A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks

Lei Chen and Wendi B. Heinzelman, University of Rochester

Abstract

The explosive growth in the use of mobile devices coupled with users' desires for real-time applications has provided new challenges in the design of protocols for mobile ad hoc networks. Chief among these challenges to enabling real-time applications for mobile ad hoc networks is incorporating support for quality of service (QoS), such as meeting bandwidth or delay constraints. In particular, it is important that routing protocols incorporate QoS metrics in route finding and maintenance to support end-to-end QoS. This article extensively and exclusively studies the issues involved with QoS-aware routing and presents an overview and comparison of existing QoS-aware routing protocols. In addition, the open issues that must be addressed to fully support QoS-aware routing are discussed.

Mobile ad hoc networks (MANETs) are distinguished from other types of networks by their physical characteristics, organizational format, and dynamic topology:

- *Physical characteristics*: Wireless channels are inherently error-prone, due to effects such as multipath fading, interference, and shadowing; causing unpredictable link bandwidth and packet delay.
- *Organizational format*: The distributed nature of MANETs means that channel resources cannot be assigned in a predetermined way.
- *Dynamic topology*: As hosts in a MANET are mobile, links are created and destroyed in an unpredictable way. Therefore, the network status can change quickly, causing hosts to have imprecise knowledge of the current network state.

Because of device mobility in MANETs and the shared nature of the wireless medium, offering guaranteed quality of service (QoS), such as bandwidth, delay, delay jitter, and packet delivery ratio, is not practical. Therefore, *soft QoS* and *QoS adaptation* are proposed instead. Soft QoS implies that failure to meet QoS is allowed, for example, when routes break or the network becomes partitioned [1]. However, if a network changes too fast to propagate the topology status information, it is challenging to offer even soft QoS. Therefore, combinatorial stability — which means that given a specific time window topology changes occur sufficiently slowly to allow successful propagation of all topology updates as necessary [2] — must be met in order to provide QoS.

Certain applications, such as real-time applications that can optimize their performance based on feedback about network resource availability, can benefit from QoS adaptation. For example, layered coding allows enhanced layers of different quality levels to be transmitted, provided a minimum bandwidth is guaranteed for transmitting the base layer. By providing feedback to the application about available resources, the application can alter its coding strategy to provide the best quality for the current resource limitations.

Routing is used to set up and maintain routes between nodes to support data transmission. Early MANET routing protocols focused on finding a feasible route from a source to a destination, without any consideration for optimizing the utilization of network resources or for supporting specific application requirements. To support QoS, the essential problem is to find a route with sufficient available resources to meet the QoS constraints and possibly to incorporate optimizations, such as finding the lowest cost or most stable of the routes that meet the QoS constraints. Given these goals, the following are the basic design considerations for a QoS-aware routing protocol.

- *Resource estimation*: To offer a resource-guaranteed route, the key concept is to obtain information about the available resources from lower layers. This information helps in performing call admission and QoS adaptation. Most existing techniques focus on bandwidth and/or delay QoS constraints, and thus, the bandwidth available to a node or link and/or the delay must be estimated. In MANETs, hosts share the bandwidth with their neighbor hosts, and thus, the bandwidth available to a node varies and is dynamically affected by the traffic of its neighbors. Therefore, the two key problems in bandwidth estimation are: how exactly to estimate the available bandwidth and how frequently to estimate it. In general, the trade-off between the benefit from using resource estimation and the cost in terms of overhead and computing resources used for resource estimation is a key issue.
- *Route discovery*: There are two main approaches to routing in MANETs: reactive routing and proactive routing. Reactive routing reduces overhead at the expense of delay in finding a suitable route; whereas, the reverse is true for proactive routing. For QoS-aware routing, another issue is determining the combination of reduced latency and reduced overhead that is best for supporting QoS.
- *Resource reservation*: As previously stated, the bandwidth resources are shared by neighboring hosts in MANETs. Therefore, another challenging issue is how to allocate these shared resources, the type of resource reservation

0890-8044/07/$20.00 © 2007 IEEE

scheme, and the kind of call admission that should be used for setting up and maintaining QoS-aware routes.

- *Route maintenance*: The mobility of nodes in MANETs causes frequent topology changes in the network, making it difficult to meet the QoS constraints. Incorporating a fast route maintenance scheme into QoS-aware routing is the fourth design consideration. The typical approach to route maintenance, which entails waiting for the host to discover a route break, significantly affects the routing performance. Therefore, a prediction scheme or redundant routing is helpful to assist in route maintenance.
- *Route selection*: QoS-aware routing has more stringent requirements on route stability, because frequent route failures adversely affect the end-to-end QoS. Thus, in some sense the route with the largest available bandwidth is not the only consideration — other metrics such as route reliability and route length also should be considered when selecting a suitable route for a QoS-aware routing protocol.

Several routing protocols have beeen developed that support QoS in one or more of the following ways:

- Choosing routes with the largest available bandwidth (or minimum delay)
- Providing a call admission feature to deny route requests if insufficient bandwidth is available to support the request
- Providing feedback to the application about available bandwidth resources or route delay estimation

Several researchers [3–5] addressed the general problem of QoS in MANETs, providing overviews and insights on the work being done in this area. In our article, we extensively and exclusively study the challenges in supporting QoS at the network layer, as opposed to [3], which discusses the broad topic of QoS support in MANETs, covering multiple layers, and thus does not provide an in-depth look at the network layer and [4], which focuses on the medium access control (MAC) layer. Thus, the major new contributions of this article are the focus on and in-depth studying of the issues involved with QoS-aware routing and the overview and comparison of existing QoS-aware routing protocols.

In the next section, we provide high-level descriptions of several QoS-aware routing protocols. Following this, we present a comparison of these protocols and point out the open research issues in QoS-aware routing.

