

QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks

Lei Chen, *Student Member, IEEE*, and Wendi B. Heinzelman, *Member, IEEE*

Abstract—Routing protocols for mobile ad hoc networks (MANETs) have been explored extensively in recent years. Much of this work is targeted at finding a feasible route from a source to a destination without considering current network traffic or application requirements. Therefore, the network may easily become overloaded with too much traffic and the application has no way to improve its performance under a given network traffic condition. While this may be acceptable for data transfer, many real-time applications require quality-of-service (QoS) support from the network. We believe that such QoS support can be achieved by either finding a route to satisfy the application requirements or offering network feedback to the application when the requirements cannot be met. We propose a QoS-aware routing protocol that incorporates an admission control scheme and a feedback scheme to meet the QoS requirements of real-time applications. The novel part of this QoS-aware routing protocol is the use of the approximate bandwidth estimation to react to network traffic. Our approach implements these schemes by using two bandwidth estimation methods to find the residual bandwidth available at each node to support new streams. We simulate our QoS-aware routing protocol for nodes running the IEEE 802.11 medium access control. Results of our experiments show that the packet delivery ratio increases greatly, and packet delay and energy dissipation decrease significantly, while the overall end-to-end throughput is not impacted, compared with routing protocols that do not provide QoS support.

Index Terms—Bandwidth estimation, mobile ad hoc networks (MANETs), quality-of-service (QoS)-aware routing.

I. INTRODUCTION

THE ATTRACTIVE infrastructure-less nature of mobile ad hoc networks (MANETs) has gained a lot of attention in the research community. With the success of solving the most basic but important problems in all network layers, people realize there is commercial value in MANETs. Most applications that attract interest for use in current wired networks (e.g., video conferencing, on-line live movies, and instant messenger with camera enabled) would attract interest for MANETs as well. However, ad hoc networks present unique advanced challenges, including the design of protocols for mobility management, effective routing, data transport, security, power management, and quality-of-service (QoS) provisioning. Once these problems are solved, the practical use of MANETs will be realizable. The overall design of a solution for all of these problems is currently too complex. In this paper, we investigate one of these

issues, providing QoS in MANETs, and we focus specifically on routing protocols to support QoS.

In order to design good protocols for MANETs, it is important to understand the fundamental properties of these networks.

- **Dynamicity:** Every host can randomly change position. The topology is generally unpredictable, and the network status is imprecise.
- **Noncentralization:** There is no centralized control in the network and, thus, network resources cannot be assigned in a predetermined manner.
- **Radio properties:** The channel is wireless, so it will suffer fading, multipath effects, time variation, etc.

With these constraints, Hard QoS (e.g., guaranteed constant bit rate and delay) is difficult to achieve. The reasons are as follows.

- To support QoS, in principle, the end host should have precise knowledge of the global status of the network. The dynamic nature of MANETs makes it difficult for hosts to determine information about their local neighborhood, much less the global status of the network.
- It is hard to establish cooperation between neighboring hosts to determine a transmit schedule for guaranteed packet delivery without centralized control. In MANETs, all hosts share the same physical channel, and each host's transmissions will interfere with neighboring hosts' transmissions. This unpredictability makes it hard to guarantee successful transmissions.
- The wireless channel's main deficiency is its unreliability caused by various reasons such as fading and interference.

Thus, our aim is to develop a routing protocol that provides Soft QoS [1] or better than best-effort service, rather than guaranteed hard QoS. However, if the topology changes too frequently, the source host cannot detect the network status changes and cannot make the corresponding adjustment to meet the specific QoS requirements, rendering the QoS meaningless. Therefore, combinatorial stability¹ must first be met before we can consider providing QoS to real-time applications. There are many networks that satisfy this requirement. For example, consider a network made up of students in a class; students may join the lecture late, some may leave the classroom, but most stay in the stationary position.

Providing QoS is desirable for many applications, as this allows them to alter what data they transmit. For example, several image compression techniques, such as MPEG-4 [3], H. 263 [4], and multiple description coding [5], are designed to meet various channel conditions. Our QoS-aware routing

¹Combinatorial stability means that, given a specific time window, the topology changes occur sufficiently slowly to allow successful propagation of all topology updates as necessary [2].

Manuscript received October 20, 2003; revised September 20, 2004

The authors are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: chenlei@ece.rochester.edu; wheinzel@ece.rochester.edu).

Digital Object Identifier 10.1109/JSAC.2004.842560

protocol can provide feedback to the application about the current network state to allow the application to appropriately adjust the amount of compression applied to the video. Without this information, the video may not be compressed enough, causing congestion in the network and a large number of dropped packets, which is much worse than transmitting video using low data rate coding. Some applications require minimum bandwidth support. If the minimum bandwidth cannot be met, all data will be useless. Thus, it is better not to transmit data in this case, because it will just waste network bandwidth and energy. Therefore, an admission control scheme is also embedded into our QoS-aware routing protocol to address this issue.

Another challenge of QoS is medium access control (MAC)-layer design. We argue that the IEEE 802.11 MAC is not the best MAC for supporting QoS. However, it is widely adopted in the wireless local area network (WLAN) community, and many devices have been commercialized with IEEE 802.11. Therefore, in our design, we choose the IEEE 802.11 standard as the underlying MAC layer. IEEE 802.11 has no support for constant bit rate streams, guaranteed delay, etc. Thus, our intention here is to develop a QoS-aware routing protocol using IEEE 802.11 that provides better than best-effort service for real-time video and audio applications.

This paper is organized as follows. Section II gives motivation for our QoS-aware routing protocol. Section III details our proposed method of incorporating QoS into ad hoc on-demand distance vector (AODV). Section IV presents the simulations comparing our QoS-aware routing protocol with AODV. Section V describes related work, and Section VI draws conclusions.

II. MOTIVATION

Routing protocols have attracted a great deal of attention from the beginning of MANET research until the present time. Early work focused on finding feasible routes without considering energy costs or QoS.

AODV is one of the most widely used table-based and reactive routing protocols [6], [7]. In AODV, a source host broadcasts a route request (RREQ) packet when it needs a route to a specific host. Each host that receives the RREQ packet checks whether it is the destination; if it is, it sends a route reply (RREP) packet, otherwise it rebroadcasts the RREQ packet. Intermediate hosts between the source and the destination create an entry in their routing tables and record the neighbor ID of the host from which the RREQ packet was received. The destination host responds to the first RREQ packet it receives by unicasting a RREP to the neighbor from which it received the RREQ packet. The intermediate hosts forward the RREP packet to the source according to their own routing tables. One unique feature in AODV is that hosts use “Hello” messages to probe their neighbors in order to validate routes. Hosts broadcast “Hello” messages in a reasonable interval. If a host does not receive a “Hello” message from a particular neighbor for a certain period, it will delete this neighbor from its neighbor cache and mark the corresponding routes as invalid.

From the description of AODV, we can see that AODV is designed to find a feasible route only. Therefore, the established route has no knowledge about the network status. Other

standardized routing protocols, such as dynamic source routing (DSR) [8], dynamic destination sequenced distance vector routing (DSDV) [9], and TORA [10], also do not incorporate schemes to detect the network status. Therefore, the established routes using these routing protocols cannot inform the application about the network condition, so the application must send its data using a default feeding rate and cannot take advantage of the adaptation feature in various coding technologies. In addition, without knowing the bottleneck throughput, the source may send much more data than the bottleneck host on the route can accommodate. The overwhelmed host must drop data, which wastes a considerable amount of energy and needlessly consumes bandwidth. Also, much time is used in transmitting these data that will eventually be dropped. Therefore, the data that finally reach the destination have to wait in packet queues for a considerably long time, which results in a significantly increased delay.

Therefore, we propose a QoS-aware routing protocol, which is based on residual bandwidth estimation during route set up. Our QoS-aware routing protocol is built off AODV, in which the routing table is used to forward packets, “Hello” messages are used to detect broken routes and “Error” messages are used to inform upstream hosts about a broken route. We explore two ways to perform bandwidth estimation, and we incorporate both an adaptive feedback-based scheme and an admission control scheme.

