RESOURCE MANAGEMENT IN WIRELESS NETWORKING
M. Cardei, I. Cardei and D.-Z. Du (Eds.) pp. – - –

Quality of Service Routing in Mobile Ad Hoc Networks

Imad Jawhar
Department of Computer Science and Engineering
Florida Atlantic University, Boca Raton, FL 33431
E-mail: imadj@cse.fau.edu

Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University, Boca Raton, FL 33431
E-mail: jie@cse.fau.edu

Contents

Abstract

With the continuing advances in computing and wireless technologies, Mobile Ad Hoc networks (MANETs) are expected to become an indispensable part of the computing environment in the near future. Wireless devices are constantly growing in computing speed, memory, communication capabilities and features, while shrinking in weight and size. With this growth and the proliferation of these devices in every aspect of society, the need for such devices to communicate in a seamless manner is becoming increasingly essential. Multiple routing protocols have been developed for MANETs [29]. As MANETs gain popularity, their need to support real time and multimedia applications is growing as well. Such applications have stringent quality of service requirements such as bandwidth, delay, and delay jitter. Design and development of routing algorithms with QoS support is experiencing increased research interest. Several approaches which propose various routing algorithms with QoS support for MANETs have been presented in the research. This chapter discusses the issues and challenges of QoS routing in MANETS. Furthermore, a classification of these QoS routing algorithms is presented. The protocols are classified according to the most closely related best effort algorithm, as well as the model and environment they assume, and the communication layer within which they operate.

Keywords: Code Division Multiple Access (CDMA), mobile ad hoc networks (MANETs), quality-of-service (QoS), routing, Time Division Multiple Access (TDMA), wireless networks.

1 Introduction

As technology advances, wireless and portable computers and devices are becoming more powerful and capable. These advances are marked by an increase in CPU speed, memory size, disk space, and a decrease in size and power consumptions. The need for these devices to continuously communicate with each other and with wired networks is becoming increasingly essential. Mobile ad hoc networks (MANETs) open the door for these devices to establish networks on the fly, i.e., formally, a MANET is a collection of mobile devices which form a communication network with no pre-existing wiring or infrastructure. They allow the applications running on these wireless devices to share data of different types and characteristics. There are many applications of MANETs, each with different characteristics

of network size (geographic range and number of nodes), node mobility, rate of topological change, communication requirements, and data characteristics. Such applications are conferences, classroom, campus, military, and disaster recovery. Each node is directly connected to all nodes within its own effective transmission range. Nodes in the network are allowed to move in and out of range of each other. Communication between nodes that are not within range of each other is accomplished by establishing and using multi-hop routes that involve other nodes which act as routers. New nodes can join the network at any time and existing nodes can leave the network as well.

Due to the dynamic nature of MANETs, designing communications and networking protocols for these networks is a challenging process. One of the most important aspects of the communications process is the design of the routing protocols used to establish and maintain multi-hop routes to allow the communication of data between nodes. A considerable amount of research has been done in this area, and multi-hop routing protocols have been developed. Most of these protocols such as the Dynamic Source Routing protocol (DSR) [29], Ad Hoc on Demand Distance Vector protocol (AODV) [31], Temporally Ordered Routing Protocol (TORA) [28], and others establish and maintain routes on a best-effort basis. While this might be sufficient for a certain class of MANET applications, it is not adequate for the support of more demanding applications such as multimedia audio and video. Such applications require the network to provide guarantees on the Quality of Service (QoS).

Some researchers have been active in the area of QoS support in MANETs, and have proposed numerous QoS routing protocols for this environment. Most of these protocols provide QoS support for the available bandwidth requirement for a given path. This is because bandwidth is the most critical parameter in most MANET applications due to the scarcity of this resource in the wireless environment. The protocols that are discussed in this chapter support quality of service to varying degrees, in different ways, and using various network and and communication models.

In this chapter, these different approaches are presented, discussed and classified according to which of the main existing best-effort routing algorithms (DSR, AODV, DSDV, TORA, etc) they extend or are most closely related. In addition, some protocols are based on new algorithms. The QoS routing protocols that are discussed operate in both the network layer and the medium access control (MAC) layer which is equivalent to the data link layer in the OSI model. There are also design approaches, such as the IP-Based quality of service framework for MANETs (INSIGNIA) [1][15][16] and the integrated mobile ad hoc QoS framework (iMAQ) [3], which are designed to support multimedia traffic and achieve better efficiency in terms of bandwidth and energy consumption through the im-

plementation of inter-layer QoS frameworks. As these protocols are classified and presented, the networking layer, or layers in some cases, within which they operate, as well as their assumed QoS model will be identified and discussed.

The remainder of this chapter is organized in the following manner. First, an overview of the topic is presented which includes a brief look at the best-effort routing algorithms which are both most popular and most used. Then the typical models used for QoS routing protocols are discussed along with the different layers within which the QoS support mechanisms are included. Existing QoS routing protocols are then classified according to which best effort routing protocol they extend or to which they are closest in design. As these protocols are discussed, the QoS model and networking environment they assume is identified along with the communication layer within which they are designed to operate. The chapter is completed with a conclusions and future research section.

2 Overview

Most QoS routing algorithms represent an extension of existing classic (or major) best-effort routing algorithms. Many routing protocols have been developed which support establishing and maintaining multi-hop routes between nodes in MANETs. These algorithms can be classified into two different categories: on-demand (reactive) such as DSR, AODV, and TORA, and table-driven (proactive) such as Destination Sequenced Distance Vector protocol (DSDV). In the on-demand protocols, routes are discovered between a source and a destination only when the need arises to send data. This provides a reduced overhead of communication and scalability. In the table-driven protocols, routing tables which contain routing information between all nodes are generated and maintained continuously regardless of the need of any given node to communicate at that time. With this approach, the latency for route acquisition is relatively small, which might be necessary for certain applications, but the cost of communications overhead incurred in the continued update of information for routes which might not be used for a long time if at all is too high. Furthermore, this approach requires more memory due to significant increase in the size of the routing table. These requirements put limits on the size and density of the network. A third hybrid approach, the Zone Routing Protocol (ZRP), has also been proposed and attempts to reap the benefits of both methods. In ZRP, the network is divided into zones. A proactive table driven strategy is used for establishment and maintenance of routes between nodes of the same zone, and a reactive on-demand strategy is used for communication between nodes of different zones. This approach can be effective in larger networks with applications that exhibit a relatively high degree of locality of communication, where communication

between nodes with close proximity to one another is much more frequent than that between nodes which are further apart.

Before presenting the current approaches for design and implementation of QoS routing protocols, it is important to briefly discuss the existing best-effort routing protocols which exist for MANETs. Many routing protocols have been designed to discover and maintain routes between source and destination nodes.

Among the most important and classic routing algorithms for MANETs that have evolved are three basic types. Each of these three basic types has its own advantages, disadvantages, and appropriateness of use in certain types of ad hoc networks depending on the mobility, number of nodes involved, node density, underlying link layer technology, and general characteristics of the environment and applications being supported. These three routing algorithms are: (1) reactive (on-demand) such as DSR (Dynamic Source Routing) protocol, AODV (Ad hoc On Demand Distance Vector) routing protocol, and TORA (Temporally Ordered Routing Alogrithm) protocol, and (2) proactive (table-driven) such as DSDV (Destination Sequenced Distance Vector) protocol. There are also other types of routing protocols designed for more scalability such as (3) the ZRP (Zone Routing Protocol), which is a hybrid framework for routing in ad hoc networks (proactive within the zone and reactive between zones), in addition to others, which will be mentioned subsequently.

2.1 DSR - the Dynamic Source Routing Protocol

DSR is one of the most well-known routing algorithms for ad hoc wireless networks. It was originally developed by Johnson, Maltz, and Broch [29]. DSR uses source routing, which allows packet routing to be loop free. It increases its efficiency by allowing nodes that are either forwarding route discovery requests or overhearing packets through promiscuous listening mode to cache the routing information for future use. DSR is also on demand, which reduces the bandwidth use especially in situations where the mobility is low. It is a simple and efficient routing protocol for use in ad hoc networks. It has two important phases, route discovery and route maintenance. The main algorithm works in the following manner. A node that desires communication with another node first searches its route cache to see if it already has a route to the destination. If it does not, it then initiates a route discovery mechanism. This is done by sending a Route Request message. When the node gets this route request message, it searches its own cache to see if it has a route to the destination. If it does not, it then appends its id to the packet and forwards the packet to the next node; this continues until either a node with a route to the destination is encountered (i.e. has a route in its own cache) or the destination receives the packet. In that case, the node sends a route reply packet which

has a list of all of the nodes that forwarded the packet to reach the destination. This constitutes the routing information needed by the source, which can then send its data packets to the destination using this newly discovered route. Although DSR can support relatively rapid rates of mobility, it is assumed that the mobility is not so high as to make flooding the only possible way to exchange packets between nodes.

2.2 AODV - The Ad Hoc On-demand Distance-Vector Protocol

AODV [31] is another routing algorithm used in ad hoc networks. Unlike DSR (Dynamic Source Routing Algorithm) it does not use source routing, but like DSR it is on-demand. In AODV, each node maintains a routing table which is used to store destination and next hop IP addresses as well as destination sequence numbers. Each entry in the routing table has a destination address, next hop, precursor nodes list, lifetime, and distance to destination.

