process a packet) with the (remaining) delay indicated in the maximum delay extension field. If the delay is less than NODE TRAVERSAL TIME, the node discards the RREQ packet. Otherwise, the node subtracts NODE TRAVERSAL TIME from the delay value in the extension and processes the RREQ as specified in the AODV protocol.

The destination node returns a RREP with the maximum delay extension field set to zero. Each intermediate node forwarding the RREP adds its own NODE TRAVERSAL TIME to the delay field and forwards the RREP towards the source. Before forwarding the RREP packet the intermediate node records this delay value in the routing table entry for the corresponding destination node.

Similarly, a minimum bandwidth extension field is also proposed to find a path (if it exists) to the destination node satisfying the minimum bandwidth constraint. A QOS LOST message is generated when an intermediate node experiences an increase in NODE TRAVERSAL TIME or a decrease in the link capacity. The QOS LOST message is forwarded to all sources potentially affected by the change in the QoS parameter.

6.5.3. Advantages and disadvantages

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

6.6. Bandwidth routing protocol

The bandwidth routing (BR) protocol [19] consists of an end-to-end path bandwidth calculation algorithm to inform the source node of the available bandwidth to any destination in the ad hoc network, a bandwidth reservation algorithm to reserve sufficient number of free slots for the QoS flow, and a standby routing algorithm to re-establish the QoS flow in case of path breaks.

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

6.6.1. Bandwidth calculation

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

The BR protocol uses a heuristic-based hop-by-hop path bandwidth calculation algorithm to assign free slots at every hop along the path. The algorithm is explained with the help of the example shown in Fig. 9, where a path from source node S to destination node D is illustrated. The process of computing pathBW(S,D) is explained below.

- pathBW(S,A): Since node S and node A are adjacent, the pathBW(S,A) = linkBW(A,S), which is four slots. The four slots are $\{2,5,6,7\}$.
- pathBW(S,B): Since pathBW(S,A) = linkBW(A,B) = $\{2,5,6,7\}$, if S uses slots 6 and 7 to send packets to A, then A can only use slots

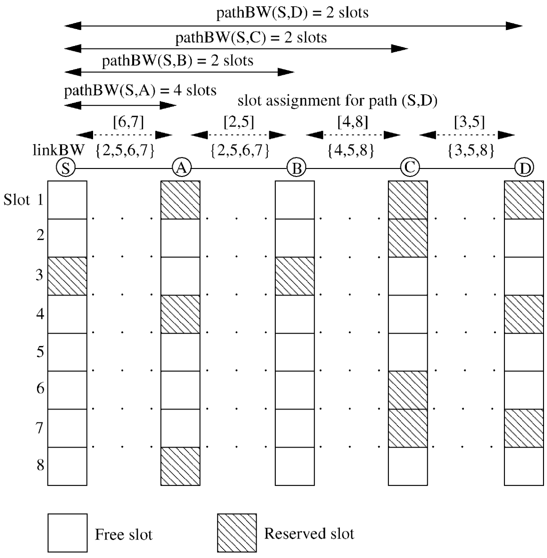


Fig. 9. An example of path bandwidth calculation in BR protocol.

2 and 5 for transmission of packets to B. This is because a node cannot be in transmission and reception modes simultaneously. Hence pathBW(S, B) is 2 slots, by assigning slots $\{6, 7\}$ on link(S, A) and slots $\{2, 5\}$ on link(A, B).

- pathBW(S, C): Here slots 4 and 8 are exclusively available for linkBW(B, C), slot 2 is exclusively available for pathBW(S, B), and slot 5 is common for both of them. So assign one of slots 4, 8 to link(B, C), for example assign slot 4 to link(B, C), and slot 2 to path(S, B). For achieving maximum bandwidth assign slot 8 to link(B, C) and slot 5 to path(S, B). Hence pathBW(S, C) is 2 slots, by assigning slots $\{6, 7\}$ on link(S, A), slots $\{2, 5\}$ on link(A, B), and slots $\{4, 8\}$ on link(B, C).
- pathBW(S, D): This case is similar to previous one. So slots 4 and 8 are assigned to path(S, C) and slots 3 and 5 are assigned to link(C, D) to get 2 slots for pathBW(S, D).

6.6.2. Slot assignment

The slot assignment algorithm in each node assigns free slots during the call setup. When a node receives a call setup packet, it checks whether the slots that the immediate sender will use for transmission are free, and it also finds if there are free slots that can be used for forwarding the incoming packets. If such free slots are available, the slot assignment algorithm reserves the required number of slots, updates the routing table, and then forwards the call setup packet to the next hop. Otherwise, all the reservations that have been made so far along the path have to be cancelled by sending a RESET packet back to the source along that path. If reservations are made successfully along the path from the source to the destination, the destination sends a REPLY packet back to the source to acknowledge having set up the connection. The reservations are soft state in nature in order to avoid resource lock-up at intermediate nodes due to path breaks.

6.6.3. Standby routing mechanism

The standby routing mechanism has to reestablish connections that are broken due to mobility of nodes. The standby route is easily computed using the DSDV algorithm [20] without any extra overhead. Each node periodically exchanges routing information with its neighboring nodes. The neighbor with the shortest distance to the destination node becomes the next node on the primary path to the destination node. The neighbor node with the second shortest distance to the destination becomes the next node on the standby route to the destination. It is to be noted that this standby route is not guaranteed to be a link or node disjoint one. When a primary path fails, the upstream node that detects the link break will try to rebuild a new path immediately using the standby route. If the standby route satisfies the QoS requirements, the new path from the point of path break is established by sending a call setup packet hop-by-hop to the destination through the standby path.

Since this scheme follows DSDV protocol, a table-driven routing protocol, and uses on-demand call admission control, similar to the on-demand routing protocols, it is classified into the category of hybrid solutions in the classifications Fig. 2.

6.6.4. Advantages and disadvantages

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

6.7. On-demand QoS routing protocol

Lin proposed an admission control scheme over an on-demand QoS routing (OQR) protocol [21] to guarantee bandwidth for real-time applications. Since routing is on-demand in nature there is no need to exchange control information periodically and maintain routing tables at each node. Similar to the bandwidth routing (BR) protocol, the network is time-slotted and bandwidth is the key QoS parameter. The path bandwidth calculation algorithm proposed in BR is used to measure the available end-to-end bandwidth. The on-demand QoS routing protocol is explained below.

6.7.1. Route discovery

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

1. If a QRREQ with the same {source ID, sequence number} had been received already, this one gets discarded.
2. Otherwise, *route list* field is checked for the address of *N*. If it is present, node *N* discards this QRREQ packet.
3. Otherwise,

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

For the example shown in Fig. 9, assume that the source *S* floods a QRREQ packet with bandwidth requirement of 2 time slots. Here, the destination *D* receives a QRREQ packet with the following information in its fields. The *route list* field contains (S, A, B, C) and the *slot array list* contains $([A, \{2, 5, 6, 7\}]$, $[B, \{2, 5\}]$, $[C, \{4, 5\}]$, $[D, \{3, 8\}])$.

6.7.2. Bandwidth reservation

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

6.7.3. Advantages and disadvantages

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

6.8. On-demand link-state multi-path QoS routing protocol

Unlike previous QoS routing protocols described in this paper which try to find a single path from the source to the destination satisfying the QoS requirements, the on-demand link-state multi-path QoS routing (OLMQR) protocol [22] searches for multiple paths which collectively satisfy the required QoS. The original bandwidth requirement is split into sub-bandwidth requirements. Notably, the paths found by the multi-path routing protocol are allowed to share the same sub-paths. OLMQR has better call acceptance rate in AWNs where finding a single path satisfying all the QoS requirements is very difficult.

In this protocol, the MAC layer is assumed to be using the CDMA-over-TDMA channel model similar to BR and OQR protocols. A mobile node in the network knows the available bandwidth to each of its neighbors. The operation of this protocol consists of three phases. Phase 1 is on-demand link-state discovery, phase 2 is uni-path discovery, and phase 3 is multi-path discovery and reply.

6.8.1. On-demand link-state discovery

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

1. Node N checks *node history* field of the QRREQ packet for its address. If it is present, the node discards this QRREQ packet.
2. Otherwise,
    - Node N decrements TTL by one. If TTL counts down to zero, it discards this QRREQ packet.
    - Node N adds itself into *node history* field, appends the free time slots of the link between itself and the last node recorded in the *node history* field into the *free time-slot list* field, and re-broadcasts this QRREQ packet.