III. QoS-AWARE ROUTING

QoS is an agreement to provide guaranteed services, such as bandwidth, delay, delay jitter, and packet delivery rate to users. Supporting more than one QoS constraint makes the QoS routing problem NP-complete [11]. Therefore, we only consider the bandwidth constraint when studying QoS-aware routing for supporting real-time video or audio transmission. We propose a QoS-aware routing protocol that either provides feedback about the available bandwidth to the application (feedback scheme), or admits a flow with the requested bandwidth (admission scheme). Both the feedback scheme and the admission scheme require knowledge of the end-to-end bandwidth available along the route from the source to the destination. Thus, bandwidth estimation is the key to supporting QoS.

Our work focuses on exploring different ways to estimate the available bandwidth, incorporating a QoS-aware scheme into the route discovery procedure and providing feedback to the application through a cross-layer design.

A. Bandwidth Estimation

To offer bandwidth-guaranteed QoS, the available end-to-end bandwidth along a route from the source to the destination must be known. The end-to-end throughput is a concave parameter [12], which is determined by the bottleneck bandwidth of the intermediate hosts in the route. Therefore, estimating the end-to-end throughput can be simplified into finding the minimal residual bandwidth available among the hosts in that route. However, how to calculate the residual bandwidth using the IEEE 802.11 MAC is still a challenging problem, because

the bandwidth is shared among neighboring hosts, and an individual host has no knowledge about other neighboring hosts' traffic status. We use two methods for estimating bandwidth in this paper. One is for hosts to listen to the channel and estimate the available bandwidth based on the ratio of free and busy times ("Listen" bandwidth estimation). The other is for every host to disseminate information about the bandwidth it is currently using in the "Hello" messages, and for a host to estimate its available bandwidth based on the bandwidth consumption indicated in the "Hello" messages from its two-hop neighbors ("Hello" bandwidth estimation).

1) *"Listen" Bandwidth Estimation:* To estimate the available bandwidth, intuitively, each host can listen to the channel to track the traffic state and determine how much free bandwidth it has available every second. The IEEE 802.11 MAC utilizes both a physical carrier sense and a virtual carrier sense [via the network allocation vector (NAV)], which can be used to determine the free and busy times. The MAC detects that the channel is free when the following three requirements are met:

- NAV's value is less than the current time;
- receive state is idle;
- send state is idle.

The MAC claims that the channel is busy when one of following occurs:

- NAV sets a new value;
- receive state changes from idle to any other state;
- send state changes from idle to any other state.

A host estimates its available bandwidth for new data transmissions as the channel bandwidth times the ratio of free time to overall time, divided by a weight factor. The weight factor is introduced due to the nature of IEEE 802.11. The DIFS, SIFS, and backoff scheme represent overhead, which must be accounted for in each data transmission. This overhead makes it impossible in a distributed MAC competition scheme to fully use the available bandwidth for data transmission.

Using the "Listen" method to estimate residual bandwidth is straightforward. However, using this approach, the host cannot release the bandwidth immediately when a route breaks, because it does not know how much bandwidth each node in the broken route consumes. "Listen" only counts the used bandwidth, but does not distinguish the corresponding bandwidth cost for each flow. This greatly affects the accuracy of bandwidth estimation when a route is broken. Therefore, we introduce another approach—"Hello" bandwidth estimation—that is better able to reallocate available bandwidth when routes break.

2) *"Hello" Bandwidth Estimation:* In the "Hello" bandwidth estimation method, the sender's current bandwidth usage as well as the sender's one-hop neighbors' current bandwidth usage is piggybacked onto the standard "Hello" message. Each host estimates its available bandwidth based on the information provided in the "Hello" messages and knowledge of the frequency reuse pattern. This approach avoids creating extra control messages by using the "Hello" messages to disseminate the bandwidth information.

To know the frequency reuse pattern, we first study the underlying IEEE 802.11 MAC. As defined in the IEEE 802.11 MAC, hosts are allowed to access the wireless channel when

| ID | Consumed Bandwidth | Timestamp |
|---------------|---------------------------|------------------|
| Neighbor ID 1 | Consumed Bandwidth | Timestamp |
| . | . | . |
| . | . | . |
| . | . | . |
| Neighbor ID n | Consumed Bandwidth | Timestamp |

Fig. 1. Hello structure. The bold item in the first row is the host's own information. The following rows are the host's neighbors' information.

the media is free. The media can be free if no hosts are transmitting packets within the interference range. Normally, the interference range is twice the transmission range, based on the settings of the 914 MHz Lucent WaveLAN card. Therefore, the frequency can be reused outside of the second neighboring hosts' range. The actual upper bound of bandwidth in the two-hop circle varies with the topology and the traffic status, but the raw channel bandwidth is the soft upper bound of total bandwidth. We use this soft upper bound bandwidth in the estimation to approximate the bandwidth usage. With the above frequency reuse pattern, we can simplify the bandwidth calculation problem to determining the residual bandwidth within the two-hop neighborhood range. Therefore, each host can approximate its residual bandwidth information based on information from hosts within two-hops (the interference range).

The first neighboring hosts' information can be obtained directly, but there is no way to get the second neighboring hosts'² bandwidth information directly. There are several ways to get the second neighboring hosts' information, such as disseminating the host bandwidth information using higher transmission power to reach the two-hop neighborhood, and setting up a separate signaling channel to broadcast the bandwidth information. However, using higher power to disseminate information not only consumes much more power, it also destroys the frequency reuse pattern and causes much more interference. Using a separate channel to disseminate the bandwidth information requires additional control that is a heavy burden for the ad hoc network in terms of bandwidth consumption and hardware support. Therefore, we propose using hop relay to disseminate the second neighboring hosts' information. AODV uses the "Hello" messages to update the neighbor caches. The "Hello" message used in AODV only keeps the address of the host who initiates this message. We modify the "Hello" message to include two fields. The first field includes ⟨host address, consumed bandwidth, timestamp⟩, and the second field includes ⟨neighbors' addresses, consumed bandwidth, timestamp⟩, as shown in Fig. 1. Each host determines its consumed bandwidth by monitoring the packets it feeds into the network. This value is recorded in a bandwidth-consumption register at the host and is updated periodically.

²If the hosts are located beyond the transmission range but within the interference range, we call them second neighboring hosts or second neighbors.

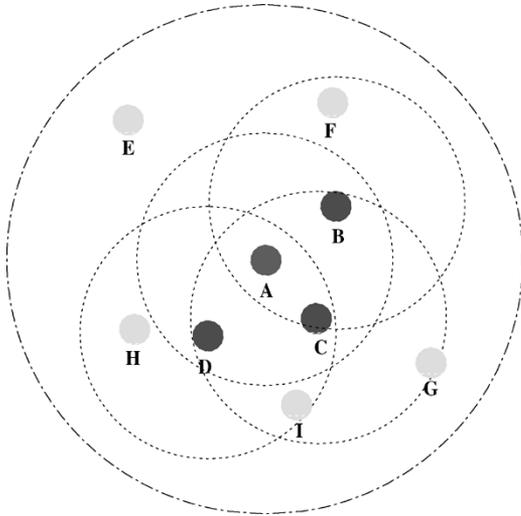


Fig. 2. Hidden node scenario. The big circle indicates host A's interference range. The small circles indicate host A and its first neighboring hosts' transmission range. Hosts B, C, and D are A's first neighbors, and hosts F, G, H, and I are host A's second neighbors. Host E is in host A's interference range, but it is hidden to A.

Using this approach to gather the first and second neighboring hosts' information is imprecise. Fig. 2 shows an example topology that will result in imprecise information. The outside big circle indicates host A's interference range, and the other small-size dotted circles indicate host A and its neighbors' transmission ranges. Host E is not in A's transmission range, but it is in A's interference range. In addition, E does not fall into any of A's neighbors' transmission range. In this situation, A will never know E's status. If E transmits data, A's knowledge of available bandwidth is imprecise. However, this "hidden node" problem does not happen frequently since it has to meet strict requirements to "hide" the host. We argue that this kind of inaccuracy is tolerable because we use a wireless channel, our ultimate aim is better than best effort, and the possibility of "hidden nodes" is low in a well connected network. Even if this situation occurs, it can be overcome by using a conservative bandwidth estimate that leaves some extra bandwidth to conceal this "hidden node" effect.

Once a host receives a "Hello" message from its neighbors, it determines whether this "Hello" is an updated one by examining the message's timestamp. We use the cache structure shown in Fig. 3, which includes a first neighbor table and a second neighbor table. The second neighbors are linked with their corresponding first neighbors in the cache.

