Delay-Sensitive Routing in Multi-rate MANETs

Chia-Cheng Hu

cchu@cna.edu.tw

Department of Information Management, Naval Academy, Kaohsiung, Taiwan

Abstract—A mobile ad hoc network (MANET) is a collection of mobile hosts, which can communicate by the aids of intermediate mobile hosts without utilizing the fixed infrastructure and centralized administration. Many MANET standards, such as 802.11a, 802.11b, and 802.11g, can be operated at various rates for Quality-of-Service (QoS) constrained multimedia communication to more efficiently use the limited resources of MANETs. Since the radio channel is shared among neighbors in MANETs, calculating one-hop delays and determining delay-sensitive routes using the IEEE 802.11 MAC are still two challenging problems. In this paper, we first exploit the busy/idle ratio of the shared channel to estimate one-hop delay based on varied data rates. Then by the aid of the estimated delay, a multi-rate routing protocol is proposed for selecting data rates and determining a route for admitting a flow with a requested delay. In MANETs, when a host is transmitting data packets, its neighbors are blocked (i.e., forbidden to send packets) since it shares the radio channel with its neighbors. We adopt the strategy by selecting the combination of data rates and a route in order to minimize the total blocking time to all hosts of the network for maximizing the network capacity, which is the number of flows admitted by the network. Simulation results show that the proposed method obtains more precise one-hop delay than a very recent work, and the proposed protocol admits more flows than an existing protocol.

Keywords**:** mobile ad hoc networks, multi-rate, delay, routing.

1 Introduction

In a mobile ad hoc network (MANET), the hosts can communicate with one another by sharing the common radio channel. Since there are more and more applications relaying on real-time routing services, such as VoIP and videoconference, routing protocols in MANETs should provide services with delay requirements. To build a route for satisfying the delay requirement, one-hop delay (i.e., the transmission time between two neighboring hosts connected by a link) and end-to-end delay (i.e., the transmission time from the source to the destination along the route) must be known. However, how to calculate the above two delays using the IEEE 802.11 MAC is still a challenging problem, because a host in MANETs shares the radio channel with its neighbors.

Nowadays, many MANET standards, such as 802.11a, 802.11b, and 802.11g, can be operated at various data rates to use more efficiently the limited resources of MANETs. The multi-rate enhancements make it more difficult to calculate the above two delays. There are several Quality-of-Service (QoS) routing protocols [1-3] which focus on satisfying delay requirements. But, they use only a single base rate. Some mechanisms for rate selection have been proposed in several routing protocols [4-5] for multi-rate MANETs. Their objectives aim to select a route which is composed of hops by using the highest available data rate. However, the selected routes in these protocols can not satisfy a required delay, because the two delays are not estimated. Besides, all QoS routing protocols [1-5] mentioned above will suffer from one problem as described below.

When a new flow with delay requirement is requesting admission, a control packet from the source is flooded in order to determine a delay-satisfied route. Each host in the neighborhood of some ongoing flows may be determined as a forwarder for the new flow if it does not induce delay violation of it and its neighbors. However, even so, delay violation may happen to its neighbors because it fails to take into account the resource consumption of those hosts that are two hops distant from it. This induces a new delay-violation problem in MANETs. The problem is named the hidden route problem (HRP).

Refer to Figure 1, where an illustrative example for HRP is shown. There are two ongoing flows from *e* to *f* and from *g* to *h*, respectively. A new flow from *a* to *d* is permitted, and the route determination for the new flow proceeds to *c*. If *c* serves as a forwarder, then delay-violation happened to *c* and *e* should be avoided. However, the routing protocols [1-5] do not consider the resource consumption of *e* induced by *g* since that *c* is not aware of the ongoing flow from *g* to *h* when it is determined to be a forwarder. Without considering the resource consumption of *e* induced by *g* will cause HRP to the ongoing flow from *e* to *f* when *c* is determined to be a forwarder of the new flow.

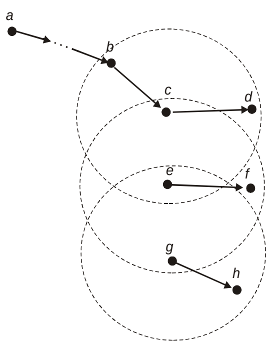


Figure 1. An example of HRP.

In MANETs, a host is a one-hop neighbor of another, if the former is within the transmission range of the latter. Further, a host is a two-hop neighbor of another, if the former is within twice the transmission range, but out of the transmission range, of the latter. HRP arises, as a consequence that the transmitters fail to estimate the resource consumption of their two-hop neighbors. When it happens, QoS (e.g., bandwidth or delay requirements) of ongoing flows cannot be satisfied. It was shown in [6] that when the network traffic was saturated, HRP happened frequently, which degraded the network performance considerably.

In this paper, a method by means of measuring the busy/idle ratio of the shared radio channel for estimating the one-hop delay and end-to-end delay in multi-rate MANETs is proposed. Then, a new routing protocol by using the delay estimation method is proposed for determining a delay-sensitive route. A delay-sensitive route can satisfy the delay requirement of the requesting service with a certain confidence level, i.e., a certain percentage of the data packets whose transmitted end-to-end delays should be smaller than the required delay. Also, the proposed routing protocol can avoid HRP by further considering the resource consumptions of two-hop neighbors.

Further, since higher data rates can reduce one-hop delay, most of the previous routing protocols aim to determine the routes with the highest data rate. But, higher data rates also induce shorter transmission range. For example, in IEEE 802.11b, the transmission range is around 30m at 11Mbps, 70m at 5.5Mbps, 90–100m at 2 Mbps and 110–130m at 1 Mbps [17]. Thus, more forwarders along the determined routes are needed to participate in packet forwarding if high data rates are used. When a forwarder is transmitting data packets, its neighbors are blocked (i.e., they can not transmit any packet within the period). So, more forwarders decline network performance since more hosts are blocked.

We seek a compromise between the data rates and the number of the neighbors of the forwarders to alleviate the performance degradation. We aim to select the combination of data rates and a route in order to minimize the total medium time, which is to sum the one-hop delay of the forwarders and the blocking time on all the neighbors of the forwarders, instead of the strategy adopted in the most previous works by selecting the highest data rate. A route with less total minimal medium time can reduce the resource consumption to the network so that the network capacity is increased, i.e., more flows are admitted into the network.

The next section will first review the previous works. Section 3 presents the proposed routing protocol. We provide simulation results in Section 4, where we validate that our method obtains more precise one-hop delay than a very recent work, show our HRP-prevented ability, and represent our medium time routing metric. Finally we conclude in Section 5.

2 Related works

There are several QoS routing protocols [1-3] which focus on satisfying delay requirements by using a single base rate. In [1], a routing protocol, Ad hoc Qos On-demand Routing (AQOR), proposes a model, which estimates end-to-end delay, for admission control in determining QoS-satisfied routes. In AQOR, the source estimates the end-to-end delay of the route by sending some probing packets to its destination. The end-to-end delay is estimated by half of the round trip delay, which is the transmission time of the probing packets from the source to the destination and back to the source. In [3], a routing protocol, Core-Extraction Distributed Ad hoc Routing (CEDAR), selects the routes constituted by stable links with high available bandwidth.