To initiate a route discovery process a node creates a route request (RREQ) packet. The packet contains the source node's IP address as well as the destination's IP address. The RREQ contains a broadcast ID, which is incremented each time the source node initiates a RREQ. The broadcast ID and the IP address of the source node form a unique identifier for the RREQ. The source node then broadcasts the packet and waits for a reply. When an intermediate node receives a RREQ, it checks to see if it has seen it before using the source and broadcast ID's of the packet. If it has seen the packet previously, it discards it. Otherwise it processes the RREQ packet. To process the packet the node sets up a reverse route entry for the source node in its route table which contains the ID of the neighbor through which it received the RREQ packet. In this way, the node knows how to forward a route reply packet (RREP) to the source if it receives one later. When a node receives the RREQ, it determines if indeed it is the indicated destination and, if not, if it has a route to respond to the RREQ. If either of those conditions is true, then it unicasts a route replay (RREP) message back to the source. If both conditions are false,i.e. if it does not have a route and it is not the indicated destination, it then broadcasts the packet to its neighbors. Ultimately, the destination node will always be able to respond to the RREQ message. When an intermediate node receives the RREP, it sets up a forward path entry to the destination in its routing table. This entry contains the IP address of the destination, the IP address of the neighbor from which the RREP arrived, and the hop count or distance to the destination. After processing the RREP packet, the node forwards it toward the source. The node can later update its routing information if it discovers a better route. This could be used for QoS routing support to choose between routes based on different criteria such as reliability and delay. To provide such support additional QoS attributes would need

to be created, maintained, and stored for each route in the routing table to allow the selection of the appropriate route among multiple routes to the destination.

2.3 TORA - The Temporally Ordered Routing Algorithm

TORA [28] is the most well known LRR (Link Reversal Routing) algorithm [29] which provides a very adaptive type of routing. It is intended to be used in networks with rapidly changing topologies. It uses a strategy of de-coupling of far-reaching control message propagation from the dynamics of the network's topology. It is efficient to use TORA in networks where the rate of topology changes is not so fast as to make flooding the only form of transmitting messages and not so slow as to make the use of algorithms supporting shortest path calculations applicable. Therefore, the algorithm's applicability is a function of the network's size, rate of topological changes, and available bandwidth. TORA minimizes the network messages in reaction to changes in topology, which are caused by link activation and failure. The algorithm localizes the reaction to these topological changes. TORA does not maintain information sufficient to support shortest path calculation, and maintains only state information sufficient to form a DAG (directed acyclic graph) routed at the destination. The destination is therefore the only node with no outgoing links (a sink). The maintenance of the DAG provides loop free communication to the destination. It also allows the existence of multiple paths to the destination. This provides good reliability, which is desirable in ad hoc networks, and possible QoS extension support, by selecting paths with particular characteristics and that can support pre-specified QoS constraints.

TORA is source initiated and demand driven. Therefore, due to its nature, it forgoes optimal routing. It does not make sure to select the shortest possible path, even though it can be shown that due to the nature of RPY message propagation, shorter paths are more likely to form. However, it provides routing which is very adaptive and scalable with relatively small overhead bandwidth usage for control messages. In addition, lower delivery latency can be achieved.

In contrast with other earlier LLR (Link Reversal Routing) algorithms, TORA's key feature is its reaction to link failures. This reaction is structured as a temporally ordered sequence of diffusing computations with each computation consisting of a sequence of directed link reversals. Each link reversal sequence effectively conducts a search for alternative routes to the destination. The search mechanism in TORA often involves only a single pass of the distributed algorithm because it simultaneously modifies the routing tables during the outward phase of the search procedure itself. This is not the case in other approaches such as DSR and AODV which take three-pass procedures (i.e. route-error/route-request/route-reply) to discover new routes when a node loses its last route [29]. The algorithm uses a "phys-

ical or logical clock" to provide a temporal order of topological change events, which is used to structure the protocol's reaction to changes. More information on TORA is available in [29] and in [28].

2.4 DSDV - The Destination Sequenced Distance Vector Protocol

DSDV [29] is one of the most well known table-driven routing algorithms for MANETs. It is a distance vector protocol. In distance vector protocols, every node i maintains for each destination x a set of distances $\{d_{ij}(x)\}$ for each node j that is a neighbor of i. Node i treats neighbor k as a next hop for a packet destined to x if $d_{ik}(x)$ equals $min_j\{d_{ij}(x)\}$. The succession of next hops chosen in this manner leads to x along the shortest path. In order to keep the distance estimates up to date, each node monitors the cost of its outgoing links and periodically broadcasts to all of its neighbors its current estimate of the shortest distance to every other node in the network. The distance vector which is periodically broadcasted contains one entry for each node in the network which includes the distance from the advertising node to the destination. The distance vector algorithm described above is a classical Distributed Bellman-Ford (DBF) algorithm.

DSDV is a distance vector algorithm which uses sequence numbers originated and updated by the destination, to avoid the looping problem caused by stale routing information. In DSDV, each node maintains a routing table which is constantly and periodically updated (not on-demand) and advertised to each of the node's current neighbors. Each entry in the routing table has the last known destination sequence number. Each node periodically transmits updates, and it does so immediately when significant new information is available. The data broadcasted by each node will contain its new sequence number and the following information for each new route: the destination's address, the number of hops to reach the destination and the sequence number of the information received regarding that destination, as originally stamped by the destination. No assumptions about mobile hosts maintaining any sort of time synchronization or about the phase relationship of the update periods between the mobile nodes are made. Following the traditional distance-vector routing algorithms, these update packets contain information about which nodes are accessible from each node and the number of hops necessary to reach them. Routes with more recent sequence numbers are always the preferred basis for forwarding decisions. Of the paths with the same sequence number, those with the smallest metric (number of hops to the destination) will be used. The addresses stored in the route tables will correspond to the layer at which the DSDV protocol is operated. Operation at layer 3 will use network layer addresses for the next hop and destination addresses, and operation at layer 2 will use layer-2 MAC addresses.

2.5 Other Approaches

In addition to the standard routing protocols discussed above, there exist other protocols which use different approaches. The following are some of these protocols [29].

Location-Assisted Routing: This approach improves route discovery and maintenance with the use of localization information which is accomplished by keeping track of the position and velocity of the mobile node. Nodes not in that general direction can be excluded from the route discovery and maintenance process to reduce bandwidth consumption and control message communication overhead.

Fisheye Routing: This is a form of routing which has nodes keeping track of more topology data for closer nodes. It is similar to the Zone Routing Protocol strategy, but with blurred boundaries between zones. This strategy can be very useful to increase the scalability of routing protocols.

Cedar (Core Extraction Distributed Ad Hoc Routing): The strategy in this type of routing algorithm is to increase scalability by creating and maintaining a backbone for communication of route requests to avoid broadcasting such information on a network-wide basis. The difficulty is in managing these backbone nodes, which can move relative to each other. Wu and Li in [36] provide a good algorithm for constructing a core which is a connected dominating set. Further research in this area is needed.

3 QoS Routing Protocols: Models and Classification

In this section, the different QoS models used in literature are presented. This is followed by a classification of the current QoS routing protocols according to the best effort routing protocol they extend as well as the model and environment they assume, and the communication layer within which they operate.

3.1 QoS Models in MANETs

Depending on the application involved, the QoS constraints could be available bandwidth, end-to-end delay, delay variation (jitter), probability of packet loss, and so on. This kind of demand puts more pressure on the network and the routing protocols which are used to support the communications. Establishing multi-hop routes between nodes is not sufficient in this case. The discovered routes can only be considered if they provide guarantees of the QoS parameters, such as bandwidth required by the application. Let $m(u, v)$ be the performance metric for the link (u, v) connecting node u to node v, and path $(u, u_1, u_2, ..., u_k, v)$ a sequence

of links for the path from u to v. Three types of constraints on the path can be identified [6][33]:

1. $Additive\ constraints$: A constraint is additive if
   $m(u, v)$ = $m(u, u_1)$ + $m(u_1, u_2)$ + ... + $m(u_k, v)$.
   For example, the end-to-end delay (u, v) is an additive constraint because it consists of the summation of delays for each link along the path.

2. $Multiplicative\ contraint$: A constraint is multiplicative if
   $m(u, v)$ = $m(u, u_1) \times m(u_1, u_2) \times ... \times m(u_k, v)$.
   The probability of a packet $prob(u, v)$, sent from a node u to reach a node v, is multiplicative, because it is the product of individual probabilities along the path.

3. $Concave\ constraint$: A constraint is concave if
   $m(u, v) = min\{m(u, u_1)$,$m(u_1, u_2)$,...,$m(u_k, v)\}$.
   The bandwidth $bw(u, v)$ requirement for a path between node u and v is concave. This is due to the fact that it consists of the minimum bandwidth between the links along the path.

Wang and Hou [33] provide a list of twelve combinations with multiple constraints. It has been proven in [35] that any multiple constraints with two or more type 1 and/or type 2 constraints are NP-complete; otherwise, they are tractable. Approximation methods exist for QoS constraints that are NP-complete. $Sequential\ filtering$ is a commonly used approach, where multiple paths between two nodes u and v that satisfy a single metric first (like bw) are found, then a subset of these paths is eliminated by optimizing over a second metric (like end-to-end delay), and so on.

In MANETs, node mobility often results in frequent topology changes, which presents a significant challenge when designing QoS routing protocols. High node mobility can make satisfying QoS requirements unreachable. Consequently, it is required that the network be $combinatorically\ stable$ in order to achieve QoS support [2]. This means that the changes in network topology must be slow enough within a particular time window to allow the topology updates to propagate successfully as required in the network.