Once a host knows the bandwidth consumption of its first neighbors and its second neighbors, the available bandwidth estimation becomes simple. The residual bandwidth is simply the raw channel bandwidth minus the overall consumed bandwidth, divided by a weight factor. We need to divide the residual bandwidth by a weight factor due to the IEEE 802.11 MAC's nature and some overhead required by the routing protocol. In the MAC layer, ready-to-send (RTS), clear-to-send (CTS), and acknowledgment (ACK) packets consume bandwidth, the backoff scheme cannot fully use the entire bandwidth, and packets can collide, resulting in packet retransmissions.

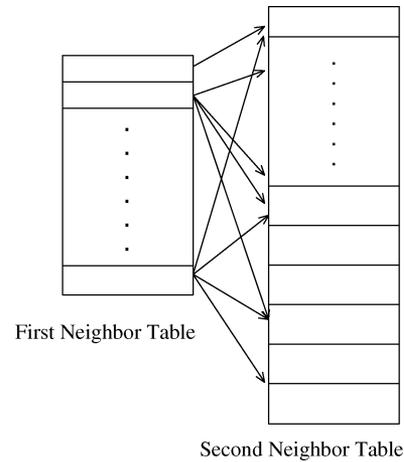


Fig. 3. Neighbor cache structure.

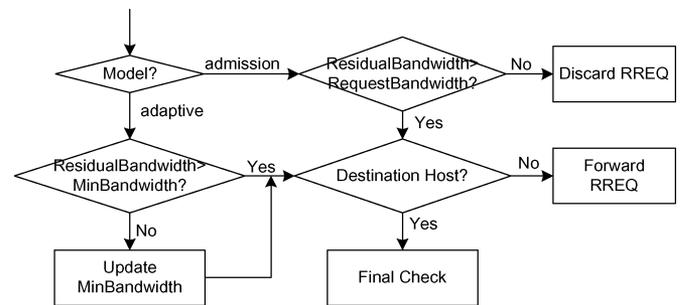


Fig. 4. Hosts' working procedure after receiving a RREQ.

Furthermore, the routing protocol needs some overhead to maintain or discover the routes.

B. Incorporating QoS in Route Discovery

As we stated previously, our QoS-aware routing protocol utilizes a cross-layer design. Therefore, the routing features depend on the application requirements. Our design supports two kinds of applications. One is where the application indicates in the request message the minimal bandwidth that must be guaranteed. The other is where the application can adjust its coding rate according to feedback received from the network.

To initiate QoS-aware routing discovery, the source host sends a RREQ packet whose header is changed to $\langle \text{model-flag, bandwidth request, min-bandwidth, AODV RREQ header} \rangle$. The model-flag indicates whether the source is using the admission scheme or the adaptive feedback scheme. When an intermediate host receives the RREQ packet, it first calculates its residual bandwidth. If the model-flag is the admission scheme, the host compares its residual bandwidth with the requested bandwidth. If its residual bandwidth is greater than the requested bandwidth, it forwards this RREQ. Otherwise, it discards this RREQ. If the model-flag is adaptive, the host compares its residual bandwidth with the min-bandwidth field in the RREQ. If its residual bandwidth is greater than the min-bandwidth, it forwards the RREQ. Otherwise, it updates the min-bandwidth value using its residual bandwidth. The whole procedure is shown in Fig. 4.

When the destination host receives the RREQ packet, it also needs to do the checking procedure as described above. However, after completing this checking procedure, it is not sufficient to say that the current network can offer the min-bandwidth indicated in the RREQ packet. The reason is that if the route is chosen, the chosen hosts will bring mutual interference into the network during transmission. We cannot put this kind of potential interference into consideration while estimating the residual bandwidth during the route discovery procedure. Therefore, one final check procedure is required before sending the RREP packet back to the source host. We directly use the relation of the end-to-end throughput with the number of hops (*HopNumber*) and the bottleneck bandwidth (*MinBandwidth*) in the route as follows (the details can be found in [13]):

$$\begin{aligned} &\text{If } (HopNumber = 1) \\ &\quad MinBandwidth = MinBandwidth \\ &\text{Else if } (HopNumber = 2) \\ &\quad MinBandwidth = \frac{MinBandwidth}{2} \\ &\text{Else if } (HopNumber = 3) \\ &\quad MinBandwidth = \frac{MinBandwidth}{3} \\ &\text{Else} \\ &\quad MinBandwidth = \frac{MinBandwidth}{4} \end{aligned}$$

This equation offers the upper bound of the available bandwidth. A more accurate estimation is studied in [14] and [15], where the interflow contention is accounted for by using the contention counter.

Finally, the destination host sends the RREP with a modified header (min-bandwidth, AODV RREP header) to the source host. Once intermediate hosts receive the RREP, they enable the route and also record the min-bandwidth in their routing table, which is useful for route maintenance of QoS-aware routing with “Hello” bandwidth estimation.

C. Route Maintenance

AODV detects a broken route by monitoring the “Hello” messages. If a host does not receive a “Hello” message from a specific neighbor within a predefined interval, it marks the routes using that neighbor host as invalid and sends a corresponding “Error” message to the upstream hosts. Only the source host reinitiates a routing discovery procedure, once receiving the “Error” message. Thus, using caches to respond to a route break in the intermediate host is not utilized.

When using QoS-aware routing with “Listen” bandwidth estimation, AODV’s route maintenance scheme is used, because releasing bandwidth from the bandwidth consumption registers is impossible without knowing how much bandwidth is consumed by each host in the route. Therefore, no change in AODV’s route maintenance scheme is needed to address the bandwidth releasing issue.

However, we cannot directly use AODV’s route maintenance scheme in the QoS-aware routing protocol with “Hello” bandwidth estimation. We use the simple topology shown in Fig. 5

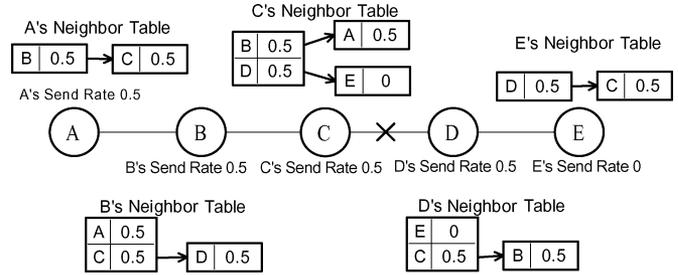


Fig. 5. Route maintenance failure example.

to illustrate what will happen if we adopt AODV’s route maintenance scheme without any modification. The topology is a single chain and is composed of five hosts. Every host is in its neighbor’s transmission range and its second neighbor’s interference range. The source host sends packets with a 0.5 Mb/s feeding rate.³ The first table shows the host’s first neighbors and the linked tables show the host’s second neighbors. If the link between C and D is broken, an “Error” message is initiated in C and A receives it through B’s propagation. Once A gets the error message, A sends a new RREQ. The time interval between claiming a broken route and initiating a route discovery is only several milliseconds. Therefore, the host neighbors’ caches have not yet updated their bandwidth consumption when the new RREQ arrives. If we do not consider the weight factor, when the new RREQ passes by, host C reports that it has no available bandwidth, since it has not released the bandwidth used by the broken route. In fact, all bandwidth is offered to this single chain transmission and the available end-to-end bandwidth is actually 0.5 Mb/s. This problem is caused by the fact that the neighbor cache was not updated in a timely fashion. Therefore, we should incorporate a forced cache update in the route maintenance scheme.

The QoS-aware routing with “Hello” bandwidth estimation uses the first neighbors’ relay to get the second neighbors’ information. Therefore, once the neighbors get the forced updates, they should disseminate the update information immediately to their neighbors. We use an “Immediate Hello” message to address this concern. This special message’s content is exactly the same as the “Hello” message, except the packet type is marked as “Immediate Hello” in order to differentiate with the regular “Hello” message. When a host receives an “Immediate Hello” message, it sends its regular “Hello” message immediately.

The “Error” message is also adopted to trigger an update of bandwidth consumption registers and the dissemination of “Immediate Hello” messages. Once a host receives an “Error” message, it will deduct the amount of bandwidth that the broken route consumes from its bandwidth consumption register to reflect the bandwidth allocation changes. We decide to use two separate packets (“Immediate Hello” and “Error”), because the bandwidth should be released among all the neighboring hosts, which a broadcast packet can do, but the “Error” message is a unicast packet. The procedure by which hosts update their neighbor cache is shown step-by-step in Figs. 5–13, in which host A sends data with 0.4 Mb/s to host E.