QoS-aware Routing Protocols

To facilitate a comparison among the different QoS-aware routing protocols, in the following sections we describe them according to the design constraints listed earlier, discussing how each protocol addresses:

- Bandwidth/delay estimation
- Route discovery
- Resource reservation
- Route maintenance
- Route selection, where appropriate

Core-Extraction Distributed Ad Hoc Routing

Core-extraction distributed ad hoc routing (CEDAR) [6] is a routing protocol that dynamically establishes a *core* set for route set up, QoS provisioning, routing data, and route maintenance. A greedy algorithm is used to proactively create an approximate minimum dominating set, whereby all hosts in the network are either members of the core or one-hop neighbors of core hosts. Only core hosts maintain local topology information, participate in the exchange of topology and available bandwidth information, and perform route discovery, route maintenance, and call admission on behalf of these nodes. Two assumptions are made in CEDAR:

- The MAC/link layer can estimate available link bandwidth
- Small-to-medium-size ad hoc networks that consist of tens to hundreds of nodes

While CEDAR assumes that the bandwidth estimation of an individual node is performed by the MAC layer, the estimated bandwidth information is disseminated to other nodes by adopting *increase waves* and *decrease waves*. These waves are generated when an estimate of the available bandwidth of a core node has changed by a certain amount. Therefore, information about small changes in available bandwidth is kept locally, and only relatively stable bandwidth information is propagated among the core hosts. Increase waves, which provide information about an increase in the available bandwidth of a core node, are propagated periodically; whereas, decrease waves that provide information about a decrease in the available bandwidth of a core node are propagated immediately so that core nodes never overestimate the available bandwidth of another core node.

Route discovery includes the establishment of the core route from the source to the destination via the core nodes. To establish a route, a source node sends a request to its *dominator*, the selected, core host of the node, and the dominator initiates a core broadcast. The core hosts who relay this broadcast attach their IDs in the packet. The dominator of the destination sends a *core_path_ack* message to the dominator of the source. The *core_path_ack* indicates a route from the dominator of the source to the dominator of the destination and thus sets up a valid core route from the source to the destination via the core nodes. Otherwise, if the source dominator has cached a core route to the destination dominator, the source dominator tries to find routes to the furthest core host (e.g., host *T*) in the cached core route that guarantees the required bandwidth, using cached local information. For route selection, the shortest-widest route is chosen among all the admissible routes using a two-phase Dijikstra algorithm. Then, host *T* performs the QoS route computation, just as the source dominator would do if it did not have any cached routes. Finally, the concatenation of the partial routes provides a QoS core route from the source dominator to the destination dominator.

CEDAR assumes that resources are reserved (i.e., bandwidth is reserved) instantaneously by locking the specified resources along the selected route. This guarantees that the resources have been reserved before processing the next route request.

Route maintenance in CEDAR is handled by source-initiated route maintenance and dynamic route maintenance initiated in the intermediate core nodes. The former works effectively when a link failure occurs near the source; whereas, the latter works effectively when a link failure occurs near the destination. Although there are no specific, redundant, reserved routes, the existence of cores provides a proactive approach to offering partially-cached core routes.

Ticket-based QoS Routing

Chen and Nahrstedt propose a distributed, ticket-based QoS routing protocol [1] that uses tickets to find delay-constrained or bandwidth-constrained routes. Tickets are distributed during route discovery to provide a means to find routes with available bandwidth/delay and limit the flooding for route request packets.

Resource estimation is required in ticket-based QoS routing to enable each node to determine the delay, bandwidth, and cost of each of its links. Ticket-based QoS routing assumes that this bandwidth/delay information can be obtained from lower layers, and thus resource estimation focuses on handling the error tolerance instead of addressing

how to measure the available bandwidth or delay of the links. Resource estimation incorporates the imprecision of each node's estimate of its neighbors' available resources for delay-aware and bandwidth-aware routing by using an imprecision model based on the node's local state for each outgoing link. Local state includes the bandwidth state, defined as the residual or unused link bandwidth and delay state, defined as the channel delay of each of the node's links. The imprecision model uses a weight function with the variables of an old bandwidth/delay state and a new bandwidth/delay state to estimate the current bandwidth/delay within some precision tolerance.

Route discovery is accomplished by multiple path searches (limited flooding) to find a qualified route using yellow or green tickets. Overall, yellow tickets are used for finding a feasible route with certain delay/bandwidth constraints, while green tickets are used for determining low-cost routes. The number of tickets indicates the number of probes made to find a feasible route. Therefore, when a source node wants to find a QoS-aware route, it first decides the number of tickets it should issue according to the QoS constraint. More tickets are issued by the source host to increase the chance of finding a feasible route if the constraints are strict.

To find a delay-constrained route, intermediate hosts forward more yellow tickets to their neighbors that have lower delay links and more green tickets to their neighbors that have lower cost links. If the delay in a certain intermediate host exceeds the maximum delay allowed, this intermediate host sets the ticket as invalid. For route selection, the destination chooses the route with the lowest cost among the routes that have valid tickets. A similar procedure is used to find a bandwidth-constrained route.

A simple example is shown in Fig. 1. In this figure, S is the source, and D is the destination. Two probes, P1 with two tickets, and P2, with one ticket, are initiated at S and are forwarded to A and B, respectively. P1 is split into P3 and P4, each with a single ticket, at A. P2 is forwarded to C and then forwarded to D. P3 is forwarded to D directly. P4 is forwarded to D through E.

Resource reservation is achieved by the destination sending a confirmation message back to the source along the reserved route after the primary route is selected. This enables nodes along the selected route to update their estimates for bandwidth/delay with each of their neighbors and lock the corresponding resources for the established route.

Route maintenance is triggered when a route is broken. The host that discovers a route break searches for a local repair by broadcasting its next-hop node's next-hop node (nh_k^2) to its neighbors and finding a neighbor that has sufficient resources to reach nh_k^2 and thus become an alternative path. If the node's neighbors have sufficient resources available, the route is repaired locally using cached information. Otherwise, a route-break message is sent to the source for re-routing.