QoS support of MANETs requires availability of network state. However, due to mobility and constant topology changes, the cost of maintenance of the network state is expensive especially in large networks. In [4] the *imprecise network state model* is introduced. It provides a cost-effective method for providing QoS support based on imprecise network information. The majority of QoS routing protocols are reservation-based. Probe messages are sent through the network from

the source to the destination in order to discover and reserve paths which satisfy a given QoS requirement. Due to the dynamic nature of the network, reserved QoS paths must be reaffirmed periodically by sending special control packets, called $refreshers$, along the path. Another approach, called $soft\ state$, relies on periodic time out at each node for path maintenance.

In addition, due to the difficulty of QoS support in the inherently dynamic environment of MANETs, some more "compromising principles" have been presented; $Soft\ QoS$ and $QoS\ adaptation$. Soft QoS [9] indicates that there may be transient periods of time during which the QoS specifications are not honored. However, the QoS satisfaction is quantified by the total disruption time over the total connection time. This ratio must be above a specified threshold in order to fulfill the QoS requirements. In the $fixed-level\ QoS$ approach, the reservation is defined in an n-dimensional space where the coordinates define the characteristics of the service [27]. On the other hand, QoS adaptation introduces the concept of $dynamic\ QoS$, where a range of QoS values, rather than a single point, is allowed to be specified by the application. This must be done through appropriate, flexible, and simple user interface which effectively maps the perceptual parameters into QoS constraints. The use of dynamic QoS provides more flexibility to the system and gives the network the ability to adjust the allocation according to the current availability of the required resources. The higher networking layers can then adapt to these changes. This adaptation can be achieved in different ways at the different layers of the architecture. The physical layer, for example, can adjust the transmission power to react to more frequent bit errors. The link layer can incorporate more error control (detection and correction) codes as well as automatic repeat requests (ARQ) in reaction to changes in link error rates.

At the other end of the OSI stack, namely the application layer (multimedia video conferencing for example), different compression techniques with varying compression ratios can be employed to adapt the application to the changes in bandwidth, delay, and error rates without drastically compromising the perceived audio and video quality. As more resources become available, the quality of the presentation can then be adjusted to take advantage of the added resources. In addition to compression algorithms, other techniques are being investigated at this level including layered encoding, rate shaping, adaptive error control, and bandwidth smoothing.

It is important at this point to state that the QoS model defines the general approach, goals, and framework for providing QoS support in a network. It does not specify a particular protocol, design, or implementation details. Providing QoS support is done at each of the layers of the OSI model starting from the application layer and ending with the physical layer. Various protocols and specifications such as QoS user interface, routing, signalling, resource reservation, and error checking,

measuring, and correcting must work and coordinate together in a collaborative and complementary fashion in order to satisfy the QoS requirements of the underlying applications. In this chapter, we focus on QoS routing, which is one of the most critical components in providing QoS support in MANETs.

3.2 QoS Routing Protocols

In order to support applications with quality of service requirements, such as multimedia, different QoS routing protocols have been developed. Of the quality of service parameters that are required by these applications that were mentioned earlier, minimum bandwidth is the most common. To support such requirement, the application layer of the source node sends a request to the lower layers (in the OSI model) with a specific destination and an amount of bandwidth which is required in order to satisfy the communication needs of that particular session. Depending on the communication requirements used, this desired amount of bandwidth is represented in different ways. In the synchronous Time Division Multiple Access (TDMA) environment, which is described later in this chapter, the bandwidth is represented by the number of slots needed to be reserved in the TDMA frame. In order for the route to satisfy this requirement, each of its links must reserve that number of slots for this particular session. When the session is ended, the reserved slots are freed and allowed to be reserved for other sessions. On the other hand, in the asynchronous environment, the slot size is variable and the amount of bandwidth reserved on particular links of the path is represented by a duration and a start time within the super frame [19][21][26][32]. The location of the slots within the frames of the different links of a path have direct affect over the total end-to-end transmission delay of data for a particular source-destination pair during a session. Different algorithms which have been proposed which choose the start times and durations of the slots according to different policies. The policies can be minimize end-to-end delay, maximize the probability of success in applying the reservation algorithm, or compromise between the two objectives.

QoS routing protocols for MANETs can be classified into different categories depending on the best effort routing algorithm which they extend or most closely resemble (DSR, AODV, DSDV, TORA, etc.) Though most of the the QoS routing protocols are designed to operate within the network layer, some of the implementations go below the network layer into the MAC layer. In the following sections, different QoS routing protocols are presented.

4 Sample QoS Routing Protocols

The following samples of QoS routing protocols are presented grouped according to which best effort routing protocol they extend or to which they are most closely related. As each of the protocols is discussed, the following characteristics will be identified: (1) the network layer within which it resides. (2) The QoS model it assumes (TDMA, CDMA-over-TDMA, etc.) (3) The networking environment within which it is designed to operate (synchronous or asynchronous).

4.1 Extensions of DSR

Many of the QoS routing algorithms represent extensions of DSR, which is a popular on-demand protocol for MANETs. Since this category has a larger share of QoS protocols than its counterparts, both it and its associated protocols are presented with more detail than the others. Some of these protocols are shown in table 1 and are discussed below.

The QoS routing algorithm in [17] by Liao et al. is an extension of the DSR protocol. It is on-demand and can operate in a single channel/code or multiple channel/code environment. In the single channel case, 1-hop neighbor nodes (nodes that are within range of each other) transmit and receive on the same channel frequency in frequency division multiplexing (FDM) networks, or code in the code division multiplexing (CDM) networks. The implementation of the protocol in [17], assumes a TDMA synchronous networking environment (single channel mode). In this network, communication between nodes is done using a synchronous TDMA frame. The TDMA frame is composed of a control phase and a data phase

[24]. Figure 3 shows the TDMA frame structure for a TDMA network (or a TDMA cluster) of N nodes. Each node in the network has a designated control time slot (control slots 1 through N in this example), which it uses to transmit its control information, but, the nodes in the network must compete for use of the data time slots (data slots 1 through M in this example) in the data phase of the frame.

As mentioned earlier, the TDMA environment is a single channel model. This model is generally practical and less expensive because only a relatively simple transmission mechanism and antenna design is needed. However, this model imposes on the designer the constraint of the hidden terminal and exposed terminal problems. The routing protocol must account for these problems and have appropriate mechanisms to avoid hidden terminal interference on one hand, and maximize channel re-use by taking advantage of the exposed terminal transmissions on the other hand. Consider the example in figure 1. A $hidden\ terminal$ problem in a wireless environment is created when two nodes which are out of range of each other, B and C for example, transmit to a third node A, which can hear them

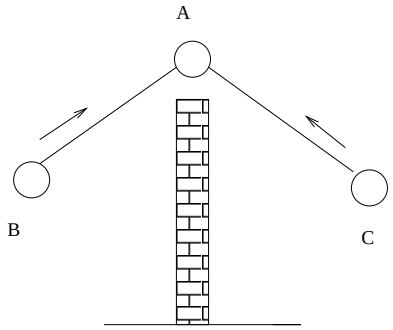

Figure 1: The hidden terminal problem creating a collision at node A.

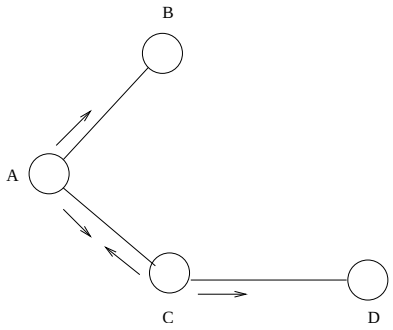

Figure 2: An exposed terminal is created at node A. Node A is exposed to node C's transmission to node D, but can still transmit to node B.

both. This creates a collision of the two transmissions at the "middle node" A. An $exposed\ terminal$ is illustrated in figure 2. It is created when a node A, is within range of and between two other nodes B and C, which are out of range of each other. When A wants to transmit to one of them, node B for example, the other node, C in this case, is still able to transmit to a fourth node, D which is in C's range (but out of the range of node A). Here A is an exposed terminal to C but can still transmit to B.

Liao and Tseng [17] specify three rules which must be satisfied for proper slot allocation at a particular node. These rules are in place in order to prevent transmission collisions that are due to the nature of the wireless medium, and to avoid the hidden terminal problem. They state that a time slot t is considered free to be allocated to send data from a node x to a node y if the following conditions are true:

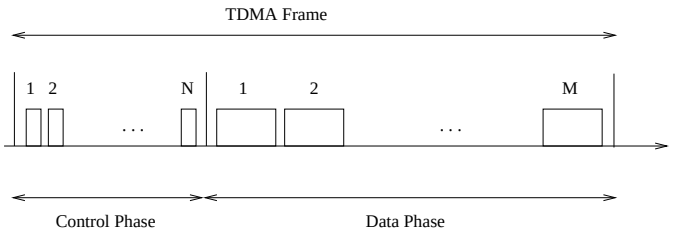

Figure 3: The structure of a TDMA frame for a network of N nodes and M data slots per frame. Each node has a fixed control slot. Nodes compete over the use of data slots.

1. Slot t is not scheduled for receiving or transmitting in neither node x nor y.
2. Slot t is not scheduled for receiving in any node z that is a 1-hop neighbor of x.
3. Slot t is not scheduled for sending in any node z that is a 1-hop neighbor of y.