However, since CEDAR did not estimate the end-to-end delay and AQOR did not take the newly admitted flows into consideration, while estimating the end-to-end delay, they failed to provide delay satisfaction. Besides, they might suffer from HRP.

In [2, 7-9, 14], the authors developed detailed methods to estimate the above two delays. In [2, 7], the authors analyze explicitly the behavior of IEEE 802.11 protocol according to different traffic load, and use a Markov-modulated Poisson process model to calculate the average one-hop delays of the IEEE 802.11 channel. However, the model assumed that the capacity provided by a wireless link between two nodes is constant. This is not true in wireless network environment since the capacity is a random variable.

In [8-9], the authors proposed an approximate model to estimate the one-hop delay by computing the collision probability while transmitting a packet. In [14], the authors also propose a model to estimate the one-hop delay by monitoring the ratio of busy and idle periods of the shared channel. Of course, the models proposed in [8-9, 14] can obtain more precise one-hop delay than those proposed in [2, 7] because they estimate the delay based on the current channel condition. This work bases on a concept similar to that applied in [14] to estimate the two delays by monitoring the current channel condition.

To increase network capacity, two routing protocols [4-5] are proposed, which use the multi-rate enhancement of MANETs. In [4], a mathematical model is proposed for data rate selection and route determination. However, it is assumed to be based on a specific MAC type, i.e., pure ALOHA type. In [5], the authors present a routing protocol based on a new metric, i.e., the interference time, for every link in multi-rate multi-radio mesh networks. The interference time of a link is defined as the total interference time induced by its neighboring links that can interfere on it. The routing protocol aims to determine a route by selecting the links with smaller interference time. Although some rate adaption mechanisms are proposed in the protocols [4-5], they can not satisfy a required delay since one-hop delay and end-to-end delay are not estimated properly.

3 Protocol

The proposed protocol is based on the following assumptions: (1) A MANET based on IEEE 802.11 that can support multiple data rates is considered. (2) A single physical channel is available for transmission among multiple contending mobile hosts. (3) A Carrier Sensing Medium Access (CSMA) protocol in MAC layer is used. (4) A PHY layer at a host is assumed to be able to monitor the status of the physical channel, which is perceived as either idle or busy. The channel is idle if the host is not transmitting/receiving a data/control packet, or does not sense a busy carrier with a signal strength that exceeds a threshold. (5) Data packets are assumed to be able to be transmitted at different rates, but control packets are transmitted at the base rate, which is the lowest of all possible rates. For example, 1 Mbps is selected as the base rate in IEEE 802.11b. (6) The carrier sense range is approximately twice of the transmission range.

The transmission range is a distance from a transmitter within which a receiver can receive and correctly decode the packet. The carrier sensing range is another distance from a transmitter within which a receiver can sense the signal but cannot decode the packet. When a transmitter is transmitting packets, all hosts within its carrier sense range are blocked. The carrier sense range is a distance from the transmitter within which the hosts can sense the signal but cannot decode the packets correctly. It was shown in [11-13] that the carrier sense range should be approximately twice the transmission range. Thus we assume that the carrier sense range is twice the transmission range.

The transmission range is affected by the data rate used by the transmitter. When the transmitter uses a higher data rate to transmit data packets, the higher data rate results in a smaller transmission range to cover fewer hosts who can receive and correctly decode the transmitted packets, and vice versa. Suppose that there are n possible data rates, denoted by r^1, r^2, …, r^n, and $r^1 < r^2 < ... < r^n$. We define v_j as an r^t-one-hop neighbor of v_i if v_j is within the transmission range of v_i by using r^t, and v_k is defined as a r^t-two-hop neighbor of v_i if v_k is a r^t-one-hop neighbor of v_j but is not a r^t-one-hop neighbor of v_i, where $1 \leq t \leq n$. Thus, whenever a host is transmitting with r^t, all its r^t-one-hop/r^t-two-hop neighbors will remain silent.

This section aims to propose a routing protocol that can select data rates and determine a delay-sensitive route to admit a flow with delay requirement in multi-rate MANETs. For the purpose, one-hop and end-to-end delays must be known first. However, to estimating the two delays is still a challenging task in IEEE 802.11 MAC, because the channel is shared among neighbors. Besides, the neighboring relations among the hosts are varied with the data rates.

One the other hand, when a host is being determined as a forwarder to transmit a new flow using a data rate, the delay violation happened to the ongoing flows that are being transmitted by it or any of its one-hop/two-hop neighboring hosts should be avoided.

In Section 3.1, we propose a method for identifying the hosts that are the r^t-one-hop (r^t-two-hop) neighbors of a host. In Section 3.2, the method for estimating one-hop delays under different data rates is presented, which is based on continuous monitoring of the network load. Section 3.3 proposes an algorithm for selecting data rates and determining a delay-sensitive route for transmitting a request flow.

3.1 Determination of relations among neighbors

This section proposes a method for determining neighboring relation based on various data rates. Each host v_i maintains two neighbor tables: the one-hop neighbor table and the two-hop neighbor table. It generates an entry in the one-hop (two-hop) neighbor table for each of its one-hop neighbor (two-hop neighbor). We use v_j (v_k) to denote a one-hop neighbor (two-hop neighbor) of v_i. Let $t\_sinr_{i,j}$ ($r\_sinr_{j,i}$) denote the Signal to Interference-plus-Noise Ratio (SINR) value of the link from v_i (v_j) to v_j (v_i) when v_i transmits (receives) a packet to (from) v_j through the link. Similarly, let $t\_sinr_{j,k}$ denote the SINR value of the link from v_j to v_k. A link is directed from v_j to v_k if v_k is a one-hop neighbor of v_j. Each entry in the one-hop (two-hop) neighbor table contains six (five) variables. The two neighbor tables in host v_i are shown in Figure 2.

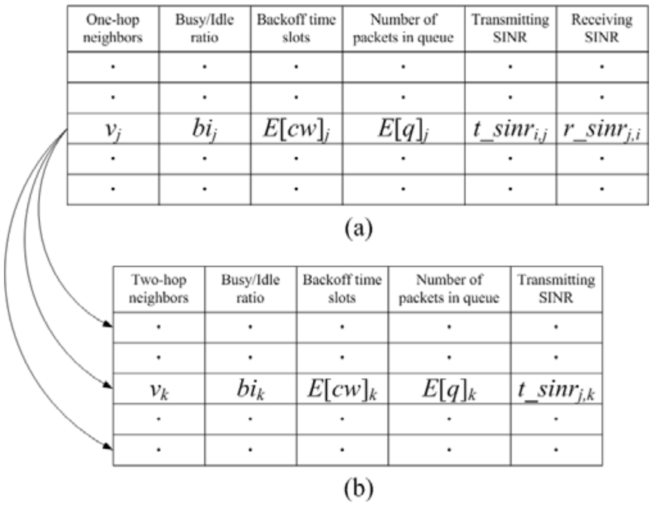


Figure 2. Two neighbor tables in host v_i (a) one-hop neighbor table (b) two-hop neighbor table

Since the receiver perceives channel quality in a more timely manner than a transmitter does, the Receiver-Based Auto Rate (RBAR) algorithm [10] (which is a receiver-based approach) yields significant throughput gains as compared with the Auto Rate Fallback (ARF) algorithm [15] (which is a transmitter-based approach). Based on a concept similar to that applied in RBAR, the three values $r\_sinr_{j,i}$, $t\_sinr_{i,j}$ and $t\_sinr_{j,k}$ are estimated by the receiver v_i, v_j and v_k, respectively. Once v_i receives a packet from v_j, $r\_sinr_{j,i}$ is estimated based on the received signal quality. Then, $r\_sinr_{j,i}$ is used for adding/updating the associated entry of one-hop neighbor table in v_i. Similarly, $t\_sinr_{i,j}$ ($t\_sinr_{j,k}$) is estimated in v_j (v_k) when a packet is received from v_i (v_j).