Ad Hoc QoS On-demand Routing

Ad Hoc QoS On-demand Routing (AQOR) [7] is an on-demand QoS-aware routing protocol with the following features:

- Available bandwidth estimation and end-to-end delay measurement
- Bandwidth reservation
- Adaptive route recovery

Bandwidth estimation is accomplished by disseminating the traffic information of a host to neighbors through periodic announcement packets, called *Hello* packets. AQOR uses the sum of the neighbors' traffic of a node as the estimated total traffic affecting the node. Note that this estimated traffic can be larger than the real overall traffic (detailed in [7]). This overestimation imposes a stringent bandwidth admission control threshold. The available bandwidth is thus a lower bound on the real available bandwidth. End-to-end, one-way downstream delay estimation is approximated by using half the round trip delay.

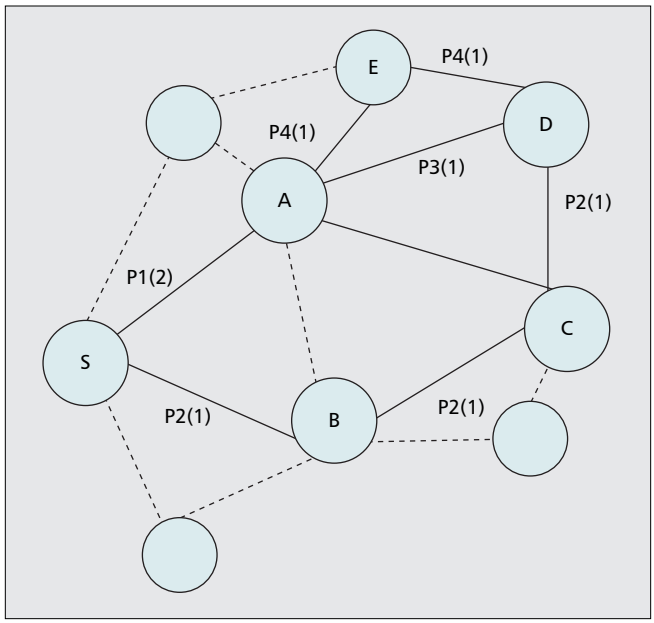

Figure 1. *Illustration of the dissemination of tickets in ticket-based QoS routing.*

Route discovery is triggered when a route is required. The source host initiates a route request, in which the bandwidth and delay requirements are specified. The intermediate hosts check their available bandwidth and perform bandwidth admission hop by hop. If the bandwidth at the intermediate host is sufficient to support the request, an entry will be created in the routing table with an expiration time. If the reply packet does not arrive in the allotted time, the entry will be deleted. Using this approach, a reply packet whose delay exceeds the requirement is deleted immediately to reduce overhead. With the knowledge of available bandwidth and end-to-end delay, the smallest delay route with sufficient bandwidth is chosen during route selection.

Bandwidth reservation is made along the route discovered, but it is activated while the data flow passes the nodes along the reserved route. Temporary reservation is used to free the reserved resources efficiently at each node when the existing routes are broken. If a node does not receive data packets for a certain time interval, the node immediately invalidates the reservation. This avoids using explicit resource release control packets upon route changes.

Route maintenance in AQOR includes network partition detection and destination QoS violation detection. The adaptive route recovery procedure includes detection of broken links and triggered route recovery at the destination, which occurs when the destination node detects a QoS violation or a time-out of the destination resource reservation. The basic neighbor lost detection is used to spot network partitions or route failures. After a broken link is detected, the source re-initiates route discovery. AQOR does not maintain redundant routes designed for fast recovery when QoS violations occur, but detection of a destination QoS violation is incorporated to react quickly to QoS violations.

Trigger-based Distributed-QoS Routing

Trigger-based Distributed-QoS Routing (TDR) is a location-based routing protocol proposed by De et al. [8]. This protocol distinguishes itself from other location-based protocols by using a local neighborhood database, an activity-based database, call admission during route discovery, soft reservations, and route break prediction to support QoS. TDR assumes that bandwidth estimation is performed in lower layers.

Every host maintains two databases: a local neighbor database and an activity-based database. Hosts are required to periodically broadcast beacons that carry their location and mobility information. The neighbors that receive these beacons record the power level of the received beacon and the location and mobility information in their local neighborhood database. In addition to the neighborhood database, every node that participates in a data transmission session keeps an activity-based database. In the activity-based database, routing information is recorded for every session. The activity-based database is refreshed by in-session data packets; this is also called *soft state*.

Route discovery in TDR uses selective forwarding. When a source node wants to initiate a route discovery, it floods route discovery packets to its neighbors; however, to ensure stable routes, only neighbors who receive the packet with power greater than a certain threshold are considered as possible links in the route. When the two-dimensional destination location is available in the source cache, selective forwarding-based route discovery is used. During the process of forwarding the route discovery packets, intermediate hosts check whether their residual bandwidth is sufficient to meet the request. If not, the intermediate hosts do not forward the route discovery packet. Thus, admission control is performed according to the resources available in the network. The destination node sends back a route acknowledgement when it receives the first discovery packet. Upon receiving this acknowledgement packet, the bandwidth reservation in the databases of all intermediate nodes is updated. The destination also sends its location update via the route acknowledgement packet when there has been an appreciable change in its location (based on the GPS information of the destination).

Route maintenance in TDR is similar to route maintenance in an adaptive QoS routing algorithm (ADQR, discussed next): three different receive-power levels are defined, $P_{th1} > P_{th2} > P_{cr}$, to predict route breaks. When the receive-power level at a particular link is lower than P_{cr}, the upstream active node initiates a rerouting process that is called link degradation triggered rerouting. When the power level is between P_{th2} and P_{cr}, the intermediate node sends a rerouting request to the source node. Upon receiving the request, the source initiates a rerouting procedure. When the power level is between P_{th1} and Pth2, the intermediate node initiates the rerouting.