³Note that for a 2 Mb/s channel, 0.5 Mb/s is the maximum data rate that can be supported, see [13].

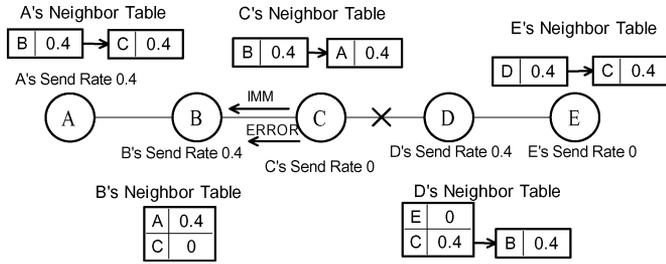


Fig. 6. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 1.

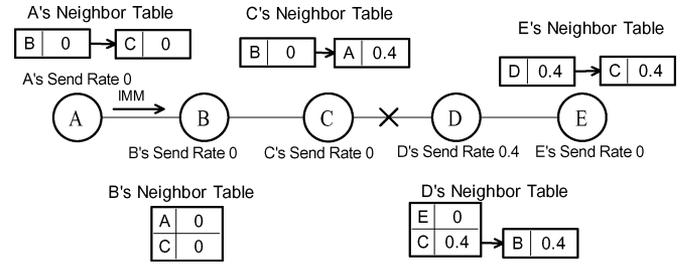


Fig. 10. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 5.

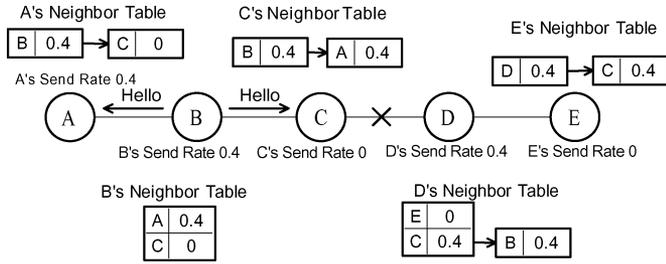


Fig. 7. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 2.

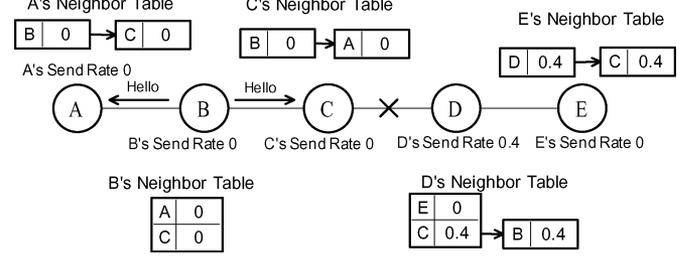


Fig. 11. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 6.

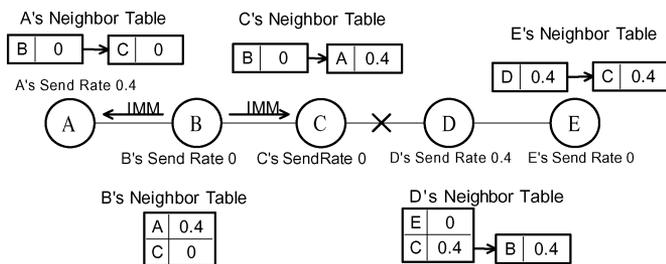


Fig. 8. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 3.

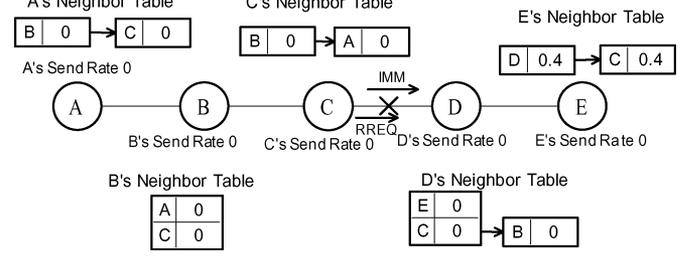


Fig. 12. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 7.

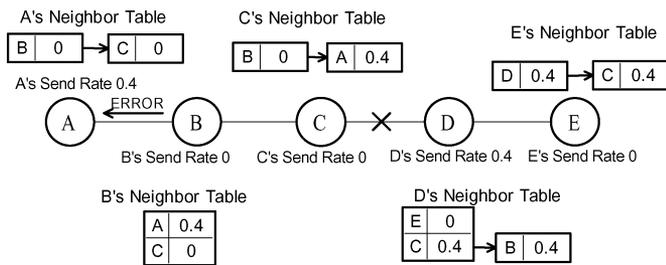


Fig. 9. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 4.

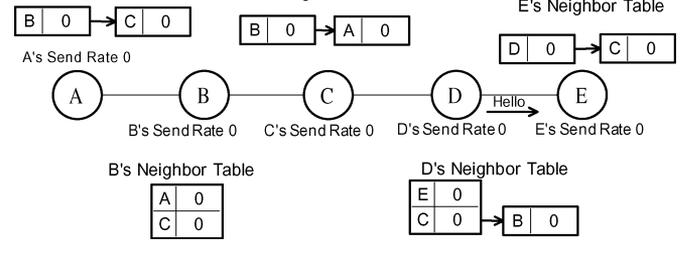


Fig. 13. QoS-aware routing with "Hello" bandwidth estimation route maintenance procedure 8.

Once host C detects the broken route between C and D, it first brings down the route that is recorded in the routing table, and at the same time it updates its bandwidth consumption register. Then, it sends an "Immediate Hello" to its neighbors to inform them of the host's update, as shown in Fig. 6. Host B updates its neighbor cache after receiving C's "Immediate Hello," and C's consumed bandwidth changes from 0.4 to 0. Right after sending the "Immediate Hello" message, C creates an "Error" message to inform its upstream hosts that the route between C and D is broken.

Host B sends a "Hello" message, which was triggered by the "Immediate Hello" received from C, to its neighbors A and C. Host A updates its neighbor cache record about C (from 0.4 to 0), as shown in Fig. 7. Host B also receives the "Error" message from C; therefore, B marks the corresponding route as invalid, updates its bandwidth consumption register (from 0.4 to 0) as shown in Fig. 8, and sends an "Immediate Hello." Both A and C change their neighbor caches regarding B's update, after receiving the "Immediate Hello." Of course, B sends an "Error" message to A right after the "Immediate Hello," as shown in Fig. 9.

Once A gets the “Error” message from B, A tears down the corresponding route in its routing table, updates its record about its own consumed bandwidth, and sends an “Immediate Hello” to B as shown in Fig. 10. B updates its record about A’s consumed bandwidth in its neighbor cache, then sends the triggered “Hello” as shown in Fig. 11. C updates its neighbor cache item about host A after receiving the “Hello” message from B. Therefore, the bandwidth used by the broken route is released correctly in hosts A, B, and C.

The bandwidth releasing in D and E is done during the route discovery procedure. Once C receives the RREQ, it sends an “Immediate Hello” first, then broadcasts the RREQ, as shown in Fig. 12. Therefore, D can update its neighbor cache before receiving the RREQ, and so can host E, as shown in Fig. 13.

IV. SIMULATIONS AND DISCUSSIONS

To test the performance of our QoS-aware routing protocol, we ran simulations using ns-2. We use the IEEE 802.11 MAC protocol in RTS/CTS/Data/ACK mode with a channel data rate of 2 Mb/s. The packet size used in our simulations is 1500 bytes. The topologies vary according to the different simulation purposes.

A. “Hello” Versus “Listen” Bandwidth Estimation When Routes Break

A broken route can be caused by two reasons: 1) the hello messages collide several times (in which case the route is not really broken) and 2) a host in that route moves out of its neighbor’s transmission range. We study these two different cases separately.

1) *Route Break Caused by Losing “Hello” Messages:* One flow in a network can be viewed as a single static chain. In order to simplify our analysis, we do the simulations in a chain topology to explain the effects brought by a broken route that is caused by losing broadcasted “Hello” messages. The simulated chain topology is composed of six hosts, where the header host is the source host and the tail host is the destination host. The source host sends data packets to the destination host using a 0.35 Mb/s feeding rate. By studying the trace files, we find that a supposed route break occurs at 13 s using the QoS-aware routing protocol with “Listen” bandwidth estimation. Supposed route breaks occur at 27, 73, 236, and 468 s using the QoS-aware routing protocol with “Hello” bandwidth estimation. Fig. 14 shows that using the route maintenance procedure described in Section III-C, “Hello” bandwidth estimation can correctly estimate the residual bandwidth after the reported route breaks; however, using “Listen” bandwidth estimation cannot, so the source host is forced to transmit below the channel capacity.