Let θ^t represent an SINR threshold at which the Corresponding Bit Rrror rate (BER) equals an acceptable working level. If the SINR value from a transmitter to a receiver exceeds θ^t, then the transmitter can transmit data packets to the receiver successfully by using r_t. Thus, v_i uses its one-hop neighbor table to determine its r^t-one-hop neighbors by the following algorithm: if $t\_sinr_{i,j} > \theta^t$, then v_j is its r^t-one-hop neighbor. Similarly, v_i uses its two-hop neighbor table to determine the r^t-one-hop neighbors v_ks of its r^t-one-hop neighbors v_js by checking whether $t\_sinr_{j,k} > \theta^t$. Its r^t-two-hop neighbors, v_ks, can be determined by checking that v_k is an r^t-one-hop neighbor of its r^t-one-hop neighbors but is not its r^t-one-hop neighbor.

The rest three variables in the two neighbor tables will be used for the estimation of one-hop delay. At the MAC layer, packets are serviced with a variable one-hop delay that mainly depends on the number of waiting packets in the MAC queue and the needed time in contending the shared channel. We use $E[q]_j$ to denote the mean number of packets waiting in the MAC queue of v_j. On the other hand, the shared channel at a host can generally be perceived as either idle or busy. The channel is idle if the host is not transmitting/receiving a data/control packet, or does not sense a busy carrier with a signal strength that exceeds a threshold. In IEEE 802.11, when a host tries to transmit a packet, it randomly selects a back-off time slots, denoted as cw, and counts down from cw to zero. The packet will be transmitted if it counts cw down to zero and the channel is observed as idle. Let bi_j and $E[cw]_j$ denote the ratio of busy time slots to idle time slots and the average back-off time slots during every period of time slots at v_j, respectively.

In order to maintain the two neighbor tables for all hosts, every host v_j has to broadcast its one-hop neighbor table periodically via control packets *d_hello* transmitted by using the base rate. Whenever v_i receives a *d_hello* packet from v_j, v_i can estimate $r\_sinr_{j,i}$ in its one-hop neighbor table by the received *d_hello* packet. The other four entries, bi_j, $E[cw]_j$, $E[q]_j$ and $t\_sinr_{i,j}$ in the one-hop neighbor table of v_i can be replaced by the carried bi_j, $E[cw]_j$, $E[q]_j$ and $r\_sinr_{i,j}$ in the one-hop neighbor table of v_j, respectively. Further, the entries bi_ks, $E[cw]_k$s, $E[q]_k$s and $t_sinr_{j,k}$s (where $k \neq i$) in the two-hop neighbor table of v_i and the associated links between the two neighbor tables of v_i can also be updated.

3.2 Estimation of one-hop and end-to-end delays

The method of estimating one-hop delay in our paper is similar to the one proposed in [14] by monitoring the busy/idle ratio of the channel. In [14], the authors proposed a delay-sensitive backoff range adaption mechanism and a distributed flow admission control mechanism for satisfying the QoS applications with delay requirements in IEEE 802.11 Wireless LANs. In the proposed mechanisms, the authors define the MAC access delay in slots, named as d_{mac}, which is the elapsed time interval from the time when the packet arrives in front of the queue to the time when it is received by the receiver. And, the authors quantitatively estimate d_{mac} as follows:

$$d_{mac} = \{E[cw] \cdot (1 + B(t)) + E[p]\} \cdot E[TransAtt] \tag{1}$$

$E[cw]$, $E[p]$, and $E[TransAtt]$ denote the mean back-off time slots, the mean time slots occupied by a single packet transmission, the mean number of transmission attempts, respectively, during every period of time slots, denoted as t. And, $B(t) = b\_slot/i\_slot$ denotes the busy/idle ratio of the channel during t, where b_slot and i_slot are the numbers of busy and idle time slots, respectively.

Further, $E[p]=[((l+p\_overhead)\times 8)/r]/(20\times 10^{-6})$, where l is the mean number of packet size in bytes, $p\_overhead$ is the PHY/MAC overhead, r is the transmission data rate, and 20×10^{-6} is a slot time in seconds. Since there may be some packets waiting in the queue, the waiting time of a packet in the queue before accessing the channel should be considered. Therefore, the one-hop delay d (in slots) can be estimated as equation (2), where $E[q]$ denotes the mean number of packets waiting in the MAC queue.

$$d = d_{mac} \times E[q] \tag{2}$$

On the other hand, the authors give the following math equation to compute the new busy/idle ratio of the channel, denoted as $\overline{B}(t)$, at new flow admission. Then, $\overline{B}(t)$ will be used for computing the new d if the new flow is admitted.

$$\overline{B}(t) = \frac{\overline{b\_slot}}{\overline{i\_slot}} = \frac{b\_slot + \beta}{i\_slot - \beta}, \text{ with } \beta = \overline{\lambda} \times t \times (20 \times 10^{-6}) \times m \tag{3}$$

Where β is the increment number of busy time slots induced by the new flow, m is the mean number of time slots occupied by a MAC packet and derived by $[(l\times 8)/r]/(20\times 10^{-6})$, $\overline{\lambda} = \lambda + \Delta\lambda$ is the overall packet arrival rate at queue, and λ ($\Delta\lambda$) is the packet arrival rate of the active flows (new flow).

However, the number of busy time slots, i.e., *b_slot*, is induced by the active flows during the current period time slots t, and the new coming packets belonging to the active flows at the queue will induce the same number of busy time slots during the next period time slots t. So, the packet arrival rate of the active flows dose not needed to be considered while computing the increment number of busy time slots. We replace the math equation in (3) as follows:

$$\overline{B}(t) = \frac{\overline{b\_slot}}{\overline{i\_slot}} = \frac{b\_slot + \beta}{i\_slot - \beta}, \text{ with } \beta = \Delta\lambda \times t \times (20 \times 10^{-6}) \times m \qquad (4)$$

Since the one-hop delay estimated above, i.e., d, is a mean value, the value used in the algorithm (presented below) should be greater than d in order to satisfy the delay requirement of the new requesting flow with a certain confidence level, i.e., a certain percentage of the packets whose transmitted one-hop delays should be smaller than d. That is we need to multiply d by a factor, denoted as $\alpha \geq 1$.

Throughput this paper, the assigned value of t is the same as that assigned in [14]. The value of t is set to 1024 idle time slots, i.e., 1024 idle time slots are needed to be sensed over the period of t. The values of $E[cw]$, $E[p]$, $E[TransAtt]$, $E[q]$ and $\overline{B}(t)$, are also averaged over the period of t. With the same reason stated in [14], these measured values are more accurate and stable as they are averaged over a long-enough period.