Adaptive QoS Routing Algorithm

Hwang and Varshney proposed an ADQR algorithm to find multiple disjoint routes with long lifetimes [9]. ADQR differs from other QoS routing protocols by using signal strength to predict route breaks and initiate a fast reroute of data. Information on the estimated bandwidth is assumed to be obtained from lower layers.

ADQR categorizes received signal strength into three levels, Th_1, Th_2, and S_r ($Th_1 > Th_2 > S_r$), as shown in Fig. 2. S_r is the minimum signal strength required to receive a data packet. Three different classes are defined for nodes, links, and routes. If the received signal strength from a neighbor node is higher than Th_1, that neighbor node is in the first node class (as shown in Fig. 2). If the received signal strength from the neighbor is between Th_1 and Th_2, that neighbor node is in the second node class. If the signal strength is between Th_2 and S_r, that neighbor node is in the third node class. Links between the first node class nodes are in the first link class; links between the second node class nodes are in the second link class; and links between the third node class nodes are in the third link class. Also, three route classes are defined, where the bottleneck link determines the route class. Each node keeps a neighbor table, which records the node's neighbors and their corresponding exponentially-averaged signal strength, defined as: $SS_{new-cumulative} = \delta \times SS_{old-cumulative} + (1 - \delta) \times SS_{new-measured}$, where δ is adjusted according to network conditions and $SS_{new-measured}$ is the current received signal strength. Additionally, each node keeps a routing table.

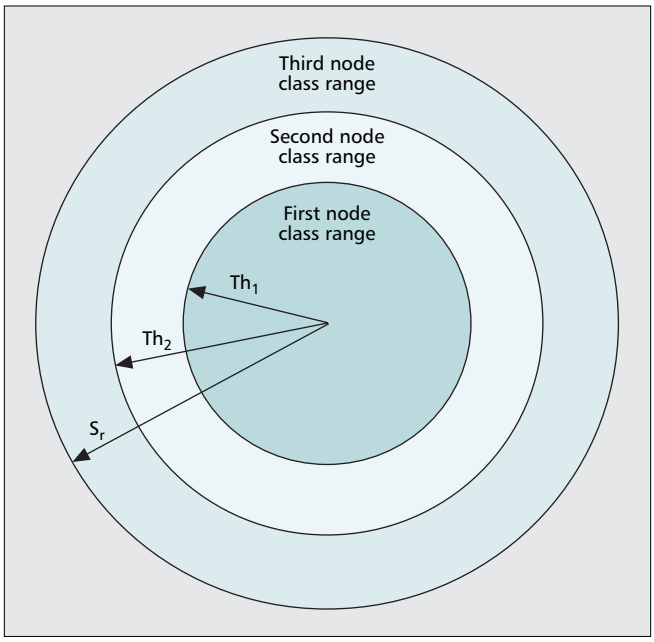

Figure 2. *Received signal strength thresholds (*Th_1, Th_2, *and* S_r*) and node class ranges for ADQR [9].*

Route discovery begins with the source node sending a *Route_Request* packet. Intermediate nodes append their own address in the *Route_Request*, update the parameters — including available bandwidth information — in the *Route_Request*, and forward it to their neighbors. The destination node checks whether this route is disjoint from other routes already found and whether the route contains links with good signal strength. If so, the destination creates a *Route_Reply* packet and inserts the route information into its routing table. When an intermediate node receives a *Route_Reply* packet, the node inserts the route into its local routing table, if there is no corresponding route entry; or the node updates its routing table, if the route already exists. When the source node receives multiple routes, route selection is based on the signal strength of the links in the route, where routes with strong links have the highest selection priority. After selecting the desired route(s), a *QoS_Reserve* packet is sent from the source to the destination along the selected route(s) to perform bandwidth reservation. To guarantee the reservation is made correctly, a *QoS_ACK* packet is sent back to the neighbor from which the *QoS_Reserve* packet is received.

Route maintenance in ADQR is designed to quickly react to network changes by using the signal strength information obtained from lower layers. A fast route maintenance scheme, called two-phase monitored rerouting, composed of *Pre_Rerouting* and *Rerouting* is link class dependent and is used to react with route signal strength changes. The

Routing protocol	QoS metric	Bandwidth/delay estimation	Route discovery	Resource reservation	Route break prediction	Redundant routes
CEDAR [6]	Bandwidth	No	Proactive/Reactive	Yes	No	No
Ticket-based [1]	Bandwidth, delay	No	Reactive	Yes	No	Yes
OLSR-based [10]	Bandwidth	Yes	Proactive	No	No	No
AQOR [7]	Bandwidth, delay	Yes	Reactive	Yes	No	No
ADQR [9]	Bandwidth	No	Reactive	Yes	Yes	Yes
TDR [8]	Bandwidth	No	Reactive	Yes	Yes	No
BEQR [12]	Bandwidth	Yes	Reactive	No	No	No

Table 1. *Comparison of QoS-aware routing protocols.*

Pre_Rerouting phase occurs when the signal strength of a link on the route decreases beyond a threshold, and the *Rerouting* phase is invoked when the signal strength further deteriorates. In *Pre_Rerouting*, the source node finds alternate routes in advance, before the current route becomes unavailable, and in *Rerouting*, the source node switches to one of these alternate routes in advance of the current route becoming unavailable. In addition, in the route table, a source host records all possible routes to the destination. By using this caching scheme, redundant routes are available in case of a weak transmission link.