The destination may receive many different QRREQ packets from the source. It constructs its own view of the current network topology. It also calculates the available bandwidths of the links present in that network topology. For example, consider the network shown in Fig. 10. The source S floods the network with a QRREQ packet by setting BW and TTL fields to 3 and 4, respectively. The destination D receives six QRREQ packets, which have traversed along the paths: $S \rightarrow A \rightarrow B \rightarrow D$, $S \rightarrow E \rightarrow F \rightarrow D$, $S \rightarrow A \rightarrow C \rightarrow B \rightarrow D$, $S \rightarrow A \rightarrow C \rightarrow F \rightarrow D$, $S \rightarrow E \rightarrow C \rightarrow F \rightarrow D$, and $S \rightarrow E \rightarrow C \rightarrow B \rightarrow D$. Using this information, a partial view of the network is constructed at the destination D.

6.8.2. Uni-path discovery

Unlike the BR [19] and the OQR [21] protocols discussed earlier in this section, here the uni-path discovery operation (i.e., path bandwidth calculation algorithm) does not follow the traditional hop-by-hop approach to determine the end-to-end path bandwidth. The uni-path discovery approach acquires higher end-to-end path bandwidth than that acquired through the hop-by-hop

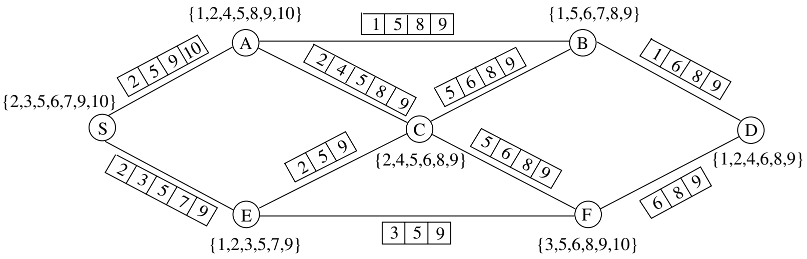


Fig. 10. An example network.

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

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

For example, consider the path $S \rightarrow A \rightarrow B \rightarrow D$ from the source S to the destination D in the network shown in Fig. 10. Let a, b, c denote free time slots of links (S,A), (A,B), and (B,D), respectively as shown in Fig. 11(a). For this path, a time slot reservation tree T can be constructed as shown in Fig. 11(b). It shows two reservation patterns, the first pattern is $\underline{ab}$, $\underline{c}$ and the second pattern is $\underline{bc}$, $\underline{a}$. In the first pattern, $\underline{ab}$ has 3 time slots bandwidth (by assigning slots 2, 5, and 10 for the link a and slots 1, 8, and 9 for the link b) and $\underline{c}$ has 1 time slot bandwidth (by assigning the remaining slot 6 for the link c). Hence, the first pattern $\underline{ab}$, $\underline{c}$ has 1 time slot path bandwidth (which is the minimum of bandwidths of $\underline{ab}$ and $\underline{c}$). Similarly in the second pattern, $\underline{bc}$ has 2 time slots bandwidth (by assigning slots 1 and 5 for the link b and slots 6 and 8 for the link c) and $\underline{a}$ has 3 time slots bandwidth (by assigning the remaining slots 2, 9, and 10 for the link a). Hence, the second pattern $\underline{bc}$, $\underline{a}$ has 2 time slots path bandwidth. From T, a least-cost-first time slot reservation tree T_{LCF} can be constructed as shown in Fig. 11(c). Comparing T-tree traversal with T_{LCF}-tree traversal scheme, the T_{LCF}-tree traversal scheme is more efficient than the T-tree traversal scheme as it reduces the time required to find a feasible QoS path.

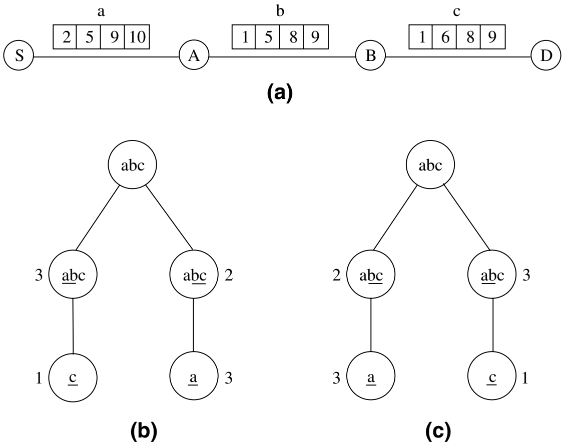


Fig. 11. Example of T and T_{LCF} trees for a path.

6.8.3. Multi-path discovery and reply

The destination initiates the multi-path discovery operation by sequentially exploiting multiple uni-paths such that the sum of path bandwidths fulfills the original bandwidth requirement BW. The destination applies the uni-path discovery operation to each path in order to determine the maximum achievable path bandwidth of each path. After accepting a path, the destination updates the network state information it maintains in order to reflect the current bandwidth availability on the links. Finally, the destination sends reply packets along these paths, which reserve the corresponding resources (sub-bandwidth requirements) on the corresponding paths on their way back to the source. In the above example, the destination D finds two uni-paths: $S \rightarrow A \rightarrow B \rightarrow D$ with two time slots path bandwidth and $S \rightarrow E \rightarrow F \rightarrow D$ with one time slot path bandwidth as shown in Fig. 12.

6.8.4. Advantages and disadvantages

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

6.9. Asynchronous slot allocation strategies

The QoS solutions discussed so far such as BR, OQR, and OLMQR assume a TDMA based network or a CDMA-over-TDMA model for the network. This requires time synchronization across all nodes in the network. Time synchronization demands periodic exchange of control packets, that results in high bandwidth consumption. AWNs experience rapid changes in topology leading to a situation where network partitions and merging of partitions can take place. Fig. 13 shows the synchronization problems arising out of dynamic topological changes in an ad hoc wireless network. A completely connected and synchronized network A at time $t = t_0$ (shown in Fig. 13(a)) may be partitioned into two disjoint networks A1 and A2 at time $t = t_1$ (shown in Fig. 13(b)). These two networks may be synchronized to two different clock times as illustrated. Due to the dynamic topology experienced in an ad hoc wireless network, it is possible to have two separately synchronized networks A1 (synchronized to t_{A1}) and A2 (synchronized to t_{A2}) merge to form a combined network A (Fig. 13(c)). During the merging process, the real-time calls existing in the network may be affected while accommodating the changes in synchronization.

The asynchronous QoS routing (AQR) scheme and slot allocation strategies proposed in [23]

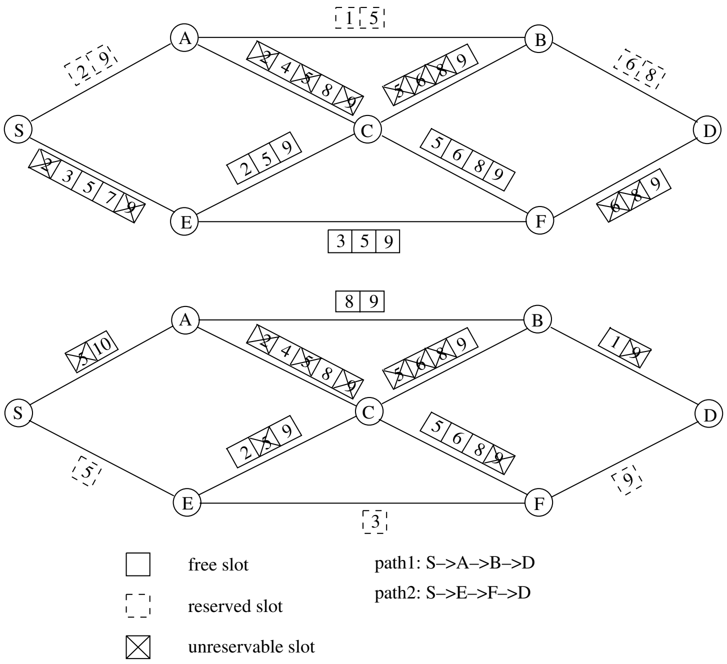


Fig. 12. The uni-paths found by multi-path discovery algorithm.