In this case, “Hello” packets are dropped often when traffic becomes heavy. After three consecutive “Hello” packets are dropped, a broken route is claimed. However, this route is not physically broken, because these three “Hello” messages are dropped by coincidentally colliding with other packets. Therefore, the packets are still successfully transmitted to the destination host during the time between the first “Hello” message being dropped and the third “Hello” message being dropped. The route discovery procedure is initiated right after the source host receives the “Error” message. The time interval

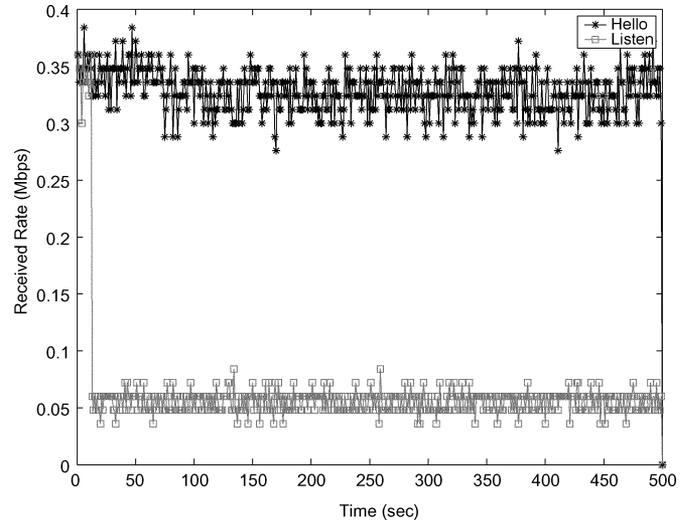


Fig. 14. Received packet rate using a six-node chain topology with “Listen” bandwidth estimation and “Hello” bandwidth estimation.

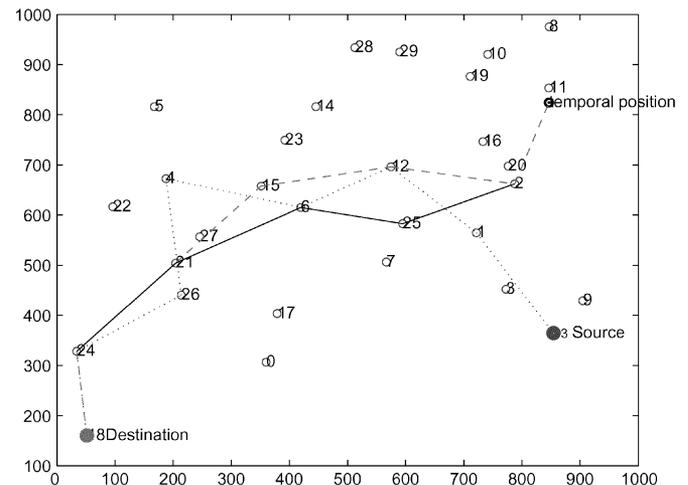


Fig. 15. Scenario used to simulate a route break caused by a moving node.

between claiming a route break and setting up the route is only several milliseconds. In such a small time interval, it is almost impossible for the hosts to automatically and correctly update their bandwidth registers in the “Listen” bandwidth estimation method, since the consumed bandwidth estimation is based on averaging bandwidth consumption every 1 s interval and the hosts in the broken route were transmitting data in the previous second. Therefore, the “Listen”-based bandwidth estimation approach has difficulty correctly estimating the residual bandwidth. Even if some forced update schemes can be adopted, the hosts still cannot release the bandwidth correctly, since the hosts do not know how much bandwidth each node in the broken route consumes. In contrast, the “Hello”-based bandwidth estimation approach can easily solve this problem by using the forced update scheme.

2) *Route Break Caused by Moving Out of a Neighbor’s Transmission Range:* To simplify the explanation, we use the topology shown in Fig. 15 to mimic the topology that will cause a route break because of a moving node. The topology is composed of 30 hosts. Host 18 is the destination host, and host 13 is the source host. Host 13 is moving toward host 11 with a

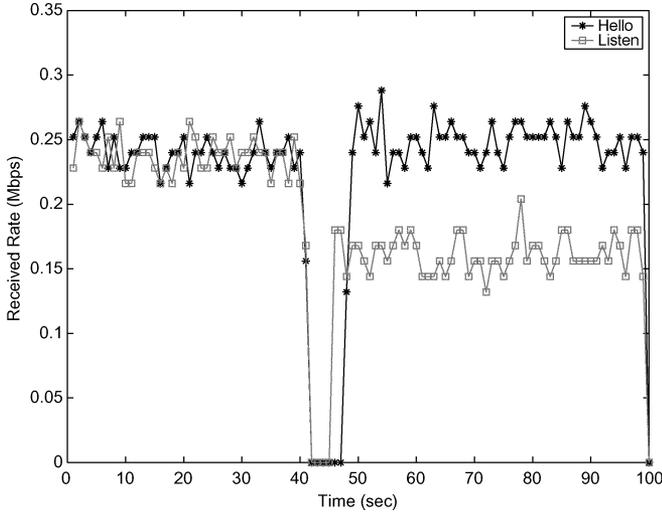


Fig. 16. Received rate using the source moving topology shown in Fig. 15 for the “Hello” bandwidth estimation method and the “Listen” bandwidth estimation method.

speed of 10 m/s. The source host sends data packets to the destination host using a 0.25 Mb/s sending rate. We ran simulations using the QoS-aware routing protocol with “Listen” bandwidth estimation and the QoS-aware routing protocol with “Hello” bandwidth estimation. In the beginning of the simulation, the chosen route goes through hosts 13, 1, 12, 6, 4, 26, 24, and 18 (the dotted line in Fig. 15). At the simulation time of 43 s, host 13 moves to a position (shown in Fig. 15) that is out of host 1’s transmission range. This causes a route break and host 13 must initiate a new discovery procedure. Using the routing protocol based on using “Listen” to estimate residual bandwidth, the new route goes through hosts 13, 2, 12, 15, 21, 24, and 18 (the dashed line in Fig. 15). Using the routing protocol based on using “Hello” to estimate residual bandwidth, the new route goes through host 13, 2, 25, 6, 21, 24, and 18 (the solid line in Fig. 15). The simulation results are shown in Fig. 16. We can see the end to end throughput using “Listen” to estimate bandwidth is much less than by using “Hello” to estimate bandwidth after the route changes. Studying the trace file, we find the reason for this difference is that there are approximately 3 s between host 13 moving out of host 1’s transmission range and the route break being claimed. During these 3 s, all hosts correctly update their bandwidth consumption registers except host 2 who is next to the source. This is caused by the fact that the source host keeps on sending RTS packets, so host 2 can hear all these RTS packets and sets its NAV vector according to the packet length that the RTS indicates. Therefore, its estimated free time is significantly less than the real free time. Thus, host 2 cannot offer the correct bandwidth estimation after receiving a “RREQ” packet. However, using “Hello” to estimate residual bandwidth will not be affected by the above reason.

These results show that the “Listen” technique cannot react well to a broken route due to the fact that the MAC’s NAV cannot truly reflect the traffic status, and the bandwidth consumption registers cannot be updated in time. Thus, when routes break, “Hello” bandwidth estimation performs better than “Listen” bandwidth estimation.

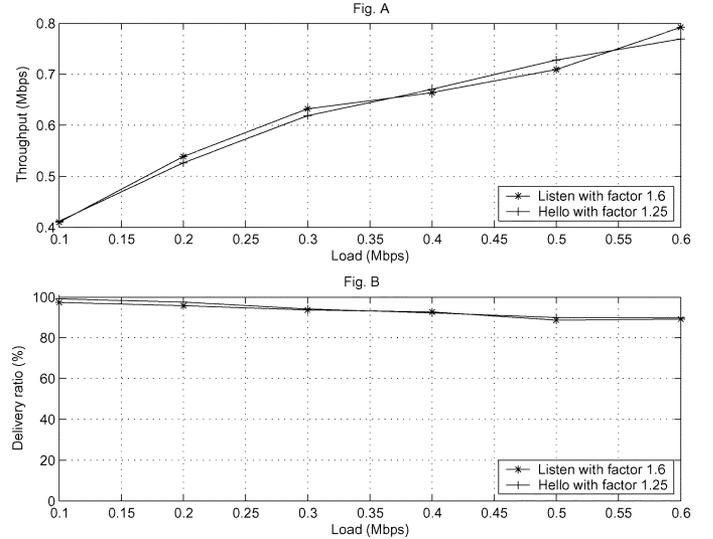


Fig. 17. Throughput and packet delivery ratio comparison (“Listen” versus “Hello”). (A) Throughput (“Listen” versus “Hello”). (B) Delivery ratio (“Listen” versus “Hello”).