Suppose that the new flow from the source v_s to the destination v_d is initiated, its delay requirement is *d_req*, and $v_s \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow \ldots \rightarrow v_m \rightarrow v_d$ is determined as the delay-sensitive route for the flow. Let r_s, r_1, r_2, $r_3 \ldots r_m$ be the data rates used by v_s, v_1, v_2, $v_3 \ldots v_m$, respectively.

From (4), we can compute β_i, where $i \in \{s, 1,2,3\ldots m\}$.

Let bs_i and $\overline{bs}_i$ (is_i and $\overline{is}_i$) be the numbers of busy (idle) time slots of v_i before and after transmitting the flow. The bandwidths of v_s, v_1, and v_2 are consumed when v_s transmits packets to v_1, the bandwidths of v_s, v_1, v_2, and v_3 are consumed when v_1 transmits packets to v_2, and so on. Consequently, $\overline{bs}_s = bs_s + (\beta_s + \beta_1 + \beta_2)$ and $\overline{is}_s = is_s - (\beta_s + \beta_1 + \beta_2)$, $\overline{bs}_d = bs_d + (\beta_{m-1} + \beta_m)$ and $\overline{is}_d = is_d - (\beta_{m-1} + \beta_m)$, $\overline{bs}_1 = bs_1 + (\beta_s + \beta_1 + \beta_2 + \beta_3)$ and $\overline{is}_1 = is_1 - (\beta_s + \beta_1 + \beta_2 + \beta_3)$, $\overline{bs}_2 = bs_2 + (\beta_s + \beta_1 + \beta_2 + \beta_3 + \beta_4)$ and $\overline{is}_2 = is_2 - (\beta_s + \beta_1 + \beta_2 + \beta_3 + \beta_4)$, $\overline{bs}_{m-1} = bs_{m-1} + (\beta_{m-3} + \beta_{m-2} + \beta_{m-1} + \beta_m)$ and $\overline{is}_{m-1} = is_{m-1} - (\beta_{m-3} + \beta_{m-2} + \beta_{m-1} + \beta_m)$, $\overline{bs}_m = bs_m + (\beta_{m-2} + \beta_{m-1} + \beta_m)$ and $\overline{is}_m = is_m - (\beta_{m-2} + \beta_{m-1} + \beta_m)$, $\overline{bs}_i = bs_i + (\beta_{i-2} + \beta_{i-1} + \beta_i + \beta_{i+1} + \beta_{i+2})$ and $\overline{is}_i = is_i - (\beta_{i-2} + \beta_{i-1} + \beta_i + \beta_{i+1} + \beta_{i+2})$ where ($3 \le i \le m$-2). Thus, the new busy/idle ratio of v_j (where $v_j \in \{v_s, v_1, v_2, v_3, \ldots, v_m, v_d\}$) can be computed as:

$$\overline{B}_j(t) = \frac{\overline{bs}_j}{\overline{is}_j} \tag{5}$$

From equations (1), (2) and (5), d_j can be estimated. Further, the route should have $\sum d_j \le d\_req$, and $\overline{is}_j \ge 0$ for every v_j.

3.3 Determination of delay-sensitive route

This section proposes a distributed algorithm that can select data rates and determine a delay-sensitive route from a source to a destination for admitting a new flow with a delay requirement. When the destination requests to receive a flow with the delay requirement from the source, it broadcasts a control packet over the network. Once a host receives the control packet, it determines a data rate and checks if the data rate will cause a delay violation to it and its one-hop/two-hop neighbors. Recall that it can determine its one-hop/two-hop neighbors and estimate their one-hop delays by the methods of Section 3.1 and Section 3.2, respectively. If not, it is allowed to be a forwarder by piggybacking its identifier and determined data rate onto the control packet and forwards the packet to its neighbors. A forwarder and the source may receive multiple control packets from multiple hosts, i.e., there are likely multiple delay-sensitive routes from each of them to the destination. To increase the number of flows supported by a MANET, they aim to select a combination of higher data rates and a route with fewer neighbors for minimizing the bandwidth consumption to the network. Once the source selects a combination, which is a route consisted of some forwarders and data rates determined by these forwarders, the new request flow is admitted to be transmitted through the route, and the source and the forwarders transmit the flow by using their determined data rates.

We use v_s, v_d, and *d_req* to denote the source, the destination, and the delay requirement, respectively. Let $\hat{F}$ be the set of forwarders in the determined route and $\hat{R}$ be the set of data rates selected by the forwarders in $\hat{F}$, where $r_f \in \hat{R}$ is the data rate selected by $v_f \in \hat{F}$ ($1 \le f \le |\hat{F}|$) and $|\hat{F}| = |\hat{R}|$. Let $\hat{F}_{i,d}$ be the set of forwarders along the v_i - v_d route, $\hat{R}_{i,d}$ is the set of determined data rates used by the forwarders in $\hat{F}_{i,d}$, and $\hat{w}_{i,d}$ is the total medium time induced by the forwarders in $\hat{F}_{i,d}$, where $\hat{w}_{i,d} = \sum_{v_j \in \hat{F}_{i,d}} \hat{w}_j$. Here, $\hat{w}_j$ is the medium time induced by forwarder v_j. With v_j's data rate r_j recorded in $\hat{R}_{i,d}$, we define $\hat{w}_j = \frac{l}{r_j} \times | I_{j,r_j} \cup II_{j,r_j} \cup \{v_j\} |$, where l is the packet size, and I_{j,r_j} (II_{j,r_j}) is the set of r_j-one-hop (r_j-two-hop) neighboring hosts of v_j.

When v_d needs to receive a flow with *d_req* from v_s, it broadcasts a control packet *d__query*, which carries six variables *d_req*, $\hat{w} = 0$, $\hat{F} = \{\}$, $\hat{R} = \{\}$, $\hat{\beta} = \{\}$, and $\hat{M} = \{\}$ (explained later), to its neighbors. When a host v_α, (a neighbor of v_d) receives the packet, v_α

executes the following algorithm to determine a data rate r_α, in which the requested flow transmitted using r_α will not cause a delay violation to it and its r_α-one-hop (r_α-two-hop) neighbors in order to avoid HRP.

for each $r^t \in R$ and $v_d \in I_{\alpha,r^t}$ **do**

$M_\alpha \leftarrow I_{\alpha,r^t} \cup II_{\alpha,r^t} \cup \{v_\alpha\}$;

Compute β_α and d_α by the method of Section 3.2 in the case of v_α using r^t;

if $d_\alpha > d\_req$

then $\hat{w}^t \leftarrow \infty$ and **break**

else **for** each $v_i \in M_\alpha$ **do**

if $\beta_\alpha > is_i$ **then** $\hat{w}^t \leftarrow \infty$ and **break**

if $\hat{w}^t \neq \infty$ **then** $\hat{w}^t \leftarrow \frac{l}{r^t} | M_\alpha |$

Determine r^x so that $\hat{w}^x = \min\{ \hat{w}^t \mid r^t \in R \}$;

$\hat{w}_{\alpha,d} = \hat{w}^x$ and $r_\alpha = r^x$.