Optimized Link State Routing-based Routing

Ge et al. [10] integrated QoS features into the Optimized Link State Routing (OLSR) protocol [11] to find a route with larger bandwidth (OLSR-based). This approach does not modify the routing scheme of OLSR, but it chooses different criteria that incorporate bandwidth into consideration to select the multipoint relay (MPR) set so as to find a larger bandwidth route. Bandwidth estimation is performed by taking advantage of the carrier-sense capability in the IEEE 802.11 MAC protocol and measuring the percentage of busy time to get the available bandwidth information. Route maintenance and resource reservation are not considered in this protocol.

QoS-aware Routing Based on Bandwidth Estimation

QoS-aware Routing based on Bandwidth Estimation (BEQR) [12], a reactive routing protocol based on ad-hoc on demand distance vector (AODV), incorporates adaptive feedback and admission control by estimating the available bandwidth at each host during route discovery. BEQR supports both call admission and adaptive feedback to the source node.

Bandwidth estimation is a crucial step in both the admission and adaptive feedback approaches in BEQR. BEQR presents two ways for nodes to estimate their residual bandwidth — *Listen* bandwidth estimation and *Hello* bandwidth estimation. Using the *Listen* bandwidth estimation method, hosts monitor their traffic flows and evaluate the ratio of free time and busy time to determine the available bandwidth. Using the *Hello* bandwidth estimation method, hosts disseminate their available bandwidth information to their neighbors through periodic *Hello* messages. Every host caches its two-hop neighbors' bandwidth information. Therefore, each host determines its available bandwidth locally by comparing the raw channel bandwidth with the cached used bandwidth. The performance of BEQR using the *Listen* bandwidth estimation method is better in terms of overhead; whereas, the performance of BEQR using the *Hello* bandwidth estimation method is better in terms of the ability to quickly release bandwidth when routes fail.

Route discovery in BEQR varies under different requirements. To perform call admission, source nodes put the requested bandwidth information in the route request (RREQ) packet header, and the hosts who receive the RREQ check their available bandwidth. If sufficient bandwidth is available, the intermediate nodes relay the RREQ packet; otherwise, they discard it. When the destination host receives a RREQ packet, it estimates the flow contention (i.e., the inter-flow interference caused by nearby nodes servicing different flows contending for channel access at the same time) [13] among the hops that will participate in the transmission of data for this new flow and then makes a final decision on flow call admission. If the flow is admitted, the destination sends a route reply (RREP) packet to the source using the reverse route taken by the RREQ packet.

To perform adaptive feedback, the RREQ packet again carries the requested bandwidth. Intermediate hosts update the requested bandwidth value if their available bandwidth is less than the value recorded in the RREQ packet. The destination host makes a final adjustment to cover the bandwidth deduction caused by flow self-contention (i.e., the intra-flow interference caused by multiple nodes on the route servicing this flow contending for channel access at the same time), and it sends back a RREP with the admissible sending rate that can be supported by the route using the reverse route taken by the RREQ packet.

Although bandwidth is not explicitly reserved during route discovery, soft bandwidth reservation automatically is made by monitoring network traffic, which aids in performing admission control and adaptive feedback. Route maintenance in BEQR is similar to AODV in that an *error message*, initiated by the host who cannot receive a *Hello* message from a downlink neighbor for a certain period, is forwarded back to the

Routing protocol	Mobility support	Routing overhead	Additional requirements	Network architecture
CEDAR [6]	Medium • Cached routes via cores	• Core setup • Proactive broadcasting of link state information among cores	No	Hierarchical
Ticket-based [1]	High • Secondary path • Local route repair	• Limited flooding of RREQ	No	Flat
OLSR-based [10]	Low	• MPR set setup • Proactive broadcasting of routing packets among MPRs • Limited flooding of RREQ	No	Hierarchical
AQOR [7]	Medium • Packets to detect link breaks	• "Hello" packets • Full flooding of RREQ	No	Flat
ADQR [9]	High • Route break prediction	• Link state inf. In pkt headers • Full flooding of RRQ	Yes • RSSI information	Flat
TDR [8]	High • Route break prediction	• Location updates • Full flooding of RRQ	Yes • Location information • RSSI information	Flat
BEQR [12]	Medium • Packets to detect link breaks	"Hello" packets • Full flooding of RREQ	No	Flat

■ Table 2. *Comparison of QoS-aware routing protocol performance.*

source host. To immediately release the reserved bandwidth, *Immediate Hello* messages also are forwarded back along the route to the source host.

Protocol Comparisons

We have described some representative protocols with various unique features for providing QoS support at the routing level. Each of these protocols addresses the problems of bandwidth/delay estimation, route discovery, resource reservation, and route maintenance in a unique manner, providing various advantages and disadvantages for each protocol. Table 1 gives a summary of each of the protocols, and the next section provides a comparison of the performance of the protocols.

Routing Protocol Performance

While there are many ways to evaluate the performance of QoS-aware routing protocols, the following are some important metrics for classifying and comparing their performance: the protocol support for different levels of node mobility, the amount of overhead imposed by the protocol, additional requirements on the node, and how gracefully the protocol scales as the network increases in size. In this section, we discuss how each of the protocols from the previous section performs according to these metrics. This discussion is summarized in Table 2.

Mobility Support

Node mobility leads to broken links, causing QoS violations while the route break is fixed. There are several well-studied approaches used in the QoS-aware routing protocols discussed in this article to reduce the affects of such link breaks, including caching schemes, route break prediction, and explicit handshaking to quickly detect link failures.

Caching is used to support node mobility in CEDAR and ticket-based QoS routing. Three different caching techniques are used — proactive dissemination of link states, the use of disjoint back-up routes, and caching alternative two-hop nodes for locally repairing a route break. Using a proactive approach, when a route breaks, a new route is chosen from among the cached routes in the routing table. Under high node mobility, the cached information may be out of date, resulting in poor support for mobility. However, under low node mobility, the cached information is more likely to be accurate, and the proactive approach provides a quick repair when a route is broken due to mobility. CEDAR uses a slow-moving increase-wave and a fast-moving decrease-wave to ensure the correctness of the cached information, and it adopts the use of a core set to cache link state information. These two techniques help reduce the overhead in maintaining up-to-date caches.