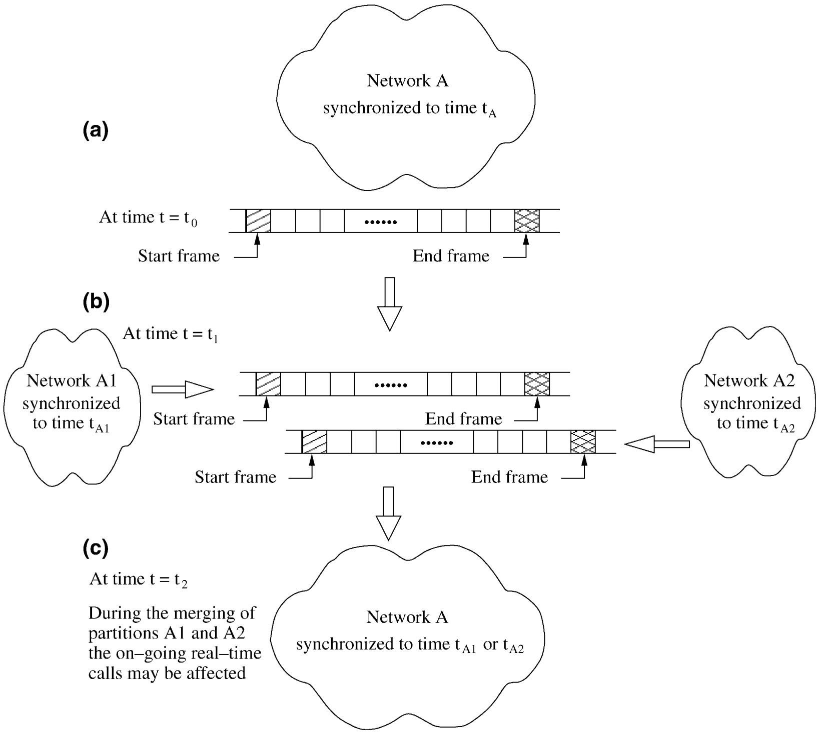


Fig. 13. Illustration of synchronization problems in a dynamic network topology.

provides a unique mechanism to reserve asynchronous end-to-end bandwidth for real-time calls in AWNs. These strategies utilize the real-time MAC (RTMAC) [13] protocol that can effect bandwidth reservation in asynchronous AWNs. RTMAC can reserve conn-slots [number of reservation slots (minimum time duration that can be reserved) sufficient for a real-time session] on a super-frame (time duration in which the existing reservations repeat). AQR is an extension of dynamic source routing (DSR) protocol [24]. The three major phases in the operation of AQR are bandwidth feasibility test phase, bandwidth allocation phase, and bandwidth reservation phase. An in-depth discussion of each of these phases is provided in what follows.

6.9.1. Bandwidth feasibility test phase

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

The destination node waits for a specific time interval and gathers a set of RREQs and chooses a shortest path with necessary bandwidth.

6.9.2. Bandwidth allocation phase

In this phase, the destination node performs a bandwidth allocation strategy that assigns free slots to every intermediate link in the chosen path. The information about asynchronous slots assigned at every intermediate link is included in the route reply (RREP) packet and propagated through the selected path back to the source. Slot allocation strategies such as early fit reservation (EFR), minimum bandwidth-based reservation (MBR), position-based hybrid reservation (PHR), and *k*-hopcount hybrid reservation (*k*-HHR) are used for allocation of bandwidth and positioning of slots. The order of links in which it is chosen for allocation and the position of assigned bandwidth slots influence the end-to-end delay of the path and the call acceptance rate. We discuss MBR allocation scheme alone here.

Minimum bandwidth-based reservation (MBR): The following steps are executed by the destination node for the MBR scheme:

- Step 1: Order the links in the non-decreasing order of free bandwidth.
- Step 2: Allocate the first free slot in the link with lowest free bandwidth.
- Step 3: Reorder the links in the non-decreasing order of free bandwidth and assign the first free slot on the link with lowest bandwidth.

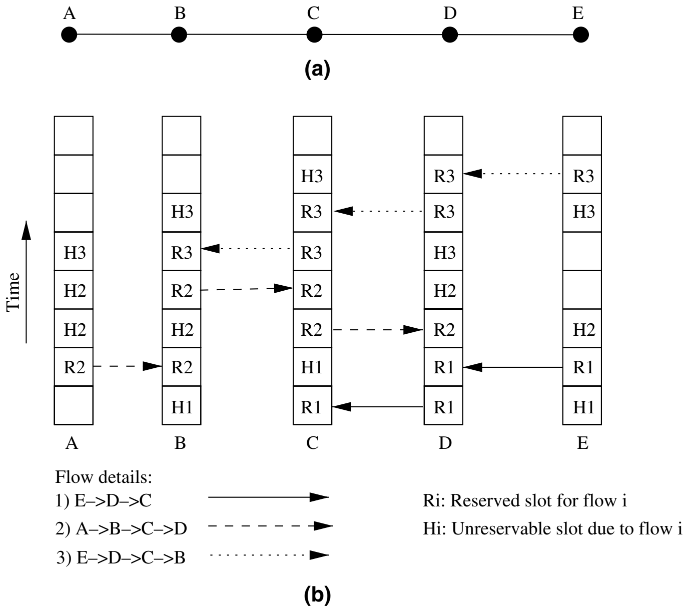


Fig. 14. Illustration of MBR scheme: (a) example network topology and (b) MBR scheme.

- Step 4: Continue Step 3 until bandwidth is allotted for all the links.

Fig. 14(b) shows the slot allocation carried out in MBR scheme over a simple string topology network. The worst case end-to-end delay provided by MBR can be $(n-1)\times t_{\mathrm{sf}}$ where n is the number of hops in the path and t_{sf} is the duration of super-frame. In the example in Fig. 14(b), the average delay experienced can be calculated as 33/3 slots.

6.9.3. Bandwidth reservation phase

The RREP packet traverses along the path chosen by the destination node. Upon receiving the RREP, the intermediate node checks the status of conn-slot at which reservation is to be carried out. If it is free, the RREP packet is further forwarded. Otherwise, the intermediate node tries to reserve any of the free slots available. If free slots are not available, it drops the RREP and sends a control packet to the destination. The control packet makes all the intermediate nodes in its way to release the bandwidth reserved for the session and the destination node to find another path with the necessary bandwidth.

6.9.4. Advantages and disadvantages

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

CEDAR: Core extraction distributed ad hoc routing (CEDAR) [25] integrates routing and support for QoS. Route establishment in CEDAR is carried out in two phases. The first phase finds a core path from the source to the destination. In the second phase, a QoS feasible path is found over the core path. 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, movement of core nodes adversely affects the performance of the protocol. Also, the core node update information could cause a significant amount of control overhead.

7. QoS frameworks for ad hoc wireless networks

A framework for QoS is a complete system that attempts to provide required/promised services to each user or application. All components within this system cooperate together in providing the required services. The key component of any QoS framework is the QoS model which defines the way user requirements are met. The key design issue here is whether to serve users on a per session basis or on a per class basis. Each class represents an aggregation of users based on certain criteria. The other key components of the framework are, QoS routing which is used to find all or some of the feasible paths in the network that can satisfy user requirements, QoS signaling for resource reservation, QoS medium access control, call admission control, and packet scheduling schemes. The QoS modules should react promptly to changes in the network state (topology changes) and flow state (change in the end-to-end view of the service delivered). In what follows, each component's functionality and its role in providing QoS in AWNs will be described:

- *Routing protocol:* The routing protocol is used to find a path from the source to the destination and to forward the data packet to next intermediate relay node. The routing protocol needs to work efficiently with other components of the QoS framework in order to provide end-to-end QoS guarantees. These mechanisms should consume minimal resources in operation and react rapidly to changes in the network state and flow state.

- *QoS resource reservation signaling:* Once a QoS path is found, the resource reservation signaling protocol reserves the required resources along that path. For example, for applications that require certain minimum bandwidth guarantees, signaling protocol communicates with the MAC subsystem to find and reserve the required bandwidth. On completion/termination of a session, the previously reserved resources are released.
- *Admission control:* Even though a QoS feasible path may be available, the system needs to decide whether to actually serve the connection or not. If the call is to be served, the signaling protocol reserves the resources, otherwise the application is notified of the rejection. When a new call is accepted, it should not jeopardize the QoS guarantees given to the already admitted calls. A QoS framework is evaluated based on the number of QoS sessions it serves and it is represented by ACAR metric. Admission control ensures that there is no perceivable degradation in the QoS being offered to the QoS sessions admitted already.
- *Packet scheduling:* When multiple QoS connections are active at the same time through a link, the decision on which QoS flow is to be served next is made by the scheduling scheme. For example, when multiple delay-constrained sessions are passing through a node, this module decides on when to schedule the transmission of packets, when packets belonging to more than one session are pending in the transmission queue of the node. The performance of a scheduling scheme is reflected by the percentage of packets that meet their deadlines.