In the algorithm, $\hat{w}^t$ is the weight of the v_α - v_d route from v_α to v_d if r^t is used, where $\hat{w}^t = \frac{l}{r^t} | I_{\alpha,r^t} \cup II_{\alpha,r^t} \cup \{v_\alpha\} |$ is the sum of time consumption to v_α and the r^t-one-hop (r^t-two-hop) neighbors of v_α. If a r^x with minimum $\hat{w}^x$ is found, v_α is the only forwarder in the v_α - v_d route, and $\hat{w}_{\alpha,d} = \hat{w}^x$, $r_\alpha = r^x$. Then, v_α replaces $\hat{w}$ with $\hat{w}_{\alpha,d}$, $\hat{F}$ with $\{v_\alpha\}$, $\hat{R}$ with $\{r_\alpha\}$, $\hat{\beta}$ with $\{\beta_\alpha\}$, and $\hat{M}$ with $\{M_\alpha\}$. Further, v_α forwards *d__query* to its neighbors. Otherwise, it discards the received packet.

Once a host v_k receives the packet forwarded from v_α, v_k executes another algorithm to determine the data rate r_k.

for each $r^t \in R$ and $v_\alpha \in I_{k,r^t}$ **do**

$M_k \leftarrow I_{k,r^t} \cup II_{k,r^t} \cup \{v_k\}$;

Compute β_k, d_α and d_k in the case of v_k using r^t;

if $d_\alpha + d_k > d\_req$

then $\hat{w}^t \leftarrow \infty$ and **break**

else **for** each $v_i \in M_k$ **do**

increment= β_k;

for each $v_j \in \hat{F}$ **do**

if $v_j \in M_k$ **then** *increment*= *increment*+ β_j

if *increment* > is_i **then** $\hat{w}_t \leftarrow \infty$ and **break**

if $\hat{w}_t \neq \infty$ **then** $\hat{w}^t \leftarrow \hat{w} + \frac{l}{r^t} |M_k|$

determine r^x so that $\hat{w}^x = \min\{ \hat{w}^t \mid r^t \in R \}$;

$\hat{w}_{k,d} = \hat{w}^x$ and $r_k = r^x$.

If an r_k is found, v_k forwards the packet that carries with $\hat{w}_{k,d}$, $\hat{F} = \{v_\alpha, v_k\}$, $\hat{R} = \{r_\alpha, r_k\}$, $\hat{\beta} = \{\beta_\alpha, \beta_k\}$, and $\hat{M} = \{M_\alpha, M_k\}$. If v_k receives another packet forwarded from v_c, where v_c is a neighbor of v_k and $v_c \neq v_\alpha$, v_k will execute the above algorithm again. If the new $\hat{w}_{k,d}$ is smaller than the previous one, then it forwards the control packet that carries with the new four variables. Otherwise, it discards the received packet.

A host who receives the forwarding packet executes the similar checking and forwarding procedure. An algorithm to conclude the above two algorithms when a host v_o receives a control packet *d_query* from v_z, which carries *d_req*, $\hat{w}_{z,d}$, $\hat{F} = \{v_1, v_2, \ldots, v_z\}$, $\hat{R} = \{r_1, r_2, \ldots, r_z\}$, $\hat{\beta} = \{\beta_1, \beta_2, \ldots, \beta_z\}$, and $\hat{M} = \{M_1, M_2, \ldots, M_z\}$, is elaborated below.

for each $r^t \in R$ and $v_z \in I_{o,r^t}$ **do**

$M_o \leftarrow I_{o,r^t} \cup II_{o,r^t} \cup \{v_o\}$;

Compute β_o, d_i for every $v_i \in \hat{F}$, and d_o in the case of v_o using r^t;

if $\sum_{v_i \in \hat{F}} d_i + d_o > d\_req$

then $\hat{w}^t \leftarrow \infty$ and **break**

else **for** each $v_i \in M_o$ **do**

increment= β_o;

for each $v_j \in \hat{F}$ **do**

if $v_j \in M_o$ **then** *increment*= *increment*+ β_j

if *increment* > is_i **then** $\hat{w}^t \leftarrow \infty$ and **break**

if $\hat{w}^t \neq \infty$ **then** $\hat{w}^t \leftarrow \hat{w}^t + \frac{l}{r^t} | M_o |$

determine r^x so that $\hat{w}^x = \min\{ \hat{w}^t \mid r^t \in R \}$;

$\hat{w}_{o,d} = \hat{w}^x$ and $r_o = r^x$.

If v_s receives the control packet, it executes the above algorithm to find a rate. If such a rate is found, then the delay-sensitive route (i.e., $\hat{F}$) from v_s to v_d is determined, and the data rates (i.e., $\hat{R}$) used by the forwarders in $\hat{F}$ are selected. Further, v_s may receive multiple *d_ query* packets from multiple hosts. It selects the combination ($\hat{F}$ and $\hat{R}$) with the minimum weight in order to minimize the total medium time to the network.

For example, refer to Figure 3, where the proposed algorithm is executed to establish a delay-sensitive route (with *d_req*) that connects the server v_s with the destination v_d. Refer to Figure 3(a), v_d broadcasts *d__query*, which carries *d_req*, $\hat{w} = 0$, $\hat{F} = \{\}$, $\hat{R} = \{\}$, $\hat{\beta} = \{\}$, and $\hat{M} = \{\}$, to its neighbors v_b and v_c. Refer to Figure 3(b) (Figure 3(c)), v_b (v_c) forwards *d__query* by carrying *d_req*, $\hat{w}_{b,d}$ ($\hat{w}_{c,d}$), $\hat{F} = \{v_b\}$ ($= \{v_c\}$), $\hat{R} = \{r_b\}$ ($= \{r_c\}$), $\hat{\beta} = \{\beta_b\}$ ($= \{\beta_c\}$), and $\hat{M} = \{M_b\}$ ($= \{M_c\}$), to v_a, v_c, v_e and v_f (to v_e and v_g). In this example, v_c discards *d__query* received from v_b since the v_c - v_d route has smaller $\hat{w}_{c,d}$ than the v_c – v_b -

v_d route. Refer to Figure 3(d), v_e forwards *d__query* to v_h and v_s when it receives *d__query* from v_b. However, v_e forwards *d__query* to v_h and v_s again when it receives *d__query* from v_c since the v_e -v_c - v_d route has smaller $\hat{w}_{e,d}$ than the v_e – v_b - v_d route. Similarly, v_s will receive *d__query* from v_e and v_g, and it selects the v_s –v_g -v_c - v_d route rather than the v_s –v_e -v_c - v_d route since the former route has smaller $\hat{w}_{s,d}$.