Another approach to providing support for mobility is to initially find more than one route and use the additional routes as back up when the primary route breaks. This idea is used in ticket-based QoS routing — when the source node receives notification of a route failure, it switches to the *back-up route*. Furthermore, ticket-based QoS routing uses local path repairing to respond to a route break. If there is a neighbor who can repair the route break, a quick local repair is performed instead of resorting to the back-up route. This greatly reduces the time in which the route is invalid. Combined, these two techniques enable ticket-based QoS routing to adapt quickly to node mobility.

Rather than recovering from a broken link after the fact, some protocols try to predict when a link will fail and proactively find a new route before the failure occurs. This technique is used in ADQR and TDR, which use received signal strength to predict a link break. This efficiently improves QoS

performance when node mobility is high and thus, links break frequently.

A technique to quickly detect broken links is to use explicit periodic handshakes. For example, AQOR and BEQR use Hello packets to determine the state of links with a node's neighbors. Although this approach cannot repair a broken path quickly, and thus it cannot support high node mobility, it enables a new route discovery to be started quickly, reducing the route down-time.

Routing Overhead

The second metric for evaluating the protocols is routing overhead, which refers to the extra packets required by the routing protocol to support its operation. Every routing protocol requires regular routing packets, such as RREQ, RREP, and so on. However, hierarchical routing protocols require additional packets to support protocol operations. For example, in CEDAR, overhead packets are required to set up and maintain the core set and in OLSR-based QoS routing, overhead packets are required to set up the MPR set. Although CEDAR and OLSR-based QoS routing use these extra packets to set up a hierarchical network, this hierarchical scheme enables the link state information and routing packets to be exchanged only among the core or MPR sets, which actually helps reduce the overhead compared with pure broadcasting. Therefore, the trade-off between the cost of setting up this hierarchical network and the savings from this scheme depends on the traffic in the network and the network topology. Ticket-based QoS routing and OLSR-based QoS routing adopt limited flooding to reduce the overhead of the route discovery procedure, thus making them low-overhead protocols. All the other protocols use full broadcasting of RREQ packets and thus have high overhead.

Additional Requirements

Routing protocols such as AQOR and TDR employ a route break prediction scheme that requires information about received signal strength. Therefore, a signal strength detector is required for these routing protocols. Similarly, for QoS-aware location-based routing protocols such as TDR, location information is required to determine the positions of the nodes.

Network Scale

In general, it is difficult to directly compare the scalability of the protocols without performing fair comparisons using common simulation parameters. However, we can comment on some general trends. The amount of overhead for a routing protocol depends on two factors: how much overhead is required to set up a route and how often route set up is required, which is a function of node mobility and traffic density. When node mobility is high, the most important factor to determine whether or not the network will scale well is the ability of the protocol to recover from route breaks using minimal overhead. Thus, protocols such as ticket-based QoS-aware routing that incorporate local route repair should scale well in networks with high mobility. Other approaches to maintaining QoS in the face of node mobility, such as cached routes, route break prediction, and back-up paths require large amounts of overhead to maintain up-to-date information when mobility is high and thus cannot help the protocol scale well.

When traffic density is high, the most important factor is the route set-up overhead. Thus, protocols such as CEDAR, ticket-based QoS-aware routing, and OLSR-based QoS-aware routing that incorporate mechanisms such as caching and back-up paths, as well as limited flooding of RREQ messages or proactive maintenance of link state information, are expected to scale well.

Inter-flow and Intra-flow Contention

Of the protocols surveyed here, only AQOR and BEQR explicitly consider inter-flow contention when performing bandwidth estimation. However, only BEQR considers both inter-flow contention and intra-flow contention, considering the transmission range and carrier sensing range among the nodes in the new route. As inter-flow contention could potentially destroy existing established QoS routes, including it in bandwidth estimation is a must for maintaining QoS in real networks.

Open Issues in QoS-aware Routing

There are still many open questions that must be solved to improve the performance of QoS-aware routing protocols. In this section, we discuss these open issues, building from the solutions proposed in the protocols surveyed in this article.

Bandwidth/Delay Estimation

The first open issue in supporting QoS-aware routing is: *what is the best way to estimate available bandwidth and/or delay to maximize accuracy and minimize overhead for resource estimation?* The challenge in wireless ad hoc networks is that neighboring hosts must share the bandwidth, and there is no centralized control for allocating bandwidth among the nodes. Furthermore, intermediate hosts take part in forwarding packets. Therefore, the total effective capacity achievable is not only limited by the raw channel capacity, but it is also limited by the interaction and interference among neighboring hosts. Thus, to offer bandwidth-guaranteed or delay-guaranteed routing, bandwidth/delay estimation is required, yet accurately estimating available bandwidth/maximum delay at each host is a challenging problem.

As shown in Table 1, most existing QoS-aware routing protocols assume that the available bandwidth is known. For those protocols that do include bandwidth estimation, two methods have been proposed.

- Exploit the carrier-sense capability of IEEE 802.11 and measure the idle and busy time ratio (used in OLSR-based QoS routing and BEQR).
- Add bandwidth consumption information to route control packets and exchange this information with neighbor hosts (used in AQOR and BEQR).

The estimated available bandwidth is different than the rates of the supported flows, due to intra-flow contention. To address the intra-flow contention problem, the following have been proposed: assuming the network is well-connected, relating the approximate available bandwidth to the number of hops in the route [13]; and the use of Pre-Reply Probe (PRP), and Route Request Tail (RRT) packets [14]. Further research is required to determine the accuracy and overhead for these bandwidth estimation and intra-flow contention estimation techniques.