B. Weight Factor Comparison

We cannot compare the performance of “Hello” bandwidth estimation and “Listen” bandwidth estimation using the same weight factor, because these two methods define the consumed bandwidth differently.

- “Listen” mode—accounts for RTS, CTS, ACK, retransmission, routing packets, and transmitted packets.
- “Hello” mode—counts the transmitted packets only.

Therefore, the “Hello” weight factor should be larger than the “Listen” weight factor if we want to get the same performance. In addition, if congestion occurs, the listen mode cannot release the bandwidth immediately, so we should choose a large weight factor to avoid congestion when we compare these two different estimation methods.

If we do not consider the bandwidth used in the retransmission of data, RTS and CTS packets, and the bandwidth used for transmitting the routing packets, the approximate ratio between the weight factor used in “Listen” mode and the weight factor used in “Hello” mode should be as follows:

$$\frac{\text{RTS} + \text{CTS} + (\text{Data} + \text{MACHdr} + \text{IPHdr}) + \text{ACK}}{\text{Data}} = \frac{44 + 38 + (1500 + 52 + 20) + 38}{1500} = 1.128. \quad (1)$$

Therefore, if we randomly choose the weight factor of “Listen” mode as 1.25, which is large enough to avoid the route breaks caused by losing “Hello” messages, the weight factor used in “Hello” mode should be larger than $1.25 \times 1.128 = 1.41$.

We investigate the performance of the “Hello” scheme and the “Listen” scheme using topologies where 50 static nodes are located randomly in 1000 m × 1000 m. Five nodes are randomly chosen as sources and five nodes are randomly choose as destinations. All sources feed the same data rate to their destinations, and the feeding rate varies from 0.1 to 0.6 Mb/s. After every 10 s interval, one source will begin to send data into the network. We randomly choose 20 different scenarios and run the simulation for 500 s. The average simulation results are as shown in Fig. 17.

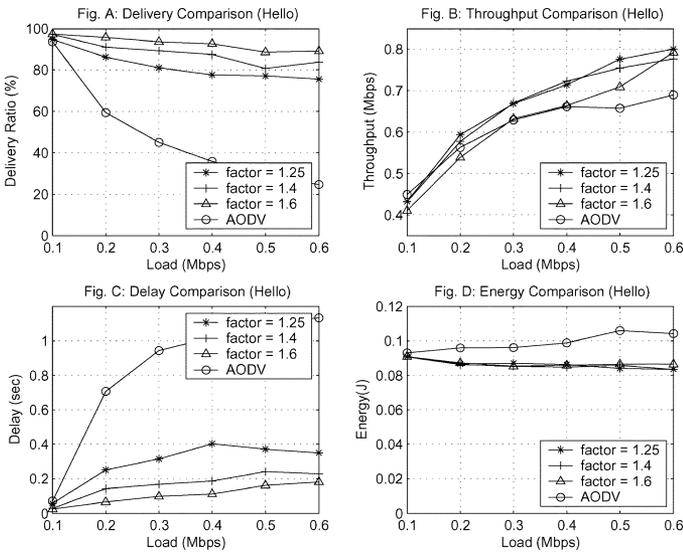


Fig. 18. Results for QoS-aware routing with “Hello” bandwidth estimation with different weight factors and AODV. (A) Packet delivery ratio. (B) End-to-end throughput. (C) Delay. (D) Energy.

We find that the performance of choosing weight factor 1.6 in “Hello” mode matches well with the performance of choosing weight factor 1.25 in “Listen” mode. Therefore, we deduce that the QoS-aware routing protocols based on “Listen” bandwidth estimation and “Hello” bandwidth estimation work equally well as long as their weight factors are chosen appropriately.

The RTS, CTS, and ACK overheads affect differently small size packets and large size packets. Therefore, different weight factors should be used for different packet sizes. In addition, different physical phenomena can bring different fading errors. The fading errors can cause necessary retransmission of RTS and data packets. Thus, these overheads may change the required weight factor’s value. However, in this paper we use the same physical channel for all the simulations.

C. Static Topology Using the Adaptive Feedback Scheme

For these simulations, we use the same topologies and simulation time as used in Section IV-B, and we compare QoS-aware routing with “Hello” bandwidth estimation, QoS-aware routing with “Listen” bandwidth estimation, and conventional AODV, which has no QoS support. The metrics used in measuring the protocols’ performance are delay, packet delivery ratio, energy consumption per packet per hop, and overall end-to-end throughput.

As the number of flows and the number of hosts increases, the negative effects brought by using “Listen” bandwidth estimation under a broken route will not be very significant. In the case that the broken route is caused by losing “Hello” messages, the underestimated bandwidth will be consumed by other flows. Therefore, we expect that both “Listen” and “Hello” bandwidth estimation will work well. Fig. 18 shows the performance using the QoS-aware routing protocol with “Hello” bandwidth estimation and AODV. Fig. 18(A) shows that there is a great improvement in packet delivery ratio (up to 260%) using QoS-aware routing with “Hello” bandwidth estimation compared with AODV.

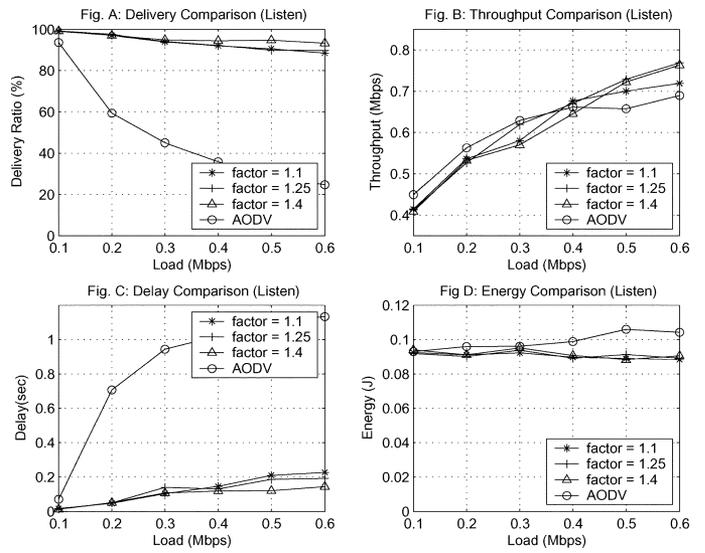


Fig. 19. Results for QoS-aware routing with “Listen” bandwidth estimation with different weight factors and AODV. (A) Packet delivery ratio. (B) End-to-end throughput. (C) Delay. (D) Energy.

We also find that the packet delivery ratio increases with increasing weight factor. This is because the available bandwidth allowed to schedule packet transmissions is $ResidualBandwidth/WeightFactor$. The bigger the weight factor is, the more conservative the packet transmission scheduling is. Therefore, there is a tradeoff between bandwidth usage and the packet delivery ratio. However, as bandwidth usage is one of the most important metrics to measure the network performance, we do not want to completely sacrifice bandwidth to get an improvement in packet delivery ratio. Fig. 18(B) shows that actually we can get almost equal overall end-to-end throughput for QoS-aware routing with “Hello” bandwidth estimation compared with AODV, and even some improvement in a highly congested network, when choosing a reasonable weight factor.

The packet delivery ratio improvement also brings side benefits such as decreased delay and energy consumption, due to congestion avoidance and the control nature inherited in the QoS-aware routing protocol. The time used waiting in the packet queue and contending for the channel decreases, and the energy used on transmitting packets which will ultimately be dropped is saved. Therefore, delay is decreased up to 795% and energy/packet/hop is decreased up to 29%, as shown in Fig. 18(C) and (D).