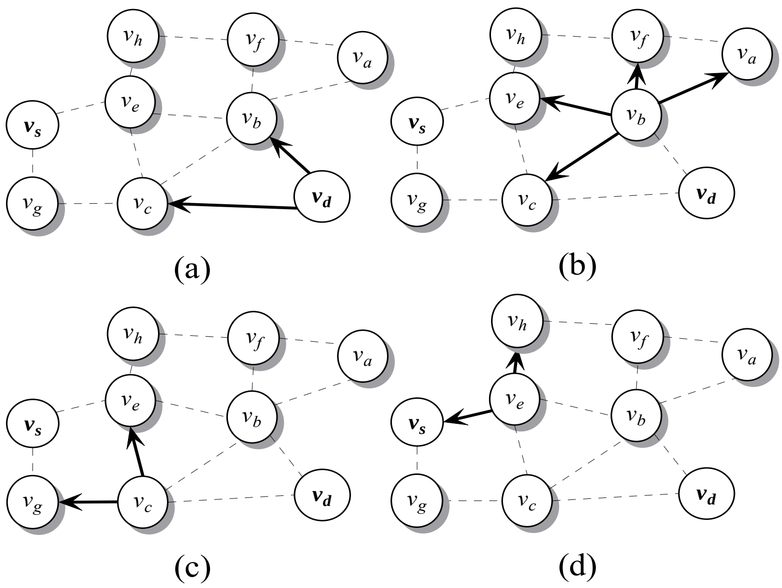


Figure 3. A delay-sensitive route. (a) v_d broadcasts *d__query* to v_b and v_c. (b) v_b broadcasts *d__query* to v_a, v_c, v_e and v_f. (c) v_b broadcasts *d__query* to v_e and v_g. (d) v_e broadcasts *d__query* to v_h and v_s.

4 Simulation and performance analysis

Simulations are implemented using the Network Simulator 2 package [16]. IEEE 802.11 Distribution Coordination Function (DCF) is used as the MAC layer protocol. Packets are sent using the un-slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In the simulations, Constant Bit Rate (CBR) data traffic flows are injected into the network from the sources and the size of the data payload is 512 bytes. Each host has a First-In-First-Out transmission queue of no more than 64 packets at MAC layer. Forty runs with different seed numbers are conducted for each scenario and collected data for these runs are averaged.

The simulations are performed with three aspects. First, in order to validate the accuracy of our one-hop delay estimating method vs. the estimating method proposed in [14]. Second, we use another example to show the effectiveness of our distributed protocol vs. AQOR (which is a representative QoS routing protocol in MANETs) in avoiding HRP. Third, performance comparisons are made between the distinct strategies in our proposed protocol and another routing protocol [5] for constructing delay-sensitive routes in multi-rate MANETs. In our proposed protocol, we aim to minimize the total medium time to the network, whereas in [5], the authors aim to determine a route by selecting the links with smaller interference time. In [5], the experiments have shown that its proposed strategy has significantly higher performance compared to other routing strategies.

4.1 One-hop delay estimation

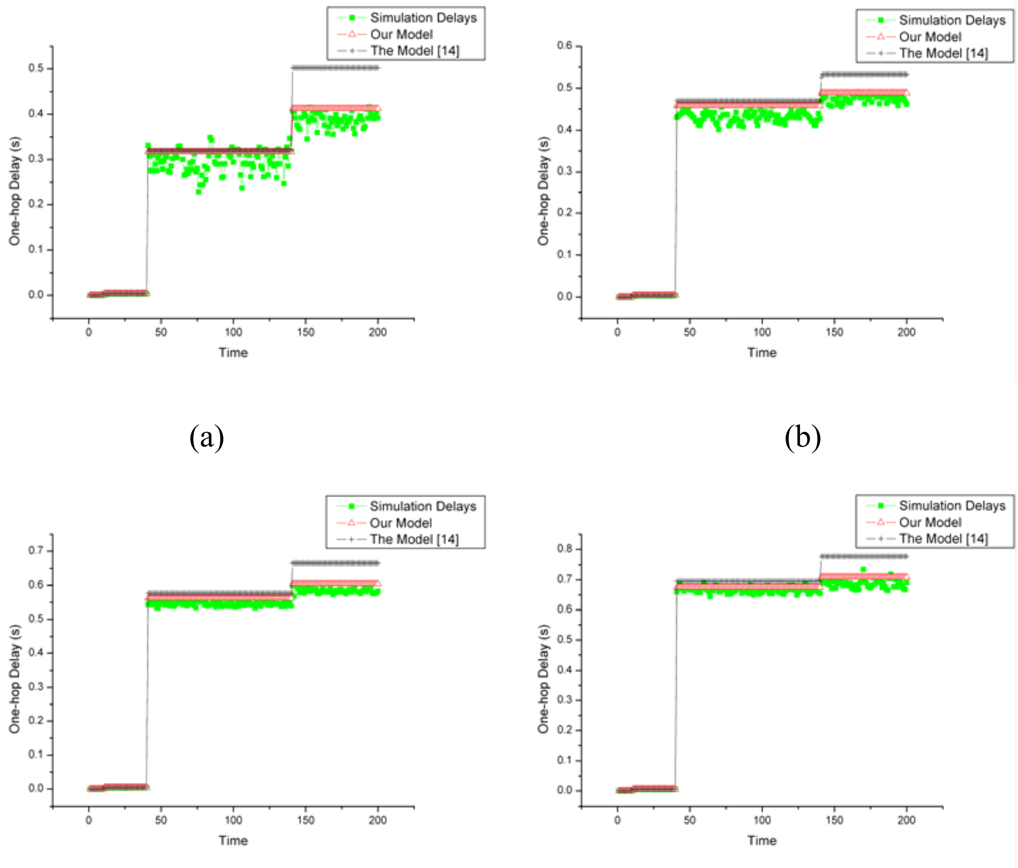
In this section, we compare our one-hop delay estimation model with the model proposed in [14]. The simulation scenarios are constructed by using the same as those constructed in Figures 10-13 of [14], where four kinds of CBR unicast flows with varied bit rates are ejected into the MANET. Traffic characteristics of these kinds are specified in table I.

Table I. Traffic characteristics of four kinds of CBR unicast flows

Traffic features	Packet size (Bytes)	Generation Interval (second)	Bit rate (bps)
Max_delay_0.5 (HP)	160	0.02	64000
Max_delay_0.5 (HP)	160	0.01	128000
Max_delay_0.8 (MP)	500	0.02	200000
Max_delay_0.8 (MP)	500	0.01	400000

In the simulations, sixteen hosts are created as a MANET, and the transmission of any host can be heard by the other hosts. Each CBR unicast flow is sent from a source to a destination, which are selected randomly among the hosts. Each run of the simulation consists of 200 seconds. From time t=0 second to t=10 seconds, the channel is empty, whereas from t=10 seconds, new flows of each kind are started at 3-seconds intervals and begin competing for the channel. By t=37 seconds, there are ten active flows in the network: two 64-Kbps flows, three 128-Kbps flows, three 200-Kbps flows and two 400-Kbps flows. From t=37 seconds to t=140 seconds, the network remains in the same state. At t=140 seconds, another 64-Kbps flow is admitted to the network.

Recall that we replace the equation (3) (which is used in [14]) with the equation (4) for estimating one-hop delays in order to correct the overestimation in [14]. In figure 4, the simulation results validate that the estimating one-hop delays obtained by our model close to the instantaneous one-hop delays obtained in the simulations within the whole simulation period.