Race condition due to multiple reservations at intermediate nodes. The protocol in [17], like many of the other QoS routing protocols [10][11][18][17], do not account for the race conditions which can become more significant with increased node mobility, network density and higher traffic loads. These race conditions can arise when multiple path reservations are being processed at intermediate nodes at the same time. A solution for such race conditions is presented in [14]. The following is a brief description of the race conditions and the solution provided. Consider the example shown in Figure 4. Here, two QoS path reservations are taking place simultaneously. A path is being reserved from node A to C (source node A sends the QoS request message QREQ1 to propagate to destination node C) and another is sent from node D to E (source node D sends the QoS request message QREQ2 to propagate to destination node E). The two requests pass through a common intermediate node B. When node B receives QREQ1 (with say b slots required) from node A to node C, it allocates b slots and forwards the request. Let slot t be among these allocated slots. Before B receives the reply message, QREP1, which would confirm the QoS path reservation from node C to A and reserve the allocated slots, it is possible the another request, QREQ2, can arrive at node B. QREQ2 from node D requests to reserve a path from node D to node E passing through node B. In the algorithm in [17], node B would potentially go ahead and allocate one or more of the same slots, including slot t in this example, for the request, QREQ2, for the path from D to E. This constitutes a double allocation of the same slots to two

different QoS paths. When the reply message, QREP1, arrives at B to confirm the QoS path reservation from C to A, node B will go ahead and confirm these slots, including slot t, and mark them as reserved in its ST and RT tables. Later, when the other reply message, QREP2, arrives at node B to confirm the QoS path from D to E, node B will potentially again reserve the same slots, including slot t in this example, for the second QoS path. Therefore, due to this race condition, the same slot t was reserved for two different QoS paths. This would create a conflict when the source nodes start using these reserved QoS paths to send data.

The conflict arises when the packets are transmitted from A to C and D to E simultaneously, and two data packets from two different paths arrive at node B. In this case, node B must decide which data packet it will actually send. The other data packet will be dropped. In this case, node B can, if the protocol requires, inform the other source of this error condition, or the source would simply time out the request. The corresponding source must then start the process of trying to reserve a new QoS path all over again. This leads to a decline in the throughput. In [14] an algorithm is proposed to fix this problem which is called the "race condition due to multiple reservations at an intermediate node".

Parallel race condition. The paper in [14] presents another possibility of a race condition which is due to a $parallel\ reservation\ problem$. An example of this race condition is shown in Figure 5. In this case, we have two parallel paths, ABCD and EFGH, being reserved. Two or more of the intermediate nodes belonging to the two parallel paths are 1-hop neighbors. In this case, node B, which belongs to the first path, and node F, which belongs to the other path are 1-hop neighbors. This is indicated in the figure using the dashed lines. The same relationship exists between nodes C and G. When QREQ1 is propagating from node A to D, the slots are allocated at the intermediate nodes. However, if the slot allocation information is not maintained by the nodes, say node B here, but only placed in the QREQ1 message, then no memory of this allocation is kept by the node, as is the case in [17]. This can cause another type of race condition, which we call the parallel reservation problem. This problem arises if, before QREQ1 propagates and is confirmed, the same process occurs with QREQ2 and node F allocates slots for the other QoS path and does not take into consideration the allocation of slots for QREQ1 at node B.

If both QREQ messages are successful in reserving their corresponding paths, a potential problem exists because the slot allocations at nodes B and F can be violating the slot allocation conditions mentioned earlier in this chapter. Nodes B and F each did the allocation based on information which did not consider the other 1-hop neighbor node's slot allocation for the corresponding parallel path being reserved. Again, if the two parallel paths are reserved successfully and data transmission is started along these paths, collisions will occur at the 1-hop neighbors belonging to

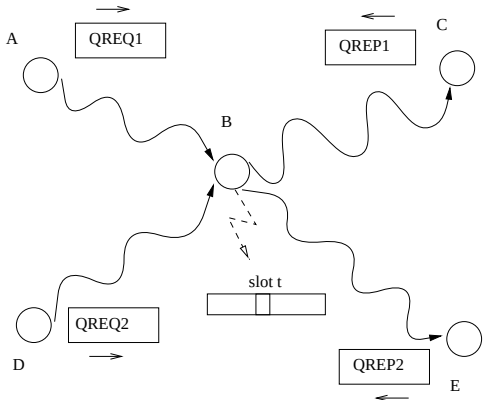

Figure 4: Multiple QoS path reservation competition. Two different QoS paths A..B..C and D..B..E are being reserved simultaneously, and they both pass through node B.

the different parallel paths. In this example, nodes B and F would experience this collision in their transmissions. A similar situation can occur between any 1-hop neighbors belonging to the two parallel paths, for example, between nodes C and G of the same figure.

It is important to note that this parallel reservation problem can occur in any situation where the two paths have 1-hop neighbors, with each belonging to the other path. This would also be the case in the example presented in Figure 6, where nodes B and E are 1-hop neighbors who belong to two different QoS paths. In [14], an algorithm which fixes this *parallel reservation problem* is proposed.

A more detailed example of the multiple QoS path reservation competition is shown in Figure 7. Node A wants to request a QoS path to node C with $b = 3$ (i.e. the required bandwidth is 3 slots). Node A sends a QoS request, QREQ1, to reserve the path. The QREQ message travels through the nodes on its way to C and arrives at node B. We see that node B has nodes F and G as 1-hop neighbors, and node G has node B and H as 1-hop neighbors. Node B will now try to allocate slots for this arriving QREQ1 message to send to each of its 1-hop neighbors, if there are b slots available to send from itself to this neighbor. It will calculate the number of slots available to each of those neighbors and will place those neighbors along with the allocated slots in the next hop list (NH). Node B will then include the next hop list (NH) in the QREQ1 message before it broadcasts (forwards) it.

Let's consider the process of calculating the number of slots available to send from node B to its 1-hop neighbor, node G. Node B has slot allocation information for itself and for all of its 1-hop and 2-hop neighbors including node G. Node B

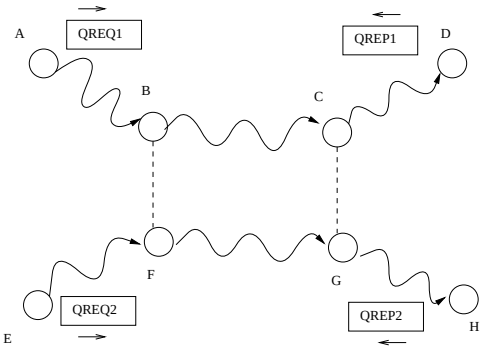

Figure 5: Parallel reservation problem. Nodes B and F (similarly, nodes C and G) are 1-hop neighbors which belong to two different QoS paths ,A..B..C..D and E..F..G..F, that are being reserved simultaneously.

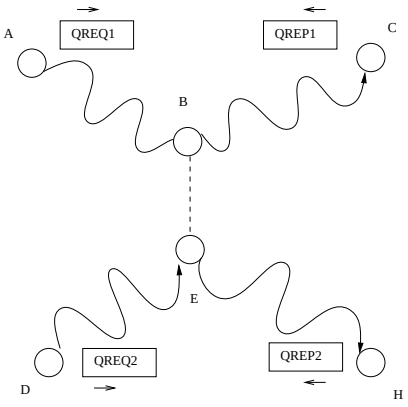

Figure 6: Parallel reservation problem. Nodes B and E are 1-hop neighbors which belong to two different QoS paths, A..B..C and D..E..F, that are being reserved simultaneously.

realizes that it cannot allocate slots 2 and 5, because they are scheduled by node B itself, to send and receive (slot selection rule 1). It also realizes that it cannot allocate slots 3 and 4, because they are scheduled to send and receive in node G (slot allocation rule 1). It cannot use slot 1 because it is scheduled to receive in one of its 1-hop neighbors, node F (slot allocation rule 2). Note that this is due to the hidden terminal problem; if node B sends to G using slot 1, this will cause a collision at node F which is using slot 1 to receive as well. Furthermore, node B cannot use slot 6, because it is scheduled to send in node H, which is a 1-hop neighbor of the node it intends to send to, node G (slot allocation rule 3). Note that this is another example of the hidden terminal problem, because if node B sends to node G using slot 6, it will cause a collision at node G. However, node B can use slot 7 to send to node G even though it is scheduled to send in node F. This is the exposed terminal problem. In fact, it would be more desirable for node B to allocate this slot to send to node G; this would increase channel reuse, a desired goal in wireless communications. Therefore, this leaves slots 7 through 12 which are free to send from node B to node G in this example.

After the calculation above, node B allocates slots 7, 8, and 9 to send from itself to G. It includes G in its next hop list NH along with the list of the slots 7, 8, and 9. It then broadcasts the QREQ1 message. In [17], node B does not keep track of this allocation which is only remembered in the forwarded QREQ1 message. So, until node B receives the corresponding QREP1 message which will be propagated from the destination C, slots 7, 8, and 9 in node B will remain $free$. They will only change status from $free$ to $reserved$ when and if the corresponding QREP1 message arrives from node C on its way to node A to confirm the QoS path A..FBG..C slot reservations. This poses no problem so long as no other requests arrive at node B during the period between forwarding QREQ1 and receiving the reply message QREP1. However, consider a situation where another request, QREQ2, arrives at node B from a source node D trying to reserve a QoS path from itself to node E with b=3 (i.e. the required bandwidth is 3 slots). Node B in this case will look at its slot status tables and will see no allocation for slots 7 through 12. It will then proceed to allocate some of these slots for this newly requested path. If the corresponding slot allocation procedure allocates slots 7, 8 and 9 for this new path and includes them in the next hop list, NH, then Node B will broadcast (forward) QREQ2 to node I which is on the path to node E. When QREP1 arrives at node B, it will change the status of slots 7, 8 and 9 to reserved. Afterwards, QREP2 will arrive at node B from node E on its way to node D. Node B will then have the problem of double allocation of slots 7, 8 and 9. In [17] the slots are reserved again (double reservation) for the second path. This will lead to a conflict at node B when data transmission using the two different paths starts. This is a multiple reservation problem due to a race condition at node B.