7.1. QoS models

A QoS model defines the nature of service differentiation. In wired network QoS frameworks, several service models have been proposed. Two of these models are, integrated services (IntServ) model [26] and differentiated services (DiffServ) model [27]. The IntServ model provides QoS on a per flow basis. The volume of information maintained at an IntServ-enabled router is proportional to the number of flows. Hence, the IntServ model is not scalable for the Internet, but it can be applied to small sized AWNs. But, per flow information is difficult to maintain precisely at a node in an ad hoc wireless network. The DiffServ model was proposed in order to solve the scalability problem faced by IntServ model. In this model, flows are aggregated into limited number of service classes. Each flow belongs to one of the DiffServ classes of service.

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

7.1.1. Flexible QoS model for mobile ad hoc networks

The flexible QoS model for mobile ad hoc networks (FQMM) [29] takes advantage of the per flow granularity of IntServ and aggregation of services into classes in DiffServ.

A source node, which is the originator of the traffic, is responsible for traffic shaping. Traffic shaping is the process of delaying packets belonging to a flow so that packets conform to a certain defined traffic profile. Traffic profile contains a description of the temporal properties of a flow such as its mean rate (i.e., rate at which data can be sent per unit time on average) and burst size (which specifies in bits per burst how much traffic can be sent within a given unit of time without creating scheduling concerns). FQMM model provides per flow QoS guarantees for the high priority flows while lower priority flows are aggregated into a set of service classes as illustrated in Fig. 15. This hybrid QoS model is based on the assumption that the percentage of flows requiring per flow QoS guarantees is much less than that of low priority flows which can be aggregated into a set of QoS classes. Based on the current traffic load in the network, service level of a flow may change dynamically from per flow to per class and vice versa.

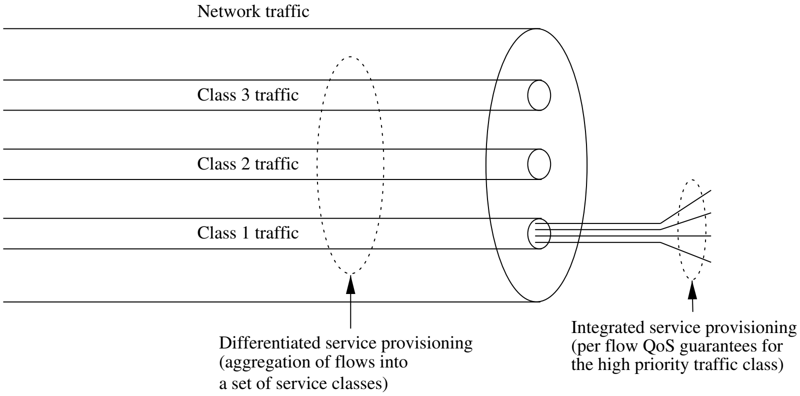

Network traffic
Class 3 traffic
Class 2 traffic
Class 1 traffic
Differentiated service provisioning
(aggregation of flows into
a set of service classes)
Integrated service provisioning
(per flow QoS guarantees for
the high priority traffic class)

Fig. 15. FQMM model.

Advantages and disadvantages: FQMM provides the ideal per flow QoS guarantees and overcomes the scalability problem by classifying the low priority traffic into service classes. This protocol addresses the basic problem faced by QoS frameworks and proposes a generic solution for AWNs that can be a base for a better QoS model. But several issues still remain unresolved, such as decision upon traffic classification, allotment of per flow or aggregated service for the given flow, amount of traffic belonging to per flow service, the mechanisms used by the intermediate nodes to get information regarding the flow, and scheduling or forwarding of the traffic by the intermediate nodes.

7.2. QoS resource reservation signaling

The QoS resource reservation signaling scheme is responsible for reserving the required resources and informing the corresponding applications, which then initiate data transmission. Signaling protocol consists of three phases, viz., connection establishment, connection maintenance, and connection termination. On establishing a connection, it monitors the path and repairs/reconfigures it if the connection suffers from any violation in its QoS guarantees. On completion/termination of a session, it releases the resources that had been reserved for that session. In the wired networks, the RSVP protocol [28] is used for resource reservation but it cannot be applied directly to AWNs due to the following reasons:

- The amount of control overhead generated during the connection maintenance phase of RSVP signaling is too heavy for bandwidth constrained AWNs.
- It is not adaptive to network dynamics. In wired networks, once the resources are reserved, they are assumed to be available to applications throughout the session. But these assumptions are not true in AWNs due to unrestricted mobility of nodes which results in dynamic changes in the network topology.

7.3. INSIGNIA

The INSIGNIA QoS framework [30] was developed for providing adaptive services in AWNs. Adaptive services support applications that require only a minimum quantitative QoS guarantee (such as minimum bandwidth) called *base QoS*. The service level can be extended later to *enhanced QoS* when sufficient resources become available. Here user sessions adapt to the available level of service without explicit signaling between the source–destination pairs. The key design issues in providing adaptive services are as follows:

- How fast can the application service level be switched from *base QoS* to *enhanced QoS* and vice versa in response to changes in the network topology and channel conditions?

- How and when to operate on the *base QoS* or *enhanced QoS* level for an adaptive application (i.e., application that can sustain variation in QoS levels)?

This framework can scale down, drop, or scale up user sessions adaptively based on network dynamics and user-supplied adaptation policies. A key component of this framework is the INSIGNIA in-band signaling system, which supports fast reservation, restoration, and adaptation schemes to deliver the adaptive services. The signaling system is light-weight and responds rapidly to changes in the network topology and end-to-end QoS conditions. The INSIGNIA framework is depicted in Fig. 16. The routing module is independent of other components and hence any existing routing protocol can be used. INSIGNIA assumes that the routing protocol provides new routes in case of topology changes.

In-band signaling module is used to establish, adapt, restore, and tear down adaptive services between source–destination pairs. It is not dependent on any specific link layer protocol. In in-band signaling systems the control information is carried along with data packets and hence no explicit control channel is required. In INSIGNIA framework, each data packet contains an optional QoS field (INSIGNIA option) to carry the control information. The signaling information is encoded into this optional QoS field. The in-band signaling system can operate at speeds close to that of packet transmissions and is therefore better suited for highly dynamic mobile network environments. Admission control module uses soft state approach to allocate bandwidth to flows based on the maximum/minimum bandwidth requested. Packet forwarding module classifies the incoming packets and delivers them to the appropriate module. If the packet has an INSIGNIA option, it is delivered to the INSIGNIA signaling module. Packets that are to be routed to other nodes are handled by the packet-scheduling module. The packets to be transmitted by a node are scheduled by the scheduler based on the forwarding policy. INSIGNIA uses a weighted round robin service discipline. INSIGNIA framework is transparent to any underlying MAC protocol. The INSIGNIA framework uses a soft state resource management mechanism for efficient utilization of resources. When an intermediate node receives a data packet with RES (reservation) flag set for a QoS flow and no reservation has been done until now, the admission control module allocates the resources based on availability. If the reservation has been done already, it is re-confirmed. If no data packets

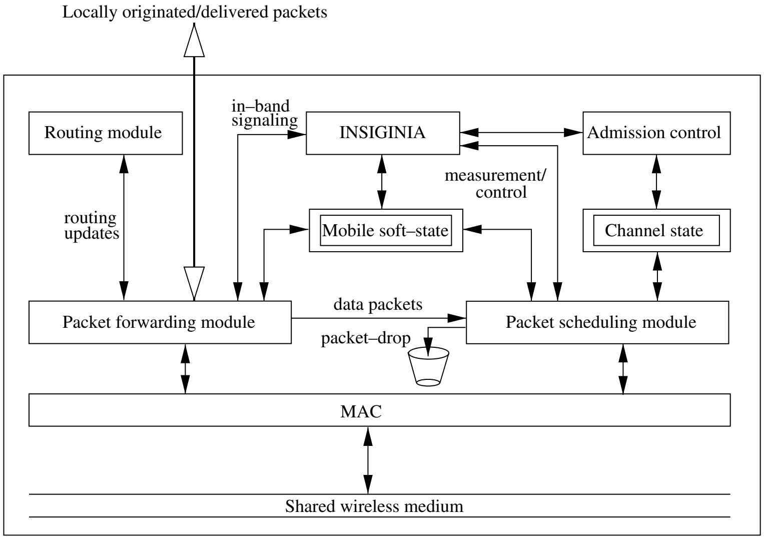


Fig. 16. INSIGNIA QoS framework.

are received for a specified timeout period, the resources are deallocated in a distributed manner without incurring any control overhead. In setting the value for the timeout period, care should be taken to avoid *false restoration* (which occurs when time interval is smaller than inter arrival time of packets) and resource lock-up (which occurs when the time interval is much greater than inter arrival time of packets).