Only two of the surveyed routing protocols incorporate delay estimation (Table 1). These protocols do not support a specified delay. They merely determine the shortest delay route during route discovery, and they do not take into account changes in contention levels that will impact the end-to-end delay significantly after the flow is started. Also, the effect of intra-flow contention on delay has not been studied sufficiently.

Unlike in wired networks, hosts in wireless networks have no knowledge about available bandwidth resources or delay at the network layer, due to the shared wireless channel. Thus, a host cannot make an accurate decision on call admission or provide feedback on the network status, based on information obtained from the network layer — a cross-layer design is key to solving this problem. All the bandwidth and delay estima-

tion methods discussed in this article are associated with the capabilities of the underlying MAC protocol.

Route Discovery

Route discovery can be categorized as proactive or reactive and optionally, as location-based. Generally, reactive routing protocols perform better in terms of overhead; whereas, proactive routing protocols require less time for route discovery. To provide QoS, timely information about network status and fast rerouting in the event of route breaks are desired. Proactive routing protocols show advantages in minimizing delay for route set up and maintenance. However, the overhead that proactive routing protocols bring is a problem for bandwidth-constrained MANETs. Therefore, the second open issue is: *which class of routing protocols, reactive or proactive, is better for supporting QoS routing to balance overhead and delay?* The traditional proactive approach may not properly meet the requirements of a QoS-aware routing protocol due to the large amount of overhead to proactively maintain routes, but protocols, such as CEDAR that provides a core to minimize overhead might be a good solution. Alternatively, a hybrid protocol may provide the optimal solution. In addition, route reliability, which is a very important metric to ensure route quality, has not been addressed by any of the QoS-aware routing protocols we have surveyed. Future work should explore incorporating route reliability into the route selection and route maintenance procedures.

Resource Reservation

One difference between regular routing protocols and QoS-aware routing protocols is that QoS-aware routing requires some form of resource reservation. TDR uses temporary reservation of bandwidth during route discovery and updates the reservation upon receiving a route deactivation packet. Also, the reserved bandwidth is updated in a fixed *soft-state* interval. Temporary reservation is a one time action, but the soft state is updated periodically by the in-session data. AQOR also uses a temporary reservation mechanism to eliminate the connection tear-down process along the old route when the route is adjusted. One unique feature of AQOR is that QoS violations are detected at the destination, prompting destination-initiated route recovery. ADQR uses a *QoS_Reserve* packet to reserve bandwidth from the source to the destination. CEDAR does not explicitly describe the signaling approach used, but it assumes the existence of some instantaneous reservation mechanism. Ticket-based QoS routing uses a confirmation message to lock the established route after the primary route is selected. OLSR-based QoS routing and BEQR do not incorporate any explicit reservation schemes.

Resource Reservation Protocol (RSVP)-type signaling [15], used extensively in wired networks, requires a large amount of overhead and thus is not directly suitable for MANETs. An in-band signaling technique for MANETs has been proposed in [16] and shown to work well in MANETs. Thus, the third open issue in QoS-aware routing is: *how should in-band signaling be coupled with the routing protocol for resource reservation?* The following ideas were proposed to minimize overhead exchange for resource reservation.

- *Soft-state reservations*: nodes use the active data transmission to reserve the corresponding bandwidth.
- *Temporary reservations*: the bandwidth is reserved only for a certain interval and if no data packets are received for a certain time, the reservation is automatically released.
- *Destination-initiated recovery*: the destination initiates a routing request procedure when a QoS violation is found.

Route Maintenance

In MANETs, routes change frequently when topology and traffic patterns change, and this adversely affects QoS at the routing level. Therefore, the fourth open issue in QoS-aware routing is: *how should the prediction of route breaks, route redundancy, and rerouting optimization be incorporated into a rerouting scheme to balance overhead with QoS performance?* The surveyed protocols use the following techniques to address this issue:

- *Signal strength triggered reroute*: Using received signal strength to predict link breaks (and hence route breaks) [17], a host prepares to reroute data when the received signal strength falls below a threshold, and it reroutes the data as the signal strength further deteriorates. Therefore, the data is rerouted through a new route that can support the QoS requirements before the route breaks, reducing the transmission break time and avoiding sending packets along a route that soon will be broken.
- *Route redundancy*: In this approach, hosts maintain secondary routes to use when the primary route fails. However, there is a trade-off between route redundancy and overhead.
- *Other proposed schemes for rerouting data*: Some protocols use route re-computation at the failure point when a link failure occurs near the destination and route re-computation at the source when a link failure occurs near the source. Destination triggered rerouting and neighbor loss detection triggered rerouting also are used in some protocols.

Each of these techniques requires extra overhead, and thus it is not obvious which one(s) will provide benefit in QoS performance for different network scenarios.

Cross-Layer Design

Cross-layer design is not a new concept in wireless networking. However, it is extremely important in supporting QoS in MANETs due to the shared media and distributed organization of the network. For example, the fact that the wireless channel is shared among neighbors makes the estimation of available resources extremely difficult. Collaboration among the layers can help with the processes of resource estimation. Similarly, feedback from the network layer to the application on available resources provides applications the opportunity to adjust their transmission appropriately. Thus, the final open issue in QoS-aware routing is: *what types of information should be shared among the layers to best support QoS-aware routing in MANETs?* Many challenges are ahead for cross-layer design, such as the optimal architecture and the trade-off of gain and increased complexity. Furthermore, one can debate how much adaptation actually should be performed at the network layer, and how much is better left to the application; using an end-to-end approach to enable the application to adapt to dynamic network conditions via approaches such as adaptive coding. While this article surveys only the network layer techniques for QoS adaptation, we must not lose sight of the importance of application-layer adaptation in an overall system solution to provide quality of service in mobile ad hoc networks.