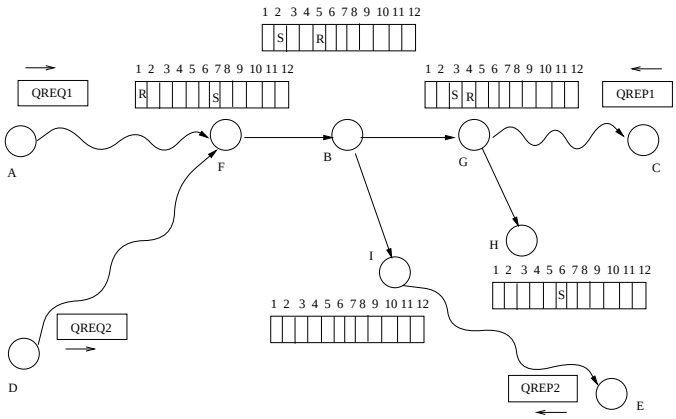

Figure 7: Multiple QoS path reservation competition. Two different QoS paths A..FBG..C and D..BI..E are being reserved simultaneously and they both pass through node B. The figure shows the slot reservation status before allocation to send from node B to G for QoS path requested by QREQ1. R: scheduled to receive. S: scheduled to send. empty: not scheduled to receive or send.

A similar example can be shown for the parallel reservation problem. This was described in Figure 6 where node B would select the slots to forward QREQ1 by considering only the status of the slots in node E prior to the allocation done by node E for the slots for QREQ2. When QREP1 returns to node B and QREP2 returns to node E, they both reserve the allocated slots. These slot selections can be in violation of the slot allocation rules and result in collisions when data transfer using the two different QoS paths begin.

In [14] a protocol is proposed to solve the race conditions described earlier and enhance network performance, especially in situations of increased node mobility, increased node density and higher traffic loads. The protocol uses a more conservative strategy. This strategy is implemented using the following features: (1) Three states for each slot: $reserved$, $allocated$, $free$. The three states are defined in the following manner: $Free$: not yet allocated or reserved. $Allocated$: in process of being reserved, but not yet confirmed. This means that the slot is allocated by a QREQ message but the corresponding QREP message has not yet arrived to confirm the reservation. $Reserved$: reservation is confirmed and the slot can be used for data transmission. (2) Discrimination between allocated (not yet reserved) and free slot status to alleviate the racing condition. (3) Wait-before-reject at an intermediate node with three conditions to alleviate the multiple reservation at intermediate node problem. ($conditon$ 1: all required slots are available, $condition$ 2:

not-now-but-wait, and $condition\ 3$: immediate drop or reject of QREQ). (4) TTL timer for allocated and reserved slots. (5) TTL timers for maximum total QREQ propagation delay allowed, and for maximum total QREQ/QREP delay allowed (i.e. maximum QoS path acquisition time). More discussions and details about the proposed protocol is presented in [14]

In [18], Liao et al. present a multi-path QoS routing protocol which is also an extension of DSR. The protocol is designed to work in a CDMA-over-TDMA environment, but can be easily extended to other types of networks. The protocol enables a source node to search for a multi-path QoS route to a particular destination satisfying certain bandwidth requirements. A number of tickets are distributed from the source. The tickets can be further partitioned in to sub-tickets to search for a satisfactory multi-path. This protocol provides a higher success rate for finding a QoS path satisfying the required bandwidth requirements when the bandwidth is very limited. If a link along the path does not have the entire required bandwidth, the path does not immediately fail to be reserved; the protocol still searches for the possibility that a multi-link path exists from that node to intermediate nodes or to the destination whose aggregate bandwidth is equal to the required one. This increases the success rate of reserving the required path between the source and the destination especially in situations where the bandwidth is scarce and the network traffic is high. The number of tickets issued by the source can affect the performance and can be empirically adjusted. When the network bandwidth is sufficient, this protocol provides a performance similar to protocols which find a uni-path between the source and the destination.

In [37], Zhu et al. present a Five-Phase Reservation Protocol (FPRP) for QoS support in synchronous TDMA-based MANETs. FPRP perfoms the tasks of channel access and node broadcast scheduling simultaneously. It uses a contention-based mechanism for nodes to reserve TDMA slots. The protocol takes into consideration the hidden terminal interference in the reservation process. Reservation is made through a localized conversation between nodes in a 2-hop neighborhood. Due to its fully-distributed nature, it is scalable.

4.2 Extensions of AODV

The protocol presented in [10] and in [11] by Gerasimov et al., which is named QoS-AODV, is an extension of the well known AODV protocol. It is on-demand and designed to work in a TDMA network. This protocol combines information from both the network and data link layer. Unlike other protocols which make path bandwidth calculations only after paths to the destination have been discovered [5][10][13][23][24], QoS-AODV incorporates path finding with the bandwidth reservation mechanism. QoS-AODV is fully aware of the bandwidth re-

source availability by coupling together routing and MAC TDMA layers. As described earlier, the nodes compete for the slots contained in the data phase of the TDMA frame. In order for the source node to send data to a destination node, it must establish a virtual circuit (VC) connection with that destination. The VC establishment process includes route discovery, path bandwidth calculation and bandwidth reservation (data-phase-slot reservation) components. Each node keeps a $schedule$ which contains information about both its own and its neighbor's time slots that are used for sending and receiving. A schedule is defined as a sequence of 1's and 0's where a number is the order of the corresponding slot in the data phase of the TDMA frame. The paper in [10] includes the algorithm used by each node to determine which slots are available to send to and receive from its neighbor, and to calculate link bandwidth scheduling from itself to each of its neighbors. The link bandwidth information is used in the calculation of the path bandwidth schedules to source and destination nodes. Modified AODV HELLO messages are used which include slot scheduling information. The HELLO messages are sent either periodically or when link bandwidth information is changed.

In QoS-AODV, path discovery is done in the following manner. A source node that wants to send data to a particular destination determines if it has enough link bandwidth available to any of its neighbors. If it does not, it then denies the request initiated by its application layer. Otherwise, it creates a routing table entry for the requested application call ID and the destination address. Note that, in QoS-AODV, there is an entry in the routing table for each application call ID/source/destination triple instead of one per source/destination tuple as in the original AODV. The source node then sends the reservation request message, RREQ, which contains call ID and number of slots required for reservation, in addition to the standard AODV information.

When an intermediate node receives a RREQ message, it checks whether it already has an entry in its routing table corresponding to the received application call ID. The node then calculates the path bandwidth schedules using algorithms similar to ones presented in [24]. If the calculated path bandwidth to the source is insufficient, then the node does not forward the RREQ message. Otherwise, the intermediate node augments the RREQ message with path and link bandwidth parameters and broadcasts it further. The link bandwidth between two nodes is calculated as the intersection of their free slot schedules. The send link bandwidth (say of a link AB at a node A) is defined as the intersection of the free send slot schedule of the sender node (A) and the free receive slot schedule of the receiver node (B). The receive link bandwidth (say of a link AB at a node A) is defined as the intersection of the free receive slot schedule of the receiver node (A) and the free send slot schedule of the sender node (B). In addition to information corresponding to the original AODV protocol, the route table entry contains the addresses of three

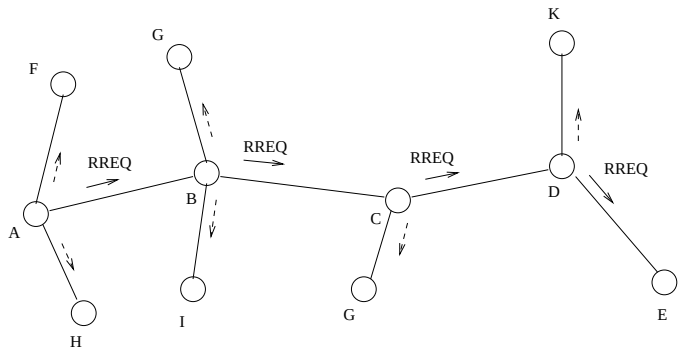

Figure 8: QoS-AODV - Propagation of RREQ message from source node A to destination node E. The RREQ message is forwarded only if the required bandwidth is available at each link. At each intermediate node, it is augmented with QoS bandwidth information

nodes along the path to the source, and link and path bandwidth schedules between those nodes. This information is needed in order to allocate slots that do not cause interference according to the slot allocation rules discussed before including the hidden terminal problem considerations.