7.3.1. Operation of INSIGNIA framework

The INSIGNIA framework supports adaptive applications which can be applications requiring best-effort service or applications with *base QoS* requirements or those with *enhanced QoS* requirements. Due to the adaptation of the protocol to the dynamic behavior of AWNs, the service level of an application can be degraded in a distributed manner if enough resources are not available.

The INSIGNIA option field contains the following information: service mode, payload type, bandwidth indicator, and bandwidth request. These indicate the dynamic behavior of the flow and the requirements of the application. The intermediate nodes take decisions regarding the flow state in a distributed manner based on the INSIGNIA option field. The service mode can be either best-effort (BE) or service requiring reservation (RES) of resources. The payload type indicates the QoS requirements of the application. It can be either *base QoS* for an application that requires minimum bandwidth, or *enhanced QoS* for an application which requires a certain maximum bandwidth but can operate with a certain minimum bandwidth below which they are useless. Examples of applications that require enhanced service mode are video applications that can tolerate packet loss and delay jitter to a certain extent. The bandwidth indicator flag has a value of MAX or MIN which represents the bandwidth available for the flow. Table 2 shows how service mode, payload type, and bandwidth indicator flags reflect the current status of flows. It can be seen from the table that the best-effort (BE) packets are routed as normal data packets. If QoS is required by an application, it can opt for *base QoS* in which a certain minimum bandwidth is guaranteed. For that application the bandwidth indicator flag is set to MIN. For *enhanced QoS*, the source sets the bandwidth indicator flag to MAX but it can be downgraded at the intermediate nodes to MIN; the service mode flag is changed to BE from RES if sufficient bandwidth is not available. The downgraded service can be restored to RES, if sufficient bandwidth becomes available. For *enhanced QoS*, the service can be downgraded either to BE service or RES service with *base QoS*. The downgraded *enhanced QoS* can be upgraded later, if all the intermediate nodes have the required (MAX) bandwidth.

Destination nodes actively monitor on-going flows, inspecting bandwidth indicator field of incoming packets and measuring the delivered QoS (for example, packet loss, delay, and throughput). Destination nodes send QoS reports (which contain information regarding the status of the on-going flows) to source nodes.

Route maintenance: Due to host mobility an ongoing session may have to be rerouted in case of a path break. The flow restoration process has to reestablish the reservation as quickly and efficiently as possible. During restoration, INSIGNIA does not preempt resources from the existing flows for admitting the rerouted flows. INSIGNIA supports

Table 2
INSIGNIA flags reflecting the behavior of flows

Service mode	Payload type	BW indicator	Degrading	Upgrading
BE	–	–	–	–
RES	Base QoS	MIN	Base QoS → BE	BE → Base QoS
RES	Enhanced QoS (EQoS)	MAX	EQoS → BE EQoS → BQoS	BE → EQoS BQoS → EQoS

three types of flow restoration viz., *immediate restoration* which occurs when a rerouted flow immediately recovers to its original reservation, *degraded restoration* which occurs when a rerouted flow is degraded for a period (T) before it recovers to its original reservation, and *permanent restoration* which occurs when the rerouted flow never recovers to its original reservation.

7.3.2. Advantages and disadvantages

INSIGNIA framework provides an integrated approach to QoS provisioning by combining in-band signaling, call admission control, and packet scheduling together. The soft state reservation scheme used in this framework ensures that resources are quickly released at the time of path reconfiguration. But, this framework supports only adaptive applications, for example, multimedia applications. Since this framework is transparent to any MAC protocol, fairness and reservation scheme of MAC protocol have a significant influence in providing QoS guarantees. Also as this framework assumes that routing protocol provides new routes in the case of topology changes, route maintenance mechanism of the routing protocol employed significantly affects the delivery of real-time traffic. If enough resources are not available because of the changing network topology, the *enhanced* QoS application may be downgraded to *base* QoS or even to best-effort service. As this framework uses in-band signaling, resources are not reserved before the actual data transmission begins. Hence INSIGNIA is not suitable for real-time applications that have stringent QoS requirements.

7.4. INORA

INORA [31] is a QoS framework for AWNs that makes use of the INSIGNIA in-band signaling mechanism and the TORA routing protocol [32]. The QoS resource reservation signaling mechanism interacts with routing protocol to deliver QoS guarantees. The TORA routing protocol provides multiple routes between a given source–destination pair. The INSIGNIA signaling mechanism provides feedback to the TORA routing protocol regarding the route chosen and asks for alternate routes if the route provided does not satisfy the QoS requirements. For resource reservation, a soft state reservation mechanism is employed. INORA can be classified into two schemes: *coarse feedback scheme* and *class-based fine feedback scheme*.

7.4.1. Coarse feedback scheme

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

7.4.2. Class-based fine feedback scheme

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

1. Let the DAG created by the TORA protocol be as shown in Fig. 17. Let $S \rightarrow A \rightarrow B \rightarrow D$ be the path chosen by the TORA routing protocol.
2. INSIGNIA tries to establish soft state reservations for the QoS flow along the path. Assume that node A has admitted the flow with class m successfully and node B has admitted the flow with bandwidth of class l ($l < m$) only.
3. Node B sends an *admission report* message ($AR(l)$) to upstream node A, indicating its ability to give only class l bandwidth to the flow.
4. Node A splits the flow in the ratio of l to $m - l$ and forwards the flow to node B and node Y, in that ratio.

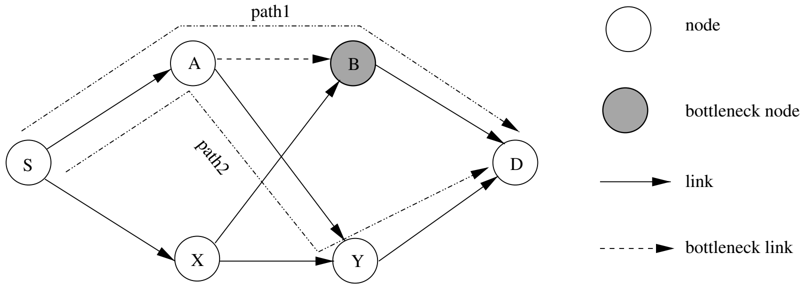


Fig. 17. INORA fine feedback scheme: node A has admitted the flow with class m, but node B is able to give it class l ($l < m$).

5. If node Y is able to give class $(m - l)$ as requested by node A, then the flow of class m is split into two flows, one flow with bandwidth of class l along the path $S \rightarrow A \rightarrow B \rightarrow D$ and the other one with bandwidth of class $(m - l)$ along path $S \rightarrow A \rightarrow Y \rightarrow D$.
6. If node Y gives only class n ($n < m - l$), it sends an $AR(n)$ message to the upstream node A.
7. Node A, realizing that its downstream neighbors are unable to give class m service, informs its ability to provide service class of $(l + n)$ by sending an $AR(l + n)$ to node S.
8. Node S tries to find another downstream neighbor, which might be able to accommodate the flow with class $(m - (l + n))$.
9. If no such neighbor is available, node S rejects the flow.

7.4.3. Advantages and disadvantages

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

7.5. SWAN

Ahn et al. proposed a distributed network model called stateless wireless ad hoc networks (SWAN) [33] that assumes a best-effort MAC protocol and uses feedback based control mechanisms to support real-time services and service differentiation in AWNs. SWAN uses a local rate control mechanism for regulating injection of best-effort traffic into the network, a source-based admission control while accepting new real-time sessions, and an explicit congestion notification (ECN) mechanism for dynamically regulating admitted real-time sessions. In this model intermediate nodes are relieved from the responsibility of maintaining per-flow or aggregate state information unlike stateful QoS models such as INSIGNIA and INORA. Changes in topology and network conditions, even node and link failures, do not affect the operation of the SWAN control system. This makes the system simple, robust, and scalable.