(a) (b)

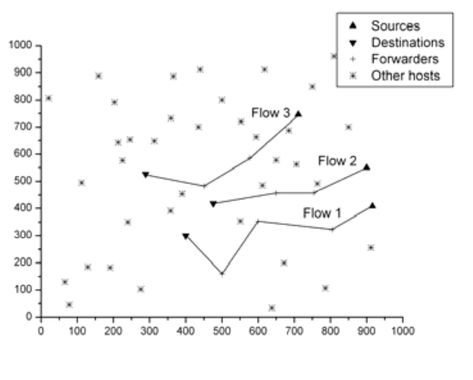
(c) (d)

Figure 4. One-hop delays: (a) 64-Kbps (b) 128-Kbps (c) 200-Kbps (d) 400-Kbps.

4.2 Avoidance of HRP

We present a simple simulation in a 1,000 m × 1,000 m area with 50 randomly positioned hosts. We measure the performance in terms of end-to-end delay and success ratio. The success ratio is the ratio of the number of data packets with satisfied end-to-end delays successfully received by the destinations to the number of data packets delivered by the sources. We first show that there are three 64 Kbps flows, i.e., Flow 1, Flow 2, and Flow 3, and the requirement of end-to-end delay of each flow is set to 50 ms. Figure 5 shows the three routes constructed by AQOR for the three flows. Flow 1 starts transmission first, Flow 2 starts transmission after 50 seconds has elapsed, and Flow 3 starts transmission after 100 seconds has elapsed. Their end-to-end delays and receiving ratios, which are both collected every second, are exhibited in figure 6 and figure 7, respectively. During the first 100 seconds, the delays of Flow 1 and 2 are lower than 0.03 seconds. However, after 100 seconds has elapsed, the delays of the three flows increase drastically as a consequence of HRP triggered by Flow 3. Like the previous situation, the success ratios of the three flows also drop drastically when Flow 3 starts.

Figures 8-10 demonstrate the effectiveness of our proposed protocol in avoiding HRP that is triggered in Figures 5-7. The delay-sensitive routes for Flow 1, 2, and 3 are shown in Figure 8. In Figure 9, the end-to-end delays of the tree flows are delay-sensitive, i.e., they are smaller than 50 ms, because the delay induced by the two-hop hosts is considered in our proposed protocol. Figure 9 also shows that the success ratios of the tree flows are higher than 95%.



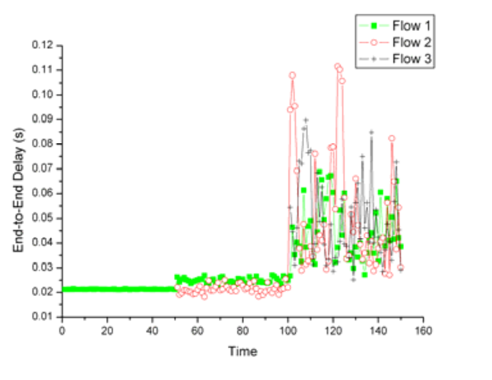
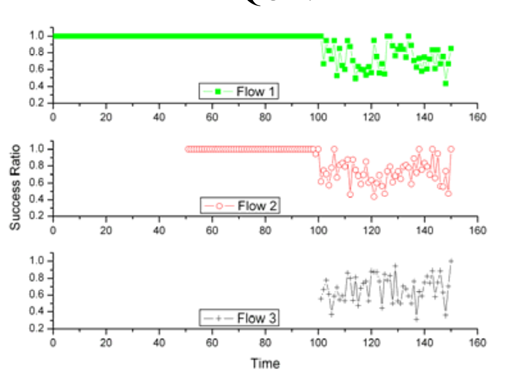


Figure 5. Three routes constructed by AQOR.

Figure 6. End-to-end delays of three flows in Figure 5.



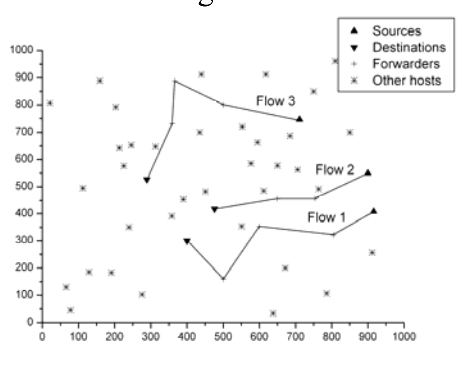
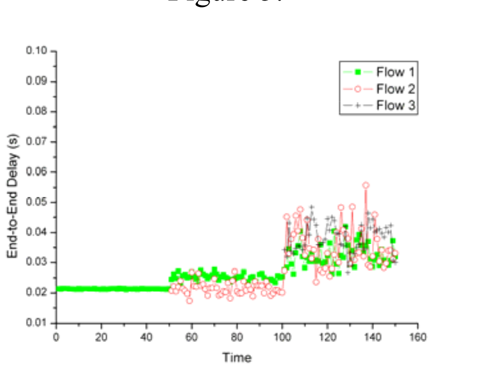


Figure 7. Success ratios of three flows in Figure 5.

Figure 8. Three routes constructed by our proposed protocol.



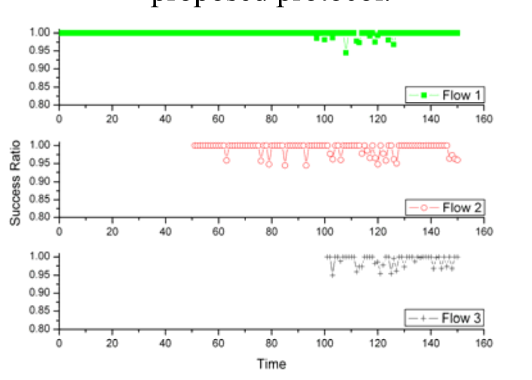


Figure 9. End-to-end delays of three flows in Figure 8.

Figure 10. Success ratios of three flows in Figure 8.

4.3 Performance comparisons of strategies in route construction

In the simulation environment, 50 hosts are randomly distributed over a 1000 m × 1000 m area. Two kinds, i.e., 128 Kbps and 200 Kbps, of flows with 0.2 second end-to-end delay requirement are ejected from the sources to the destinations. The IEEE 802.11b is used as our

MAC/PHY protocol, i.e., there are four available data rates 1, 2, 5.5, and 11 Mbps. We measure the performance in terms of end-to-end delay and admission ratio for the distinct strategies of route construction in our proposed protocol and the routing protocol [5]. The admission ratio is the ratio of the number of admitted source-destination pairs to the number of requested source-destination pairs.

In figures 11 and 12, the end-to-end delays of both protocols are small when the number of flows is small, and the routing protocol [5] obtains smaller end-to-end delays than our proposed protocol as a consequence that the routes constructed by it have the links with smaller interference time. However, when the number of flows with 128 Kbps (200 Kbps) increases from 12 (8) to 30 (20), the average end-to-end delays of the protocol [5] increases drastically. It is likely that the routing protocol [5] selects the routes which may blocks many hosts. On the other hand, the values of our proposed protocol increase slightly as a consequence that our proposed protocol adopts the strategy of selecting the routes with the minimum total medium time to the network. Figures 13 and 14 validate that our proposed protocol can accommodate more flows.

Figure 15 shows that our proposed protocol generates more control bytes per second than the routing protocol [5] since the control packets *d_hello* exchanged periodically among the hosts in our proposed protocol. In the simulations of Figure 15, we set the frequency to one per second. However, by the aid of the more control bytes, our proposed protocol can accommodate more flows than the routing protocol [5].