An example of QoS-AODV route discovery is shown in Figure 8. A source node A needs to create a new VC to send data to a destination node E. Node A broadcasts an RREQ message which contains the call ID and the number of slots required for the QoS path. Upon receiving the RREQ message, node B, knowing that node A is one of its neighbors, determines that the available path bandwidth from A to B is equal to the receive link bandwidth from A to B. Path bandwidth AB is calculated as a portion of the receive link bandwidth AB. Node B then augments the RREQ message with the calculated link bandwidth AB and the address of A and rebroadcasts the RREQ message to all of its neighbors. When node C receives the propagated RREQ message from B, it knows that A is B's neighbor. Therefore it calculates the available path bandwidth using the AB and BC receive link bandwidth to avoid any interference conflicts including the hidden terminal problem. If the path bandwidths AB and BC contain the required number of slots, then C augments the RREQ message with the address of A and B, the receive link bandwidth BC, and the path bandwidth AB before it rebroadcasts it. Node D then receives the RREQ message; it calculates the path bandwidth to A using the link bandwidths AB, BC, and CD. If the calculated path bandwidth is sufficient, then D rebroadcasts the RREQ message after adding the address of C and B along with the receive link bandwidth CD and path bandwidth BC. When the destination node

E receives the message, it uses the same algorithms to determine the path bandwidth scheduling CD and DE. Once the destination node E determines that there is enough bandwidth to the source node, it starts the reservation process by creating a reservation instance. The reservation parameters stored at each node along the VC for that VC ID include: (1) source and destination ID's, (2) application call ID, (3) next hop, previous hop and next hop bandwidth scheduling, and (4) reservation status. The destination node (E in this example) reserves MAC receive slots corresponding to previous hop bandwidth scheduling and composes a reservation message, RSV, which is a new message added to the AODV protocol. The RSV message is propagated back to the source, A in this example, by the intermediate nodes (B, C, and D in this example) which reserve corresponding MAC receive and send slots. When the source node receives the RSV message, it informs its application layer of the establishment of a VC connection to the destination. The algorithm also defines an unreserved message, $URSV$ which can be used to release slot resources if multiple reservations at a particular node are done due to race conditions caused by node mobility.

In [38], Zhu et al. present an AODV-based QoS routing protocol. It is designed to function in the network layer. The protocol establishes QoS routes with reserved bandwidth on a per flow basis in a TDMA network. It incorporates an algorithm for calculating end-to-end bandwidth on a path. This algorithm is included in the path discovery mechanism of AODV to establish QoS routes. The protocol in [38] protects active routes with soft-state, i.e., a timer is associated with an active route at a node and is refreshed every time the route is used. If the route is not used within a certain amount of time and the timer expires, the corresponding entry in the routing table is deleted. The protocol defines the five possible states of a QoS route, which indicate whether the route exists, and if so, if it is processed but not established, set up and used to forward packets, broken at upstream of the node and is being repaired, or broken at downstream of the node and is being repaired. Transitions among these states is done by either receiving or transmitting a packet, or expiration of the timer associated with the state. The paper defines eleven conditions and operations associated with transitions among these states. The QoS routing protocol builds different QoS routes for individual flows even between the same source and destination. The protocol is also capable of restoring a route when it breaks due to some topological change, which allows it to cope more robustly with some degree of node mobility. The simulation in the paper shows that the protocol produces higher throughput and lower delay than the best-effort AODV protocol. It performs best in smaller networks with low node mobility.

4.3 Extensions of TORA

The protocol in [10] by Gerasimov et al., named QoS-TORA, is based on the link reversal best effort protocol TORA. It is designed to work in a TDMA network where the bandwidth of a link is measured in terms of slot reservations in the data phase of the TDMA frame. This protocol makes use of information in the network and MAC layers.

QoS-TORA operates in the following manner which is illustrated in Figure 9. When a source desires to communicate with a particular destination node, it checks whether it has a best-effort path to that destination. If there is no path, it tries to establish one by sending the original TORA QRY packet as indicated in [28]. When there is at least one best-effort path to the destination, the source node sends a QoS specific BQRY message, which contains the number of slots needed by the application along with the application ID. This BQRY message is propagated to the destination along the best-effort path. When the BQRY message reaches the destination, it checks whether it has enough slots available to receive. If it does, it then broadcasts a QoS-TORA specific UBW message, which contains the application ID, number of slots required and the source ID. Upon receiving the UBW message, each intermediate node checks whether it has already received a UBW message with the same application ID from the same neighbor and whether there is an existing path to that destination node with the required bandwidth. If the node does not have a QoS path available or the new path contains a smaller number of hops, the new path bandwidth is saved, which corresponds to the path that is going through the neighbor from which the UBW message was received. The intermediate node calculates the path bandwidth based on the information for three nodes along the path to the destination. This information is necessary to make sure that the slot allocation is done according the the rules stated earlier which provide interference-free operation. The source node waits for the reception of several UBW messages from its neighbors before it starts the reservation process. This allows the source node to choose which neighbor it wants to use for the establishment of the QoS path. This is in contrast to AODV-based QoS protocols which have single table entries for each destination. This gives QoS-TORA more flexibility to respond to link breakage due to node mobility. The simulation experiments presented in [10] show considerable improvements in the probability of being able to find an end-to-end QoS path. Simulation also shows that QoS-TORA provides higher throughput under higher mobility circumstances. This is due to the fact that when a VC breaks, unlike the case in AODV-based QoS protocols, the source node might have another neighbor to start reservations, so the path discovery procedure can be skipped.

In [7] Dharmaraju et al. present another TORA-based QoS routing protocol for

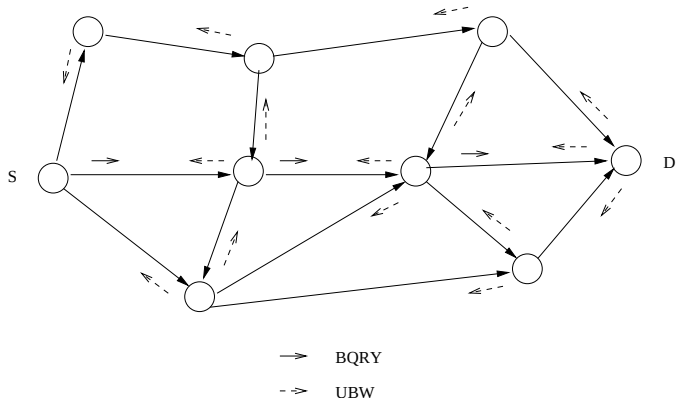

Figure 9: The DAG in the QoS-TORA protocol. The figure shows the propagation of the BQRY message from the source node, S, to the destination. The destination node, D, responds with the UBW message which contains the application ID, No. of slots, and source node ID.

MANETs called INORA (INSIGNIA [1][15][16] + TORA). INORA is a network layer QoS support mechanism that makes use of the INSIGNIA in-band signalling mechanism and the TORA routing protocol for MANETs. In INORA, QoS signalling is used to reserve and release resources, and set up, tear down and renegotiate flows in the network. These reservations can be either hard state or soft state. The latter is more desirable in MANETs due to their dynamic nature. The INORA protocol operates the signalling mechanism independently from the TORA routing protocol. This provides decoupling of the two mechanisms and there is no interaction between. TORA provides the route between the source and the destination of a flow. Then the signalling mechanism (INSIGNIA) establishes resources for the route provided by TORA. INORA tries to find paths in the network that can satisfy the desired QoS requirements. In INORA, INSIGNIA asks TORA for alternative routes when the current route is not able to meet the QoS requirements. The INORA scheme provides load-balancing in the network which aids in the performance of non-QoS flows. Future work will try to alleviate congestion in the wireless network by establishing QoS flows which avoid congested neighborhoods. The decoupling between the signalling and routing protocols allows for more flexibility in the design to incorporate load-balancing, congestion control, class-based admission control, and so on. This added flexibility comes at the price of more overhead when compared with other TORA-based QoS routing protocols which do not have the decoupling mechanism stated above.

In [12], Gupta et al. propose a framework for providing quality of service guarantees given limits on the rate of change in the topology. The protocol specifies changes which need to be incorporated in buffer requirements and play out times. The QoS support mechanisms reside in the network layer and constitute an extension of the INSIGNIA protocol mentioned earlier. The protocol also uses a soft state model (flows time out according to a soft state timer if not used to transmit data before timer expiration) which allows it to withstand link failures and route changes. The protocol relies on local action taken by a node detecting link failure to repair the path locally instead of generating an error message back to the source. The node is expected to buffer the received packets in the meantime before the path is repaired. The underlying protocol is assumed to be able to provide link failure information to the network layer. The IEEE 802.11 link layer protocol supports this operation. Route repair is done using a Route Repair Request (RRReq) packet which is sent to a limited area specified by a TTL (Time to Live in number of hops) field. Other nodes send a Route Repair Reply (RRRep) packet listing a path. The best path is used. The route restoration process and its ability to repair is limited by the rate of change of the topology. Otherwise, quality of service guarantees would not be possible and in such circumstances the application must be adaptive and able to live with the best-effort service.

4.4 Extensions of DSDV

In [26], Manoj et al. propose a MAC layer protocol named Real-time MAC (RTMAC) for MANETs, which provides a bandwidth reservation mechanism. The protocol is designed to work in an asynchronous environment. The protocol relies on the flexibility of placement of reservation slots (of variable start and finish times) in the super-frame. The protocol makes use of holes (short free slots in the super-frame which otherwise cannot be utilized). The authors provide simulation which compares the protocol performance with the MACA/PR protocol, proposed by Lin and Gerla [22], and show that RTMAC outperforms MACA/PR in call blocking ratio, average end-to-end delay, packet delivery ratio, and provides less effect of the presence of best-effort traffic on real time traffic. RTMAC is an extension of DSDV [30]. It is responsible for finding an end-to-end path that satisfies the QoS bandwidth requirements. Bandwidth reservation for Constant Bit Rate (CBR) traffic is provided by dividing the transmission time into successive super-frames. This scheme can also be extended to support Variable Bit Rate (VBR) traffic as well.