7.5.1. SWAN model

The SWAN model has several control modules which are depicted in Fig. 18. Upon receiving a packet from the IP layer, the *packet classifier* module checks whether it is marked (i.e., real-time packet) or not (i.e., best-effort packet). If it is a best-effort packet, it is forwarded to the *traffic shaper* for regulation. If it is a real-time packet, the module directly forwards it to the MAC layer bypassing the *traffic shaper*. The *traffic shaper* represents a simple leaky bucket traffic policy. The traffic shaper delays best-effort packets in conformance with the rate calculated by the *traffic rate controller*. The *call admission controller* module is responsible for admitting or rejecting new real-time sessions. The decision on whether to admit or reject a real-time session is taken solely by the source node based on the result of an end-to-end

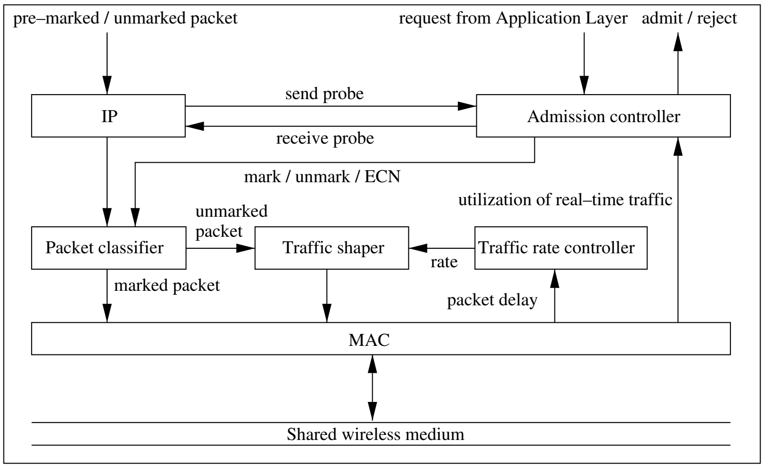


Fig. 18. The SWAN model.

request/response probe. The SWAN distributed control algorithms are described in the following sections.

7.5.2. Local rate control of best-effort traffic

The SWAN model assumes that most of the traffic existing in the network is best-effort, which can serve as a "buffer zone" or absorber for real-time traffic bursts introduced by mobility (because of rerouting of the already admitted real-time sessions) or traffic variations (for example, bursty data). The best-effort traffic can be locally and rapidly rate controlled in an independent manner at each node in order to yield the necessary low delays and stable throughput for real-time traffic. The best-effort traffic utilizes remaining bandwidth (if any) left out by real-time traffic. Hence this model does not work in scenarios where most of the traffic is real-time in nature.

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

7.5.3. Source-based admission control of real-time traffic

The process of admitting a new real-time session is as follows. The admission controller module at the source node sends a probing request packet towards the destination node to assess the end-to-end bandwidth availability. This is a best-effort control packet that contains a bottleneck bandwidth field. Each intermediate node on the path between the source–destination pair that receives the probing request packet updates the bottleneck bandwidth field in the packet if the bandwidth availability at the node is less than the current value of the field. On receiving the probing request packet, the destination node sends a probing response packet back to the source node with the bottleneck field copied from the received probing request packet. After receiving the response message, the source node admits the new real-time session only if sufficient end-to-end bandwidth is available. In this model, no bandwidth request is carried in the probing message, no admission control is done at intermediate nodes, and no resource allocation or reservation is done on behalf of the source node during the lifetime of an admitted session.

7.5.4. Regulation algorithms

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

The ECN-based regulation of real-time sessions operates as follows. Each node continuously estimates the locally available bandwidth. When a node detects congestion/overload conditions, it starts marking the ECN bits in the IP header of the real-time packets. If the destination receives a packet with ECN bits marked, it notifies the source using a regulate message. After receiving a regulate message, the source node initiates re-establishment of its real-time session. If the node detecting violations marks (i.e., sets) the ECN bits of all packets, then all sessions passing through this node are forced to re-establish their sessions at the same instance. Since such an approach is inefficient, the SWAN model considered two approaches in which only a small number of sources are penalized.

Source-based regulation: In this scheme the source node waits for a random amount of time after receiving a regulate message from a congested or overloaded intermediate node on the path to the destination node and then initiates the re-establishment process. This can avoid flash-crowd conditions. In this scheme the rate of the real-time traffic will gradually decrease until it reaches below the admission control rate. Then the congested or overloaded nodes will stop marking packets. Even though this scheme is simple and source-based, it has a disadvantage that sources that regulate earlier than other sources are more likely to find the path overbooked and be forced to terminate their sessions.

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

7.5.5. Advantages and disadvantages

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

7.6. Proactive RTMAC

Proactive RTMAC (PRTMAC) [34] is a cross layer framework, with an on-demand QoS extension of DSR routing protocol at the network layer and RTMAC (real-time MAC) [13] protocol at the MAC layer. PRTMAC is a tightly coupled solution, which requires the bandwidth reservation and bandwidth availability estimation services from the underlying MAC protocol. It is designed to provide enhanced real-time traffic support and service differentiation to highly mobile ad hoc wireless networks such as that formed by military combat vehicles. The performance of real-time sessions in ad hoc wireless networks are affected by mobility of nodes in many different ways.

The two major ways in which mobility affects real-time sessions are *breakaway*s and reservation *clash*s. If a node participating in a QoS session moves out of the transmission range of either or both of its upstream and downstream nodes, we say the QoS session is broken due to breakaway. Assume that node *A* is transmitting to node *B* over a given slot (say slot #1). Similarly, at some other region in the network, node *C* is transmitting to node *D* over the same slot (slot #1). Now, if node *C* moves into the transmission range of node *B* (assume no breakaway due to mobility for the session between nodes *C* and *D*), packets transmitted by nodes *A* and *C* result in a collision at node *B*. This problem is referred as clash.

7.6.1. Operation of PRTMAC

The PRTMAC framework is shown in Fig. 19. RTMAC [13] is used as the MAC protocol. The out-of-band signaling channel gathers additional information about the ongoing real-time sessions, such that proactive measures can be taken to protect these sessions from breakaways and clashes. A narrow band control channel that operates over a transmission range with twice that of the data transmission range, is used as the out-of-band signaling channel. Every node sends out control beacons (short fixed sized packets) at regular intervals over the control channel. The information carried by the beacons, and the beacon itself, are used by the nodes to gather information about real-time sessions. Firstly, the signal strength of the received beacon is used to gain an idea about the relative distance of the node which sent the beacon. Further, the information carried by the beacon is used in predicting breakaways and clashes. The beacons carry information about each of the sessions that the originating node is carrying, and the slots in the super-frame that have been reserved for them. Each node originates periodic beacons on the control channel. The beacon has information about all on-going real-time sessions at the node. The information includes the start and end times of the reservation slot of each session, the sender and the receiver of the session, and the service class (service classes are used to provide differentiated services among the real-time sessions existing in the system, for example, the

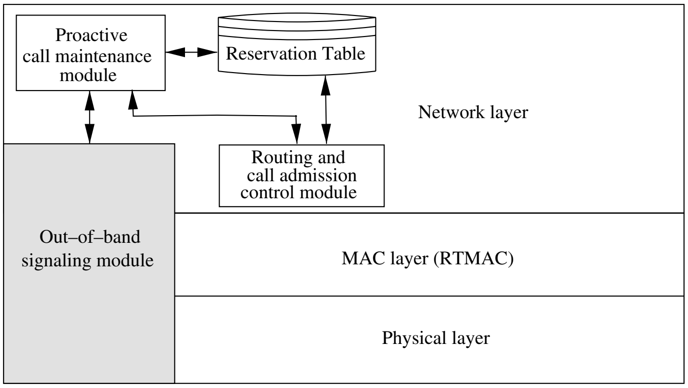

Proactive call maintenance module
Reservation Table
Network layer
Routing and call admission control module
Out–of–band signaling module
MAC layer (RTMAC)
Physical layer

Fig. 19. Modules in PRTMAC framework.

command and control sessions in a military communication system may require higher priority than the other sessions) to which the session belongs. The range of the control channel has to be sufficiently larger than that of the data channel so that all possible events that can cause a session to be interrupted can be discovered well in advance.