Fig. 19 shows the performance when the QoS-aware routing protocol with “Listen” bandwidth estimation is used compared with AODV. Fig. 19(A) shows the great improvement in packet delivery ratio (up to 280%) using QoS-aware routing with “Listen” bandwidth estimation compared with AODV. However, the end-to-end throughput is decreased by 10% as shown in Fig. 19(B), when the feeding rate is low, even when the weight factor is quite small (e.g., 1.1). Our best guess is that the possibility of route breaks caused by losing “Hello” messages is high in some scenarios, when a small weight factor is used. Using the QoS-aware routing protocol with “Listen” bandwidth estimation underestimates the bandwidth after a

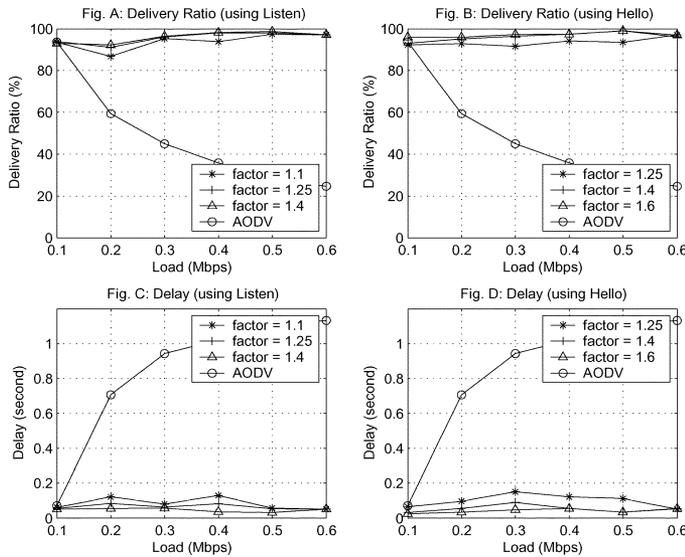


Fig. 20. Results for QoS-aware routing using the admission scheme with different weight factors and AODV. (A) Packet delivery ratio using “Listen.” (B) Packet delivery ratio using “Hello.” (C) Delay using “Listen.” (D) Delay using “Hello.”

route break, and the residual bandwidth cannot fully be used by other flows. Therefore, the end-to-end throughput is lower than using AODV. However, the underestimated bandwidth can be used when the load is high; therefore, there is bandwidth improvement when the load is high. There are also some side benefits brought by the improvement of the packet delivery ratio. The delay is decreased up to 800% and energy/packet/hop is decreased up to 22%, as shown in Fig. 19(C) and (D).

D. Static Topology Using the Admission Scheme

The other scheme incorporated into our QoS-aware routing protocol is the admission scheme. In the admission scheme, flows are denied if there is not enough bandwidth available to support their request. This results in the total capacity of the admitted flows being less than that of the feedback scheme, so packet collisions occur less frequently. Thus, we expect that the packet delivery ratio using the admission scheme should be larger than that of using the feedback scheme. Correspondingly, the packet delay should be decreased significantly due to fewer collisions. We use the same topologies as in Section IV-C, and we obtain the simulation results shown in Fig. 20. Using QoS-aware routing, the packet delivery ratio remains constantly above 90%, and the delay remains lower than 0.17 s.

E. Mobile Topology

Our routing protocol is designed with the restriction of combinatorial stability. Therefore, if the network changes too fast, we do not expect the QoS-aware routing protocol to perform well. Thus, we choose low mobility scenarios that mimic pedestrian speeds to test our protocol. In the scenarios we choose, each node moves toward a random destination using a speed randomly chosen between 0–3 m/s. Five random source-destination pairs send packets using a requested rate between 0.1 and 0.6 Mb/s. The simulation time is 500 s. Fig. 21 shows the results obtained by averaging ten different

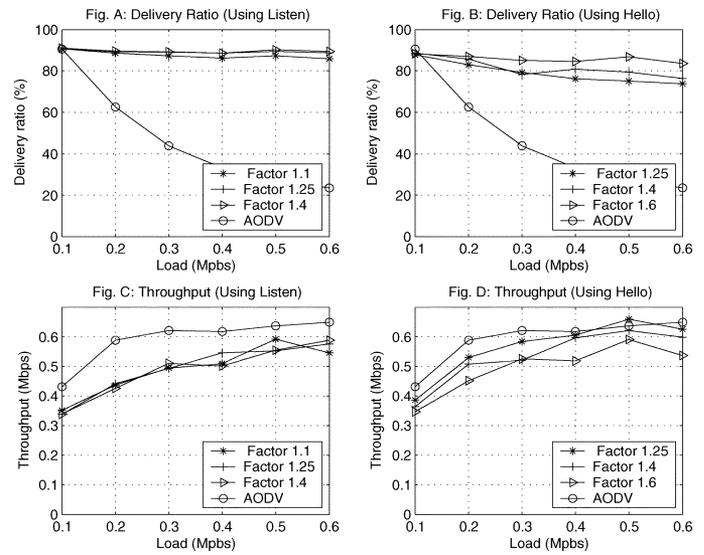


Fig. 21. Results for QoS-aware routing using mobile topologies (maximum speed of 3 m/s) with different weight factors and AODV. (A) Packet delivery ratio using “Listen.” (B) Packet delivery ratio using “Hello.” (C) end-to-end throughput using “Listen.” (D) End-to-end throughput using “Hello.”

scenarios. The packet delivery ratio is between 85%–90% using the QoS-aware routing protocol with “Listen” bandwidth estimation, and the packet delivery ratio is between 75%–90% using the QoS-aware routing protocol with “Hello” bandwidth estimation. QoS-aware routing shows great improvement over using AODV, which achieves very low packet delivery ratio for high requested loads. As there is a tradeoff between packet delivery ratio and throughput that we discussed previously, the higher the packet delivery ratio, the lower the achievable throughput. Therefore, using the “Listen” scheme, the end-to-end throughput is slightly decreased compared with using the “Hello” scheme, as shown in Fig. 21.

We would expect that the QoS-aware routing protocol’s performance will degrade as the moving speed increases, because we designed the QoS-aware routing protocol with a model of low mobility. Therefore, we did not incorporate any predictive scheme to find a new route before the old route is broken. This results in very long transient time when the required QoS is not guaranteed, due to a route break or network partition, which significantly decreases the packet delivery ratio. However, our QoS-aware routing protocol still gets relatively higher packet delivery ratio compared with AODV, as shown in Fig. 22. The “Hello” scheme’s performance is better than the “Listen” scheme’s performance in term of end-to-end throughput, while the “Listen” scheme’s performance is better than the “Hello” scheme’s performance in term of packet delivery ratio.

V. RELATED WORK

Sinha *et al.* [16] have proposed a core-extraction distributed algorithm (CEDAR). In the CEDAR routing protocol, it is assumed that the available bandwidth is known. The available bandwidth is disseminated among the cores. In this way, the overhead used to propagate the link state information can be minimized. However, if the core is moving out of the selected route, rerouting is very costly. Chen *et al.* [1] have proposed a

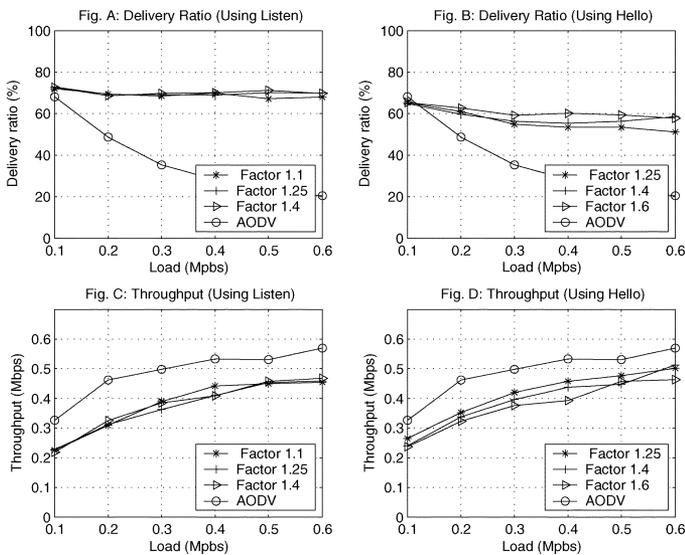


Fig. 22. Results for QoS-aware routing using mobile topologies (maximum speed of 20 m/s) with different weight factors and AODV. (A) Packet delivery ratio using “Listen.” (B) Packet delivery ratio using “Hello.” (C) End-to-end throughput using “Listen.” (D) End-to-end throughput using “Hello.”

ticket-based probing algorithm with imprecise state. The bandwidth and delay information are also assumed available. This algorithm tries to limit flooding by issuing limited tickets. Both CEDAR and ticket-based probing emphasize minimizing overhead used in setting up the route.