The main concept in this approach is the flexibility of slot placement in the super-frame. Each super-frame consists of a sequence of reservation-slots (resv-slot). The time duration of a resv-slot is twice the maximum propagation delay. Each session between source and destination nodes requires the reservation of a

block of consecutive resv-slots. A node must first reserve a set of resv-slots and a guard band to cushion the propagation delay (henceforth it is referred to as a connection-slot) on a super-frame and uses the same connection-slot to transmit in successive super-frames. A reservation table must be maintained by each node. The table contains information such as sender, receiver, and starting and ending time of the reservations that are active within its transmission range. This scheme is different from that of the TDMA environment because it requires no time synchronization (no need to maintain a global clock with the associated communication overhead) and uses a relative time for all reservation purposes [26]. Each node transmits its reservation table along with the route update packet of DSDV. The protocol includes the capability of a node to designate a specific connection-slot to be reserved for a particular connection, which gives the routing protocol the flexibility to position the connection slot. The protocol applies different schemes to reserve connection-slot (conn-slot). Different schemes can be used to allocate connection-slots such as first fit (reserve slot in the immediate freely available connection-slot), best fit (place connection-slots at a place that succeeds the connection-slot on which the node receives the real-time packets) and fair fit (reserve connection-slots in a way which creates free slots that can be used for best-effort traffic). A source node desiring to transmit data to a certain destination node checks the reservation information of its neighbors, finds free slots that can be reserved, and initiates the reservation process for those free slots.

In [19], Lin introduces the MACA/PR (Multiple Access Collision Avoidance with Piggy-back Reservation) protocol. It is an asynchronous network based on the collision avoidance MAC scheme used in the IEEE 802.11 standard. MACA/PR avoids collisions due to the hidden terminal problem by establishing an RTS-CTS (request to send - clear to send) dialogue, which can be used as a building block to eliminate hidden terminal interference. The key components of the MACA/PR architecture are: a MAC protocol for transmission of data packets, the reservation protocol for setting up real-time connections, and the QoS routing algorithm which is an extension of the best-effort table-driven DSDV protocol. Figure 10 shows the MACA/PR protocol's RTS-CTS-PKT-ACK...PKT-ACK sequence. A $CYCLE$ is defined to be the maximum interval allowed between two real-time packets. The sender schedules its next transmission after a time CYCLE following the current data packet. The intended receiver enters the reservation in its reservation table and confirms it in the ACK returned to the sender.

The protocol has two features: 'flexible' reservations within a CYCLE (as opposed to slotted reservation of TDMA schemes), which is defined below, and QoS loop-free routing. The first data packet in the multimedia stream makes the reservations along the path. Once the first data packet is accepted on a link, a transmission window is reserved on that link at appropriate time intervals for all the subsequent

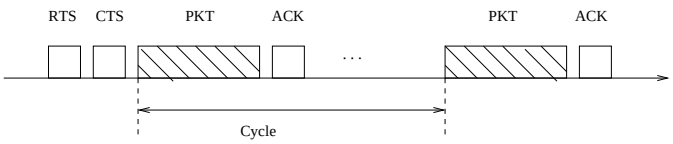

Figure 10: The MACA/PR protocol: RTS-CTS-PKT-ACK...PKT-ACK sequence. A $CYCLE$ is the maximum interval allowed between two real-time packets.

packets in the connection. The window is released if it is idle for a specified number of cycles. The RTS-CTS exchange is used in the first packet transmission to set up the reservation. The subsequent data packets do not require the RTS-CTS exchange and are done using only packet transmission followed by corresponding acknowledgements.

The QoS routing protocol must first find a path which satisfies the QoS resource requirements desired by the application. The resource setup protocol starts hop-by-hop negotiation and setup along the path. Each node periodically broadcasts to its neighbors the $\{$bandwidth, hop distance$\}$ pairs for the preferred paths (one per bandwidth value) to each destination. The number of preferred paths is the maximum number of slots (or packets) in a cycle. In the implementation in [19] each node keeps track of the bandwidth on the shortest path, and the maximum bandwidth (over all possible paths). A node drops real time packets with bandwidth requests which it cannot satisfy by the currently available path to the destination. If a link has no bandwidth or the bandwidth is below a predefined threshold, its weight is set to ∞. This will prevent any new connections from being established until an old connection releases some bandwidth. MACA/PR strikes a good compromise between the totally asynchronous, unstructured PRNET and the highly organized cluster TDMA.

5 Other QoS Routing Protocols and Related Issues

In addition to the categories of protocols mentioned above, there exist other protocols which are not direct extensions of DSR, AODV, TORA, and DSDV. In this section, some of these protocols are presented.

In [21], Lin and Gerla present a network architecture for multimedia. The architecture assumes a code division access scheme. Specifically, direct-sequence spread-spectrum for CDMA is used. The protocol operates at the level of the MAC layer. The nodes in the MANET are organized into $clusters$. All links in a cluster are assigned the same spread spectrum code. Any two nodes in a cluster are only

one hop away from each other. A round robin scheme is used to provide channel access to the nodes in a cluster with bandwidth reservation for real time traffic. A virtual circuit is allocated at call setup time between a source and a destination node. The protocol is designed to work in an asynchronous environment. The protocol uses piggy-back reservations with packet transmissions to reserve time slots dynamically for packets from active voice sources without conflict. The protocol uses CSMA-PR (Code Division Multiple Access with Piggy-back Reservation) to resolve the conflict situation which arises when some traffic sources are trying to access the channel at the same time. Because CDMA is used, all clusters can operate simultaneously.

In [32] by Sheu et al., a distributed bandwidth allocation/sharing/extension (DBASE) protocol is proposed. It supports multimedia traffic with CBR (constant bit rate) and VBR (variable bit rate) traffic over ad hoc WLAN. This protocol functions at the MAC layer. It is asynchronous and uses RTS-CTS-asynchronous data-ack exchanges for channel access. In DBASE, $rt(realtime) - stations$ can reserve and free channel resources dynamically. The bandwidth allocation procedure is based on a contention (and back-off) process that only occurs before the first successful access and a reservation process after the successful contention. When the rt-station leaves, the bandwidth is immediately released by DBASE. The proposed protocol is compliant with the IEEE 802.11 standard. The simulation presented in the paper shows an improvement over the conventional 802.11 standard in terms of high channel utilization, low access delay and small delay variation for real-time services.

In [34] by Wang et al., a QoS Routing protocol with Mobility Prediction (QRMP) is presented. QRMP uses mobility prediction and QoS requirements on bandwidth and delay to select the most stable path. The route setup phase consists of request and reply stages. In addition to the usual routing information such as node_id and sequence_number, all QoS requirements information and $node\_info$ including related information of the node, e.g. link delay, link bandwidth and interface velocity, are considered. QRMP reduces route setup time and control overhead as well as increases packet delivery ratio by selecting the most stable route based on mobility prediction. The latter is also used to reduce the update message frequency.

In [8], Dong et al. propose an on-demand Supernode-based Reverse Labeling (SRL) algorithm for QoS provisioning, specifically bandwidth and delay, in MANETs. The algorithm utilizes a hierarchical structure, which is formed by dynamically electing $super\ nodes$. The other nodes are $slave\ nodes$ and are always one hop away from their corresponding super node. Slave nodes regularly communicate with their super node through periodic HELLO messages. The authors provide algorithms to perform effective route discovery and local route information management: virtual route discovery, reverse link labeling and dynamic route

repairing. A node sends its QREQ message to its super node. A sequence of super nodes is considered a virtual route (VR). Delay requirement and accumulated delay fields are supported and can be used by delay sensitive applications. The simulation in [8] shows that SRL is efficient in terms of packet delivery ratio and average end-to-end delay. It also has reduced packet loss ratio and route request overhead caused by node mobility. A source node which needs to transmit information to a destination sends a route request message to its super-node. The request includes QoS requirements, which can be bandwidth and/or delay constraints depending on the needs of the application involved. The request propagates through intermediate super nodes to find the destination and its corresponding super node, which will then process the request in a manner similar to DSR [8]. Transmitted messages include Delay Requirement and Accumulated Delay fields for applications that have such requirements.

A summary of the classification of the protocols discussed in this chapter is presented in table 1. The table contains the following columns. First the QoS routing protocol is listed. Then, ”Net. Layer” parameter indicates the networking layer within which the protocol is designed to operate. The ”Syn./Asyn.” parameter indicates whether the protocol operates within a synchronous or asynchronous environment. The ”Comm. Mode” parameter indicates the communication network assumed such as TDMA, CDMA-over-TDMA, and so on. The ”BE Routing Prot.” parameter indicates the best effort routing protocol that is extended by or is most closely related to the corresponding QoS protocol. The ”Proact./React.” parameter indicates whether this QoS protocol is reactive (on-demand) or proactive (table-driven). Then the ”Comments” field contains additional information about the QoS protocol. There are other parameters which can also be considered such as a protocol being location assisted or not, which were not included in the table.

6 Conclusions and Future Research

In this chapter, we discussed the existing QoS routing protocols. The different approaches taken by researchers who are active in this area were discussed. The most popular best effort routing protocols in MANETs, such as DSR, AODV, DSDV and TORA, and the different QoS parameters were presented with a brief description of each. Classification of the existing QoS routing protocols was done according to different criteria such the related best effort routing protocol, the OSI layer, communication model, and synchronization mechanisms used.

Significant advances are constantly taking place to increase the capabilities and use of wireless devices. Communication between these devices will become an essential part of their growth. As applications including audio and video multimedia

are developed to support the growing services that these networked devices provide, the need for QoS guarantees to be given by lower layers of the network to the application layer will become an indispensable part of supporting communication. Many areas of research in this field provide considerable challenge and potential to enhance the growth and proliferation of MANETs and their applications. These areas include power consumption, resource availability, location management, inter-layer integration of QoS services, support for heterogenous MANETs, as well as robustness and security. Continued growth is expected in this area of research in order to develop, test and implement the critical building blocks to provide efficient and seamless communications in mobile ad hoc networks. QoS routing protocols will play an essential role in providing the required support mechanisms.