Summary

In this article, we presented a survey of several QoS-aware unicast routing protocols for MANETs. We compared these routing protocols in terms of their support for node mobility, their routing overhead, their requirements for extra node hardware, and their support for scaling of the network. We

also pointed out the open issues that must be addressed in QoS-aware routing in terms of bandwidth/delay estimation, route discovery, resource reservation, and route maintenance.

This article presented a survey of QoS-aware unicast routing protocols for MANETs. Another important aspect in providing QoS at the routing layer is multipath routing, which provides spatial redundancy in data transmissions. Furthermore, as end-to-end communication is the result of the cooperation of all the network layers [3, 4, 18, 19], a cross-layer design is the key to providing QoS to applications in MANETs.

Acknowledgements

The authors would like to thank the supervising Associate Editor-in-Chief and the anonymous reviewers for their feedback, which has greatly improved this article.

References

[1] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service in Ad Hoc Networks," *IEEE JSAC*, vol. 17, no. 8, 1999.
[2] S. Chakrabarti, "QoS Issues in Ad Hoc Wireless Networks," *IEEE Commun. Mag.*, vol. 39, no. 2, Feb. 2001, pp. 142–48.
[3] P. Mohapatra, J. Li, and C. Gui, "QoS in Mobile Ad Hoc Networks," *IEEE Wireless Commun.*, Special Issue on QoS in Next-Generation Wireless Multimedia Communications Systems, vol. 10, no. 3, June 2003, pp. 44–52.
[4] H. Zhu *et al.*, "A Survey of Quality of Service in IEEE 802.11 Networks," *IEEE Wireless Commun.*, vol. 11, no. 4, Aug. 2004, pp. 6–14.
[5] T. Bheemarjuna Reddy *et al.*, "Quality of Service Provisioning in Ad Hoc Wireless Networks: A Survey of Issues and Solutions," *Ad Hoc Networks*, vol. 4, no. 1, Jan. 2006, pp. 83–124.
[6] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: A Core-Extraction Distributed Ad hoc Routing Algorithm," *IEEE INFOCOM '99*, New York, NY, Mar. 1999.
[7] Q. Xue and A. Ganz, "Ad Hoc QoS On-demand Routing (AQOR) in Mobile Ad hoc Networks," *J. Parallel and Distrib. Comp.*, vol. 62, no. 2, Feb. 2003, pp. 154–65.
[8] S. De et al., "Trigger-Based Distributed QoS Routing in Mobile Ad Hoc Networks," *ACM SIGMOBILE Mobile Comp. and Commun. Rev.*, vol. 6, no. 3, July 2002, pp. 22–35.
[9] Y. Hwang and P. Varshney, "An Adaptive QoS Routing Protocol with Dispersity for Ad-hoc Networks," *Proc. 36th Hawaii Int'l. Conf. Sys. Sci.*, Jan. 2003.
[10] Y. Ge, T. Kunz, and L. Lamont, "Quality of Service Routing in Ad-Hoc Networks Using OLSR," *Proc. 36th Hawaii Int'l. Conf. Sys. Sci.*, Jan. 2003.
[11] P. Jacquet *et al.*, "Optimized Link State Routing Protocol," draft-ietf-manet-olsr-05.txt, Internet draft, IETF MANET Working Group.
[12] L. Chen and W. Heinzelman, "QoS-aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks," *IEEE JSAC*, Special Issue on Wireless Ad Hoc Networks, vol. 23, no. 3, Mar. 2005, pp. 561–72.
[13] J. Li *et al.*, "Capacity of Ad Hoc Wireless Networks," *Proc. ACM MobiCom '01*, 2001, pp. 61–69.
[14] K. Sanzgiri, I. Chakeres, and E. Belding-Royer, "Determining Intra-Flow Contention along Multihop Paths in Wireless Networks," *Proc. Broadnets 2004 Wireless Networking Symp.*, San Jose, CA, Oct. 2004.
[15] L. Zhang *et al.*, "RSVP: A New Resource Reservation Protocol," *IEEE Network*, vol. 7, Sept. 1993, pp. 8–18.
[16] S. B. Lee *et al.*, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," *J. Parallel and Distrib. Comp.*, Special Issue on Wireless and Mobile Computing and Commun., vol. 60, no. 4, Apr. 2000, pp. 374–406.
[17] T. Goff *et al.*, "Preemptive Maintenance Routing in Ad Hoc Networks," *Proc. ACM MobiCom '01*, 2001, pp. 43–52.
[18] K. Wu and J. Harms, "QoS Support in Mobile Ad Hoc Networks," *Crossing Boundaries — The GSA J.*, Univ. of Alberta, vol. 1, no. 1, Nov. 2001, pp. 92–106.
[19] L. Chen and W. Heinzelman, "Network Architecture to Support QoS in Mobile Ad Hoc Networks," *Proc. Int'l. Conf. Multimedia and Expo*, vol. 3, June 2004, pp. 1715–18.

Biographies

LEI CHEN (chenlei@ece.rochester.edu) received a B.S. degree in information science and electronics engineering from Zhejiang University, China, in 1999, and M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, New York, in 2002 and 2006, respectively. Her research interests are in the areas of QoS-aware architectures and congestion control in ad hoc networks.

WENDI B. HEINZELMAN [SM] (wheinzel@ece.rochester.edu) received a B.S. degree in electrical engineering from Cornell University in 1995, and M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT) in 1997 and 2000, respectively. She is an associate professor in the Department of Electrical and Computer Engineering at the University of Rochester. Her current research interests are in the area of wireless communications and networking, mobile computing, and multimedia communication. She received the NSF CAREER award in 2005 for her work on cross-layer architectures for wireless sensor networks, and the ONR Young Investigator Award in 2005 for her work on balancing resource utilization in wireless sensor networks. She is General Vice Chair for SECON '07 and a member of the ACM.