Crossover-time prediction: Crossover-time is defined as the time at which a node crosses another node's data transmission range *r*. This event is defined as *crossover*. As apparent from Fig. 20(a) and (b), there are two different *crossover-time*s, namely *crossover-time-in* and *crossover-time-out*.

The *crossover-time-in* is the expected time at which node *B* in Fig. 20(a), reaches the crossover-point such that a bidirectional link forms between nodes *A* and *B*. Fig. 20(b) shows the *crossover-time-out*, which happens at the instant node *B* moves away from node *A* such that the link between nodes *A* and *B* breaks. Each node (say node *A*), upon reception of every new beacon from another node (say node *B*), predicts the *crossover-time* based on the signal strength history obtained from past beacons i.e., if node *B* is inside the range of the data channel of node *A*, node *A* predicts the *crossover-time-out*, and if node *B* is outside the range of the data channel of node *A*, node *A* predicts the *crossover-time-in*. The prediction of *crossover-time-out* of node *B* with respect to node *A* is performed by keeping track of the signal strengths of the beacons previously sent by node *B* to node *A*. A node stores a fixed number of ⟨*time*, *signal strength*⟩ tuples of the beacons received from any other node. Using this, it generates a polynomial on the variation of signal strength with time. The roots of the polynomial refer to the time at which the signal strength can cross a receiving threshold. When node *A* predicts that node *B* is going to cross the data channel range within the next beacon interval, it takes proactive actions described in the next section. If node *B* is already within the data channel range of node *A*, then the prediction will be for a *crossover-out* event, and all sessions going on between nodes *A* and *B* will be interrupted. If node *B* is outside the range of node *A*, then it is a *crossover-in* event, and any packets belonging to existing real-time sessions at node *A* and node *B* will collide if their reservation times overlap. Note that if the predicted time of entry is beyond the next beacon interval, no action needs to be taken as of now, since the event would be predicted again, on receipt of the next beacon.

Handling breakaways: The event of breakaways can be handled in two different ways, first is the local reconfiguration and second is the end-to-end reconfiguration. In local reconfiguration, the upstream node (say node *U*) that has detected breakaway takes the responsibility and issues fresh route probe packets to obtain a path with reservation from that node to the destination. But, in the case of end-to-end reconfiguration, node *U* informs the source node about the breakaway, so that the source finds a new path to the destination. In PRTMAC a combination of the above two types is attempted which is described as below: Node

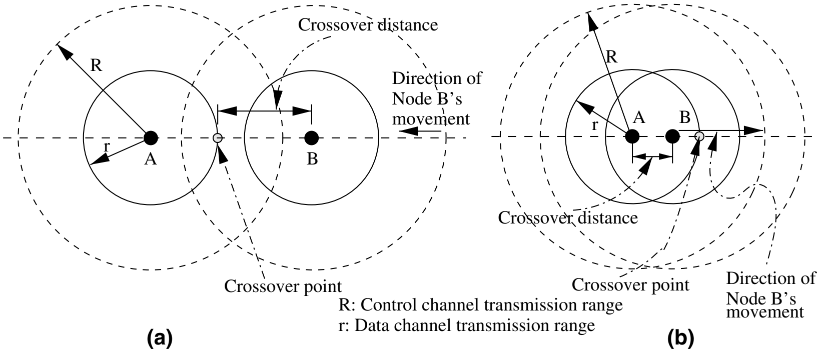

Crossover distance
R
r
A
B
Direction of Node B's movement
Crossover point
R
r
A
B
Crossover distance
Crossover point
Direction of Node B's movement
R: Control channel transmission range
r: Data channel transmission range
(a)
(b)

Fig. 20. Illustration of *crossover-in* and *crossover-out*.

U checks if its routing table has another path towards the destination node (say node *F*). If there exists such a node, then node *U* makes reservations on the link *U–F* for the on-going session. If the session is interrupted and reconfigured locally a number of times, then end-to-end reconfiguration is attempted.

Handling clashes: Fig. 21(a) illustrates how two nodes can reside safely within range of each other if the reserved slots do not overlap with each other. If the reservation slots clash for the two nodes, as indicated in Fig. 21(b), then PRTMAC handles it in such way that the flow between say node *N* and node *C* is assigned to a new slot (#5) as shown in Fig. 22. In the absence of any measures taken to resolve a clash, both the sessions that experience a clash will be reconfigured from the source to the destination, resulting in degradation of performance. PRTMAC prevents such an occurrence to the extent possible, by pro-actively shifting one of the sessions to a new slot, so that the two sessions do not clash. This benefit of clash resolution is more important when a higher priority session clashes with a lower priority session. In such a case, the node having the low priority session has to reconfigure it to a new slot.

As illustrated in Fig. 22, the node whose responsibility it is to reconfigure the session is denoted by node *N*, the other node, whose session clashes with node *N*s session, is denoted by node *O*, and the counterpart of node *N* in its session by node *C*. Node *N* goes through its reservation tables and its neighbor reservation table corresponding to node *C* and tries to come up with a free reservation slot in both nodes *N* and *C* large enough to accommodate the session to be shifted. If it succeeds in finding such a free slot, the existing reservations for the session have to be dropped and new reservations have to be made for the session in the free slot. This is achieved by the originator of the session freeing the earlier reservation and issuing a request for the reservation of the slots belonging to the free slot.

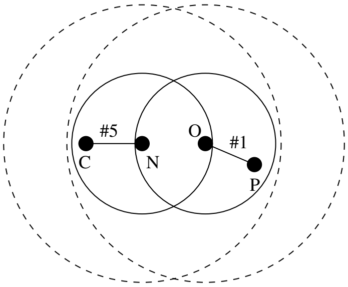


Fig. 22. Clash handling.

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

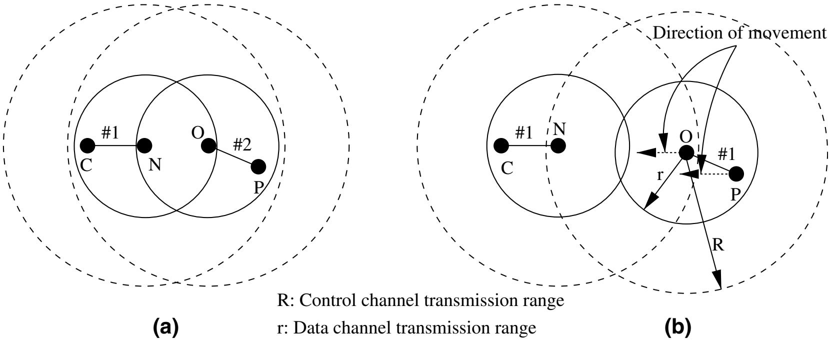

#1
C
N
O
#2
P
(a)
Direction of movement
#1
N
C
O
#1
r
P
R
R: Control channel transmission range
r: Data channel transmission range
(b)

Fig. 21. (a) No clash and (b) before clash.

7.6.2. Advantages and disadvantages

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

8. Summary

In this paper several solutions proposed in the literature for QoS provisioning in AWNs were discussed. First the issues and challenges involved in providing QoS in AWNs were identified. Then the existing QoS approaches were classified according to several criteria such as interaction between routing protocol and resource reservation signaling, interaction between network and MAC layer, and routing information update mechanism. A layer-wise classification of the existing QoS solutions was also provided. The existing QoS solutions were then discussed in a layer-wise order. Finally, some of the important QoS frameworks for AWNs were described.

References

[1] P. Karn, MACA: A new channel access method for packet radio, in: Proceedings of ARRL/CRRL Amateur Radio 9th Computer Networking Conference, September 1990, pp. 134–140.

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

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

[4] J. Deng, Z.J. Haas, Dual busy tone multiple access (DBTMA): a new medium access control for packet radio networks, in: Proceedings of IEEE ICUPC 1998, vol. 1, October 1998, pp. 973–977.

[5] M. Gerla, J.T.C. Tsai, Multicluster, mobile, multimedia radio network, Wireless Networks 1 (3) (1995) 255–265.

[6] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor, IEEE 802.11e wireless LAN for quality of service, in: Proceedings of the European Wireless 2002, vol. 1, February 2002, pp. 32–39.