QoS Routing	Net. Layer	Syn./ Asyn	Comm. Mode	BE Routing	React./ Proact.	Comments
Gerasimov et al. [10]	net./ MAC	syn.	TDMA	AODV	react.	QoS-AODV.
Gerasimov et al. [11]	net./ MAC	syn.	TDMA	TORA	react.	QoS-TORA.
Ho et al. [13]	net.	syn.	TDMA	ODQoS	react.	ODQoS (On-demand QoS-based routing prot.)
Liao et al. [17]	net.	syn.	TDMA	DSR	react.	QREQ from source to dest. allocating slots. QREP from dest. to source reserves slots.
Liao et al. [18]	net.	syn.	C-o-T or FDMA	DSR	react.	Multi-path QoS (ticket-base) routing.
Manoj et al. [26]	MAC	asyn.	N/A	DSDV	proact.	Ext. of 802.11 DCF function.
Lin [19]	MAC	asyn.	N/A	DSDV	proact.	Flexible reservations within a CYCLE.
Lin et al. [21]	MAC	asyn.	C-o-TDM	N/A	react.	CDMA-over-TDMA. Each cluster has different code.
Lin et al. [25]	net.	syn.	C-o-T	DSDV	proact.	Destination does calc. of the path BW.
Lin [20]	net.	syn.	C-o-T	DSR	react.	RREQ packets to find paths and calc. BW.
Sheu et al. [32]	MAC	asyn.	N/A	Lower level	react.	Compliant with 802.11. RTS-CTS-Asyn. Data-ACK chan. access.
Wang et al. [34]	net.	syn.	N/A	QRMP (source r.)	react.	QoS routing with mobility prediction.
Dong et al. [8]	net.	gen.	gen.	SRL (DSR-like)	react.	Supernode-based Reverse Labeling Algorithm
Zhu et al. [38]	net.	syn.	TDMA	AODV	react.	BW calc. integrated with AODV prot.
Dharmaraju et al. [7]	net.	gen.	gen.	TORA	react.	INORA: Uses signalling done at higher level than routing
Gupta et al. [12]	net.	gen.	gen.	TORA	react.	Extension of INSIGNIA.
Zhu et al. [37]	net.	syn.	TDMA	FPRP (DSR-like)	react.	Five-phase reservation protocol.

Table 1. QoS Routing Algorithm Classification. Abbriviations: gen.: general which also indicates applicability to all cases of that classification (higher level); C-o-T: CDMA-over-TDMA; C-o-TDM: CDMA-over-TDM.

References

[1] G.-S. Ahn, A. T. Campbell, S.-B. Lee, and X. Zhang. Insignia. *Internet Draft, draft-ietf-manet-insignia-01.txt*, October 1999.

[2] S. Chakrabarti and A. Mishra. QoS issues in ad hoc wireless networks. *Communications Magazine, IEEE*, 39(2):142–148, February 2001.

[3] K. Chen, S. H. Shah, and K. Nahrstedt. Cross layer deisgn for data accessibility in mobile ad hoc networks. *J. Wireless Commun.*, 21:49–75, 2002.

[4] S. Chen. Routing support for providing guaranteed end-to-end quality-of-service. *http://www.cs.uiuc.edu/Dienst/UI/2.0/Describe/ncstrl.uiuc_cs/UIUCDCS-R-99-2090, UIUCDCS-R-99-2090, University of Illinois at Urbana-Champain*, July 1999.

[5] T.-W. Chen, J. T. Tsai, and M. Gerla. QoS routing performance in mulithop, multimedia, wireless networks. *IEEE 6th International Conference on Universal Personal Communications Record*, 2:557–561, October 1997.

[6] X. Chen and J. Wu. Multicasting techniques in mobile ad hoc networks. *Chapter 2, The Handbook of Ad Hoc Wireless Networks, edited by M. ILyas*, pages 2.1–2.16, 2003.

[7] D. Dharmaraju, A. Roy-Chowdhury, P. Hovareshti, and J. S. Baras. Inora-a unified signaling and routing mechanism for QoS support in mobile ad hoc networks. *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pages 86–93, August 2002.

[8] Y. Dong, T. Yang, D. Makrakis, and I. Lambadaris. Supernode-based reverse labeling algorithm: QoS support on mobile ad hoc networks. *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*, 3:1368–1373, May 2002.

[9] A. Veres et al. Supporting service differentiation in wireless packet networks using distributed control. *IEEE JSAC*, October 2001.

[10] I. Gerasimov and R. Simon. A bandwidth-reservation mechanism for on-demand ad hoc path finding. *IEEE/SCS 35th Annual Simulation Symposium, San Diego, CA*, pages 27–33, April 2002.

[11] I. Gerasimov and R. Simon. Performance analysis for ad hoc QoS routing protocols. *Mobility and Wireless Access Workshop, MobiWac 2002. International*, pages 87–94, 2002.

[12] A. Gupta and D. Sanghi. QoS support in mobile ad-hoc networks. *Personal Wireless Communications, 2000 IEEE International Conference on*, pages 340–344, December 2000.

[13] Y.-K. Ho and R.-S. Liu. On-demand QoS-based routing protocol for ad hoc mobile wireless networks. *Fifth IEEE Symposium on Computers and Communications, 2000. Proceedings. ISCC 2000*, pages 560–565, July 2000.

[14] I. Jawhar and J. Wu. A race-free bandwidth reservation protocol for QoS routing in mobile ad hoc networks. *Hawaii international conference on system sciences - HICSS-37, (to be held on: January 5-8, 2004), Big Island, Hawaii-paper submitted*, January 2004.

[15] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Campbell. Insignia: An ip-based quality of service framework for mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 60(4), April 2000.

[16] S.-B. Lee and A. T. Campbell. Insignia: In-band signaling support for QoS in mobile ad hoc networks. *proceedings of 5th international workshop on mobile multimedia communications (MoMuC, 98), Berlin*, October 1998.

[17] W.-H. Liao, Y.-C. Tseng, and K.-P. Shih. A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network. *Communications, ICC 2002. IEEE International Conference on*, 5:3186–3190, 2002.

[18] W.-H. Liao, Y.-C. Tseng, S.-L. Wang, and J.-P. Sheu. A multi-path QoS routing protocol in a wireless mobile ad hoc network. *IEEE International Conference on Networking*, 2:158–167, 2001.

[19] C. R. Lin. Multimedia transport in multihop wireless networks. *Communications, IEE Proceedings-*, 145(5):342–346, October 1998.

[20] C. R. Lin. Admission control in time-slotted multihop mobile networks. *Selected Areas in Communications, IEEE Journal on*, 19(10):1974–1983, October 2001.

[21] C. R. Lin and M. Gerla. A distributed control scheme in multi-hop packet radio networks for voice/data traffic support. *Communications, 1995. ICC 95 Seattle, Gateway to Globalization, 1995 IEEE International Conference on*, 2:1238–1242, June 1995.

[22] C. R. Lin and M. Gerla. Asynchronous multimedia multihop wireless networks. *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Com-*

puter and Communications Societies. Proceedings IEEE, 1:118–125, April 1997.

[23] C. R. Lin and C.-C. Liu. An on-demand QoS routing protocol for mobile ad hoc networks. *Conference on IEEE International Networks, (ICON 2000) Proceedings*, pages 160–164, Spetember 2000.

[24] C. R. Lin and J.-S. Liu. QoS routing in ad hoc wireless networks. *IEEE Journal on selected areas in communications*, 17(8):1426–1438, August 1999.

[25] H.-C. Lin and P.-C. Fung. Finding available bandwidth in multihop mobile wireless networks. *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, 2:912–916, 15-18 May 2000.

[26] B. S. Manoj and C. S. R. Murthy. Real-time traffic support for ad hoc wireless networks. *Networks, 2002. ICON 2002. 10th IEEE International Conference on*, pages 335–340, August 2002.

[27] P. Mohapatra, J. Li, and C. Gui. QoS in mobile ad hoc networks. *IEEE Wireless Communications*, pages 44–52, June 2003.

[28] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 3:1405–1413, April 1997.

[29] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesley, Upper Saddle River, NJ, USA, 2001.

[30] C. E. Perkins and P. Bhagawat. Highly dynamic destination-sequenced distance-vector (dsdv) for mobile computers. *Proc. of ACM SIGCOMM '94*, pages 234–244, August 1994.

[31] C. E. Perkins and E. M. Royer. Ad hoc on demand distance vector (aodv) routing. *Internet Draft*, August 1998.

[32] S.-T Sheu and T.-F. Sheu. Dbase: a distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless lan. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1558–1567, April 2001.

[33] B. Wang and J. C. Hou. Multicast routing and its QoS extension: problems, algorithms, and protocols. *Network, IEEE*, 14(1):22–36, January-February 2000.

[34] J. Wang, Y. Tang, S. Deng, and J. Chen. QoS routing with mobility prediction in manet. *Communications, Computers and signal Processing, 2001. PACRIM. 2001 IEEE Pacific Rim Conference on*, 2:357–360, August 2001.

[35] Z. Wang and J. Crowcroft. Qos routing for supporting resource reservation. *IEEE Journal on selected areas in communications*, 14:1228–1234, 1996.

[36] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. *Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 7–14, August 1999.

[37] C. Zhu and M. S. Corson. A five-phase reservation protocol (fprp) for mobile ad hoc networks. *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1:322–331, 29 March - 2 April 1998.

[38] C. Zhu and M. S. Corson. QoS routing for mobile ad hoc networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, 2:958–967, June 2002.