Ge *et al.* [17] have proposed a proactive QoS routing, which is based on OLSR, for static networks. They use monitoring of the channel’s idle time to measure available bandwidth. Their work focuses on correctly finding the maximum bandwidth path, without considering route breaks and mobile topologies. Another proactive QoS routing protocol is proposed in [18], called adaptive dispersity QoS Routing (ADQR). The ADQR routing protocol is based on SPAFAR [19] and chooses the longer-lived connection based on signal strength. With the knowledge of all the routes available to the destination, rerouting can be done before the old route is broken. As proactive routing consumes bandwidth on routing table maintenance, the overhead is a big concern. However, preselecting alternative routes becomes easy in proactive routing protocols.

All the above approaches do not consider that the supported bandwidth should be less than the bandwidth available during the route discovery, which is caused by the potential bandwidth sharing brought by the new routes. In [20], Xue and Ganz proposed AQDR, which addresses the bandwidth sharing among the neighbors in the new route. However, they do not consider the bandwidth consumption caused by the interference during the available bandwidth estimation. In addition, AQDR does not consider the underestimated bandwidth situation caused by a broken route, as discussed in Section IV-A.

Many QoS routing protocols use time-division multiple access (TDMA) as the underlying MAC. Lin *et al.* [21] have proposed an available bandwidth calculation algorithm for ad hoc networks with TDMA. By exchanging routing tables, the bandwidth in the shortest path can be determined. In [22], Lin focuses on finding a feasible bandwidth route instead of

the shortest-path route. A QoS-aware routing protocol based on AODV using TDMA is also proposed in [23].

VI. CONCLUSION AND FUTURE WORK

This paper proposes incorporating QoS into routing, and introduces bandwidth estimation by disseminating bandwidth information through “Hello” messages. A cross-layer approach, including an adaptive feedback scheme and an admission scheme to provide information to the application about the network status, are implemented. Simulations show that our QoS-aware routing protocol can improve packet delivery ratio greatly without impacting the overall end-to-end throughput, while also decreasing the packet delay and the energy consumption significantly.

We have compared two different methods of estimating bandwidth. The “Hello” bandwidth estimation method performs better than the “Listen” bandwidth estimation method when releasing bandwidth immediately is important. The “Hello” and “Listen” schemes work equally well in static topologies by using large weight factors to reduce the congestion and minimize the chance of lost “Hello” messages incorrectly signaling a broken route. In a mobile topology, “Hello” performs better in term of end-to-end throughput, and “Listen” performs better in term of packet delivery ratio. From the perspective of overhead, “Listen” does not add extra overhead, but “Hello” does add overhead by attaching neighbors’ bandwidth consumption information in the “Hello” messages.

In our protocol, we have not incorporated any predictive way to foresee a route break, which causes a performance degradation in mobile topologies. Therefore, some methods such as preemptive maintenance routing [24] and route maintenance based on signal strength [18] might help to reduce the transient time when the required QoS is not guaranteed due to a route break or network partition, so that the routing protocol can react much better to mobile topologies.

The accurate measurement of the capacity of a multihop mobile network is an open issue right now. Further study of the 802.11 MAC layer’s behavior could be helpful to understand this capacity issue. Also, in a real scenario, shadowing will cause a node’s transmission range to vary, and it will not be the ideal circle that is assumed here. How to incorporate these non-idealities into our protocol is the subject of our future research.

Furthermore, incorporating different transmission ranges among all the hosts and analyzing fairness among the hosts will be explored in our future work. Our ultimate goal is to provide a model from the application layer to the MAC layer for supporting service differentiation. A transport layer protocol to support different data streams, queue management and a QoS-supported MAC will be addressed in our future work.

REFERENCES

- [1] S. Chen and K. Nahrstedt, “Distributed quality-of-service in ad hoc networks,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, Aug. 1999.
- [2] S. Chakrabarti, “QoS issues in ad hoc wireless networks,” *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 142–148, Feb. 2001.
- [3] Y. Zhang, “Very low bit rate video coding standards,” in *Proc. Vis. Commun. Image Process.*, vol. 2501, May 1995, pp. 1016–1023.

- [4] G. Coete, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 849–866, Nov. 1998.
- [5] S. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt, "Multiple-description wavelet based image coding," in *Proc. IEEE Int. Conf. Image Process.*, 1998, pp. 659–663.
- [6] "Ad hoc on demand distance vector (AODV) routing," IETF Internet Draft, Work in Progress, [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-manet-adov.03.txt>, Jun. 1999.
- [7] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop Mobile Comput. Syst. Appl.*, 1999, pp. 90–100.
- [8] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Comput.*, pp. 153–181, 1996.
- [9] C. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM'94*, Oct. 1994, pp. 234–244.
- [10] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. INFOCOM*, Apr. 1997, pp. 1405–1413.
- [11] S. Chen, "Routing support for providing guaranteed end-to-end quality-of-service," Ph.D. dissertation, Univ. Illinois at Urbana–Champaign, Urbana–Champaign, IL, 1999.
- [12] P. Mohapatra, J. Li, and C. Gui, "QoS in mobile ad hoc networks," *IEEE Wireless Commun. Mag. (Special Issue on QoS in Next-Generation Wireless Multimedia Communications Systems)*, pp. 44–52, 2003.
- [13] J. Li, C. Blake, D. D. Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. 7th ACM Int. Conf. Mobile Comput. Netw. (MobiCom'01)*, 2001, pp. 61–69.
- [14] Y. Yang and R. Kravets, "Contention-aware admission control for ad hoc networks," Univ. Illinois at Urbana–Champaign, Urbana–Champaign, IL, Tech. Rep. 2003-2337, 2003.
- [15] K. Sanzgiri, I. Chakeres, and E. Belding-Royer, "Determining intra-flow contention along multihop paths in wireless networks," in *Proc. Broadnets 2004 Wireless Netw. Symp.*, Oct. 2004, pp. 611–620.
- [16] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," in *IEEE INFOCOM'99*, New York, Mar. 1999, pp. 1454–1465.
- [17] Y. Ge, T. Kunz, and L. Lamont, "Quality of service routing in ad hoc networks using OLSR," in *Proc. 36th Hawaii Int. Conf. Syst. Sci.*, 2003, pp. 300–308.
- [18] Y. Hwang and P. Varshney, "An adaptive routing protocol for ad hoc networks using multiple disjoint path," in *Proc. Veh. Technol. Conf.*, May 2003, pp. 2249–2253.
- [19] Y. Hwang, H. Lee, and P. Varshney, "An adaptive QoS routing protocol with dispersity for ad hoc networks," in *Proc. 36th Hawaii Int. Conf. Syst. Sci.*, 2003, pp. 302–311.
- [20] Q. Xue and A. Ganz, "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks," *J. Parallel Distrib. Comput.*, pp. 154–165, 2003.
- [21] C. R. Lin and J. S. Liu, "QoS routing in ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1426–1438, Aug. 1999.
- [22] C. R. Lin, "On-demand QoS routing in multihop mobile networks," in *Proc. IEEE INFOCOM*, 2001, pp. 1735–1744.
- [23] C. Zhu and M. Corson, "QoS routing for mobile ad hoc networks," in *IEEE INFOCOM*, 2002, pp. 958–967.
- [24] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Pre-emptive maintenance routing in ad hoc networks," in *Proc. MobiCom*, 2001, pp. 43–52.



Lei Chen (S'03) received the B.S. degree in information science and electronics engineering from Zhejiang University, Zhejiang, China, in 1999 and the M.S. degree in electrical and computer engineering from the University of Rochester, Rochester, NY, in 2002. She is currently working towards the Ph.D. degree in electrical and computer engineering at the University of Rochester.

Her current research interests lie in the areas of QoS-aware architectures and congestion control in ad hoc networks.



Wendi B. Heinzelman (S'95–M'01) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1997 and 2000, respectively.

She is an Assistant Professor in the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY. Her current research interests lie in the areas of wireless communications and networking, mobile computing, and multimedia communication.

Dr. Heinzelman received the National Science Foundation (NSF) CAREER Award in 2005 for her research on cross-layer architectures for wireless sensor networks. She is a member of Sigma Xi and the Association for Computing Machinery (ACM).