[7] IEEE 802.11 TGe, EDCF proposed draft text, TR-01/131r1, March 2001.

[8] IEEE 802.11 TGe, Hybrid coordination function (HCF)—proposed updates to normative text of D0.1, TR-01/110r1, March 2001.

[9] IEEE 802.11 TGe, HCF ad hoc group recommendation—normative text to EDCF access category, TR-02/241r0, March 2001.

[10] IEEE 802.11 TGe, Proposed normative text for AIFS—revisited, TR-01/270r0, February 2003.

[11] S. Sheu, T. Sheu, DBASE: A distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LAN, in: Proceedings of IEEE INFOCOM 2001, vol. 3, April 2001, pp. 1558–1567.

[12] C.R. Lin, M. Gerla, Real-time support in multihop wireless networks, Wireless Networks 5 (2) (1999) 125–135.

[13] B.S. Manoj, C. Siva Ram Murthy, Real-time traffic support for ad hoc wireless networks, in: Proceedings of IEEE ICON 2002, August 2002, pp. 335–340.

[14] S. Chen, K. Nahrstedt, Distributed quality-of-service routing in ad hoc networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1488–1504.

[15] S.H. Shah, K. Nahrstedt, Predictive location-based qos routing in mobile ad hoc networks, in: Proceedings of IEEE ICC 2002, vol. 2, May 2002, pp. 1022–1027.

[16] S. De, S.K. Das, H. Wu, C. Qiao, Trigger-based distributed QoS routing in mobile ad hoc networks, ACM SIGMOBILE Mobile Computing and Communications Review 6 (3) (2002) 22–35.

[17] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, February 1999, pp. 90–100.

[18] C.E. Perkins, E.M. Royer, S.R. Das, Quality of service for ad hoc on-demand distance vector routing (work in progress), IETF Internet Draft, draft-ietf-manet-aodvqos-00.txt, July 2000.

[19] C.R. Lin, J. Liu, QoS routing in ad hoc wireless networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1426–1438.

[20] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proceedings of ACM SIGCOMM 1994, vol. 24 (4), October 1994, pp. 234–244.

[21] C.R. Lin, On-demand QoS routing in multihop mobile networks, in: Proceedings of IEEE INFOCOM 2001, vol. 3, April 2001, pp. 1735–1744.

[22] Y. Chen, Y. Tseng, J. Sheu, P. Kuo, On-demand, link-state, multi-path QoS routing in a wireless mobile ad-hoc network, in: Proceedings of European Wireless 2002, February 2002, pp. 135–141.

[23] V. Vidhyashankar, B.S. Manoj, C. Siva Ram Murthy, Slot allocation schemes for delay sensitive traffic support in

asynchronous wireless mesh networks, in: Proceedings of HiPC 2003, December 2003.

[24] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–181.

[25] P. Sinha, R. Sivakumar, V. Bharghavan, CEDAR: A core extraction distributed ad hoc routing algorithm, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1454–1466.

[26] R. Braden, D. Clark, S. Shenker, Integrated services in the Internet architecture: an overview, in: IETF RFC1633, June 1994.

[27] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, in: IETF RFC2475, December 1998.

[28] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource reservation protocol (RSVP)—Version 1 functional specification, IETF RFC 2205, September 1997.

[29] H. Xiao, K.G. Seah, A. Lo, K.C. Chua, A flexible quality of service model for mobile ad-hoc networks, in: Proceedings of IEEE Vehicular Technology Conference, vol. 1, May 2000, pp. 445–449.

[30] S.B. Lee, A. Gahng-Seop, X. Zhang, A.T. Campbell, INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks, Journal of Parallel and Distributed Computing 60 (4) (2000) 374–406.

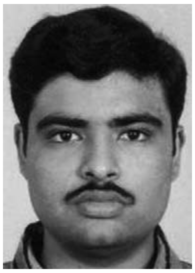
[31] D. Dharmaraju, A.R. Chowdhury, P. Hovareshti, J.S. Baras, INORA—A unified signalling and routing mechanism for QoS support in mobile ad hoc networks, in: Proceedings of ICPPW 2002, August 2002, pp. 86–93.

[32] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of IEEE INFOCOM 1997, vol. 3, April 1997, pp. 1405–1413.

[33] H. Ahn, A.T. Campbell, A. Veres, L. Sun, Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks, IEEE Transactions on Mobile Computing 1 (3) (2002) 192–207.

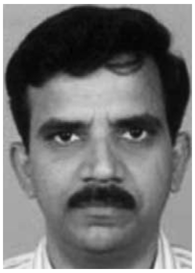
[34] V. Vivek, T. Sandeep, B.S. Manoj, C. Siva Ram Murthy, A novel out-of-band signaling mechanism for enhanced real-time support in tactical ad hoc wireless networks, in: Proceedings of IEEE RTAS 2004, May 2004.

T. Bheemarjuna Reddy received the B.Tech. degree in Computer Science and Engineering from Andhra University, India, in 2000 and the M.E. degree in Computer Science and Engineering from the National Institute of Technology (NIT), Rourkela, India, in 2002. He is currently a doctoral student in the Department of Computer Science and Engineering at the Indian Institute of Technology (IIT), Madras, India. His research interests include QoS provisioning and Multimedia streaming in Ad hoc wireless networks.



I. Karthigeyan received the B.E. degree in Computer Science and Engineering from University of Madras, Tamilnadu, India, in 2000. He is currently pursuing his MS (by Research) degree in Computer Science and Engineering at the Indian Institute of Technology (IIT), Madras, India. His research interests include Wireless Networks and Optical Networks.

B.S. Manoj completed his graduation in 1995 and post graduation in 1998 both in Electronics and Communication Engineering from Institution of Engineers (India) and Pondicherry Central University, Pondicherry, India, respectively. He has worked as a Senior Engineer with Banyan Networks Pvt. Ltd., Chennai, India from 1998 to 2000 where his primary responsibility included design and development of protocols for real-time traffic support in data networks. He has been an Infosys doctoral student in the Department of Computer Science and Engineering at the Indian Institute of Technology (IIT) Madras, India, where he focused on the development of architectures and protocols for Ad hoc wireless networks and next generation hybrid wireless network architectures. Indian Science Congress Association has awarded him the Young Scientist Award for the Year 2003. Since January 2004, he is a Project Officer at the Department of Computer Science and Engineering, Indian Institute of Technology (IIT), Madras, India. His current research interests include Ad hoc wireless networks, next generation wireless architectures, and wireless sensor networks.



C. Siva Ram Murthy received the B.Tech. degree in Electronics and Communications Engineering from Regional Engineering College (now National Institute of Technology), Warangal, India, in 1982, the M.Tech. degree in Computer Engineering from the Indian Institute of Technology (IIT), Kharagpur, India, in 1984, and the Ph.D. degree in Computer Science from the Indian Institute of Science, Bangalore, India, in 1988. He joined the Department of Computer Science and Engineering at IIT, Madras, as a Lecturer in September 1988 and became an Assistant Professor in August 1989 and an Associate Professor in May 1995. He has been a Professor with the same department since September 2000. He has held visiting positions at the German National Research

Centre for Information Technology (GMD), Bonn, Germany, the University of Stuttgart, Germany, the University of Freiburg, Germany, the Swiss Federal Institute of Technology (EPFL), Switzerland, and the University of Washington, Seattle, USA. He has to his credit over 100 research papers in international journals and over 75 international conference publications. He is the co-author of the textbooks *Parallel Computers: Architecture and Programming* (Prentice-Hall of India, New Delhi, 2000), *New Parallel Algorithms for Direct Solution of Linear Equations* (John Wiley & Sons, Inc., USA, 2000), *Resource Management in Real-time Systems and Networks* (MIT Press, USA, 2001), and *WDM Optical Networks: Concepts, Design, and Algorithms* (Prentice-Hall PTR, USA; reprinted by Prentice-Hall of India, New Delhi, India). He is a recipient of the Best Ph.D. Thesis Award and also of the Indian National Science Academy Medal for Young Scientists. He is a co-recipient of Best Paper Awards from 5th IEEE International Workshop on Parallel and Distributed Real-Time Systems held in Geneva, Switzerland in 1997 and 6th International Conference on High Performance Computing held in Calcutta, India in 1999. He is a Fellow of Indian National Academy of Engineering. His research interests include Parallel and Distributed Computing, Real-time Systems, Lightwave Networks, and Wireless Networks.