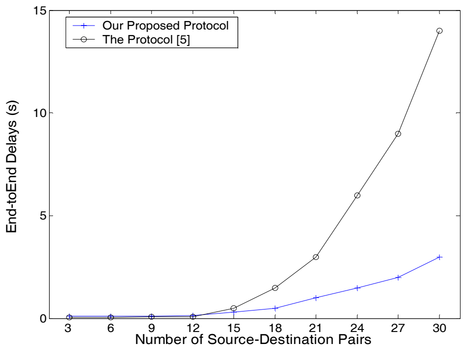


Figure. 11. End-to-end delays of 128 Kbps flows.

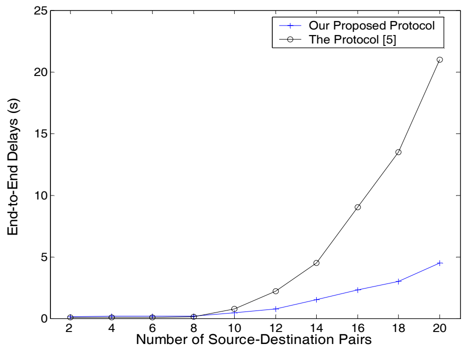


Figure. 12. End-to-end delays of 200 Kbps flows.

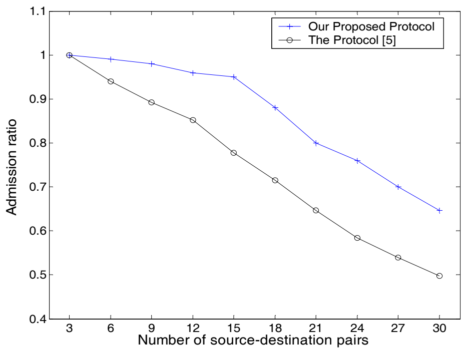


Figure 13. Admission ratio of 128 Kbps flows.

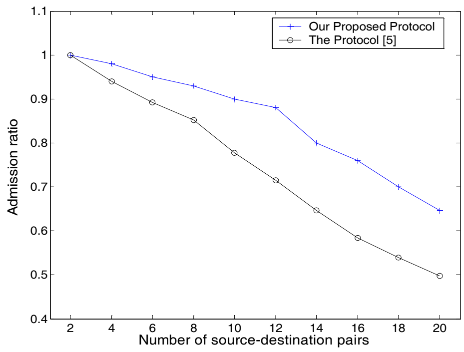


Figure 14. admission ratio of 200 Kbps flows.

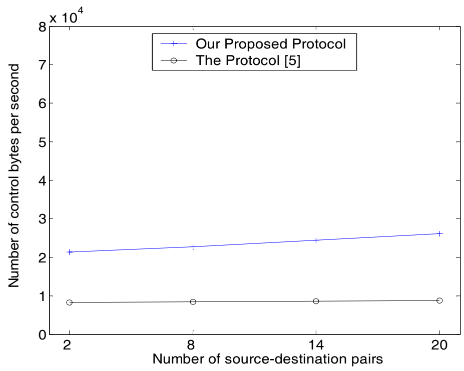


Figure 15. Number of control bytes per second

5 Conclusion

In order to exploit wireless resource efficiently and provide QoS for MANET services, we proposed a delay-sensitive routing protocol in multi-rate MANETs. Since one-hop delay should be known before constructing a delay-sensitive route, we first propose a method of estimating one-hop delay based on varied data rates. Simulation results validate that the proposed method can obtain more precise one-hop delay than that proposed in a very recent work [14]. Then by the aid of one-hop delay estimation, end-to-end delay can be accurately accumulated while constructing the route.

HRP, which is one delay-violation problem that may occur to previous QoS routing protocols, is introduced in this paper. In this paper, the proposed routing protocol is proposed to avoid HRP by further considering the resource consumption of one-hop/two-hop neighbors when a route is being determined. Further, it adopts the strategy of selecting the combination of higher data rates and a route with fewer neighbors in order to minimize the total medium time. The simulation results show that the proposed protocol can improve network throughput by admitting more flows.

References

[1] Q. Xue and A. Ganz, "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 41, pp. 120-124, 2003.

[2] A. Abdrabou and W. Zhuang, "Statistical QoS routing for IEEE 802.11 multihop ad hoc networks," *IEEE Transaction on Wireless Communications*, vol. 8, pp. 1542-1552, 2009.

[3] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1454-1465, 1999.

[4] J.L. Hsu and I. Rubin, "Cross-Layer Multi-Rate Routing Strategies in Wireless Multi-Hop Random Access Networks," *IEEE Global Telecommunications Conference* (*GLOBECOM '07*), pp. 609-613, 2007.

[5] M. Genetzakis and V.A. Siris, "A contention-aware routing metric for mult-rate multi-radio mesh networks," *IEEE Communications Society Conference on Sensor, Mesh and Ah Hoc Communications and Networks* (*SECON '08*), pp. 242-250, 2008.

[6] C.C. Hu, E.H.K. Wu, and G.H. Chen, "Bandwidth-satisfied multicast tree in MANETs,"

IEEE Transations on Mobile Computing, vol. 7, no. 6, 2008, pp. 712-723.

[7] A. Abdrabou and W. Zhuang, "Stochastic delay guarantees and statistical call admission control for IEEE 802.11 single-hop ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3972-3981, 2008.

[8] T. Sakurai, and H.L. Vu, "MAC access delay of IEEE 802.11 DCF," *IEEE Transactions on Wireless Communications*, vol. 6 no. 5, 2007.

[9] Q. Zhao, D.H.K. Tsang, and T. Sakurai, "A simple and approximate model for nonsaturated IEEE 802.11 DCF," *IEEE Transactions on Mobile Computing*, to appear.

[10] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," International Conference on Mobile Computing and Networking, pp. 236-251, 2001.

[11] H. Lim, D. Jung, J. Hwang, K.-J. Park, and J. C. Hou, "Contention control for high end-to-end throughput performance of wireless mesh networks," *Third Annual International Wireless Internet Conference*, 2007, pp. 22-24.

[12] A. Muqattash and M. Krunz, "POWMAC: a single-channel power-control protocol for throughput enhancement in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 1067-1084, 2005.

[13] C.R. Cerveira and L.H.M.K. Costa, "A time-based admission control mechanism for IEEE 802.11 ad hoc networks," *Mobile and Wireless Communication Networks*, vol. 211, pp. 217-228, 2006.

[14] A. Nafaa and A. Ksentini, "On Sustained QoS Guarantees in Operated IEEE 802.11 Wireless LANs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, pp. 1020-1033, 2008.

[15] Y. Xi, B.S. Kim, J.B. Wei, and Q.Y. Huang, "Adaptive multirate auto rate fallback protocol for IEEE 802.11 WLANs," *Military Communications Conference*, pp. 1-7, 2006

[16] Network Simulator (Version 2), http://www.isi.edu/nsnam/ns/, 2008.

[17] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "IEEE 802.11b ad hoc networks: Performance measurements," *Cluster Computing*, vol. 8, pp. 135-145, 2005.