Journal of Computers Vol. 31 No. 3, 2020, pp. 168-182
doi:10.3966/199115992020063103013

Delay-Guaranteed Multicast Trees in Multi-Rate MANETs

Chia-Cheng Hu[1,2]*, Zhong-bao Liu[2], Su-Zhen Ge[2], Yu-Mei Lin[2]

[1] College of Information Engineering, Yango University, Fuzhou 350015, Fujian, China

[2] School of Software, Quanzhou Institute of Information Engineering, Quanzhou 362000, Fujian, China
cchu.chiachenghu@gmail.com, zhongbao@hotmail.com, 592260817@qq.com, 45664819@qq.com

Received 17 November 2018; Revised 25 May 2019; Accepted 31 May 2019

Abstract. Nowadays, some MAC standards such as 802.11a, 802.11b, and 802.11g can operate with multiple data rates for QoS-constrained multimedia applications to utilize the limited resources of mobile ad hoc networks (MANETs) more efficiently. In this paper, a delay-guaranteed multicast protocol is proposed to determine a multicast tree for real-time applications in multi-rate MANETs. In the proposed protocol, a method for estimating one-hop delay is proposed first. In order to maximize the network capacity, the tree is determined by adopting the strategy of minimizing the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts. Simulation results show that the method is more accurate in estimating one-hop delay and the proposed protocol obtains higher network capacity than previous works.

Keywords: delay guarantee, IEEE 802.11, multicast, multi-rate, MANET, QoS

1 Introduction

In mobile ad hoc networks (MANETs), there are more and more applications that rely on real-time multicast services such as VoIP, video conferencing, emergent warning, and battlefield operation. The multicast protocols in MANETs should provide delay guarantees for the services. To build a delay-guaranteed multicast tree, one-hop delay and end-to-end delay must be known in advance. The one-hop delay is the time required to transmit a packet between two neighboring hosts, and the end-to-end delay is the time required to transmit a packet from the source to the destination. However, to estimate the two delays is difficult because of the common radio channel shared among the neighboring hosts in MANETs.

Previously, there were some routing/multicasting protocols proposed for single-rate MANETs [1-7] or multi-rate MANETs [8-13]. However, they could not provide either of bandwidth guarantee and delay guarantee since that they the one-hop and two-hop neighboring information is not considered. Thus, the hidden route problem (HRP) or the hidden multicast route problem (HMRP) will be induced [14]. 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. Both HRP and HMRP arise as a consequence that the transmitters fail to estimate the resource consumption of their two-hop neighbors. When the two problems are induced, QoS requirements (e.g., bandwidth guarantees or delay guarantees) of ongoing flows cannot be satisfied. In [14], the simulation results showed that the two problems happen frequently and the network performance degrades considerably when the network traffic was saturated.

By means of measuring the busy/idle ratio of the shared radio channel, a method for estimating the one-hop delay and end-to-end delay in multi-rate MANETs is proposed in [15]. By integrating the delay estimation method into ODMRP [16], a multicast protocol is proposed. However, the protocol also suffered from the two problems due to not considering two-hop neighboring information. In this paper, an enhanced multicast protocol for multi-rate MANETs is proposed in providing delay guarantees by the two problems for the requesting flow and all ongoing flows. Further, it aims to minimize the sum of the

* Corresponding Author

total transmission time of the forwarders and the total blocking time of the blocked hosts by adjusting data rates. The multicasting routes thus found can not only provide delay guarantees by avoiding HRP and HMRP but also enhance the network capacity of admitting more applications.

In the next section, related works are reviewed. In Section 3, the method in estimating one-hop delay and the protocol in determining delay-guaranteed multicast are introduced. In Section 4, the performance of the method and the protocol is validated by extensive simulation. Finally, this paper concludes with some remarks in Section 5.

2 Related Works

Previous routing/multicasting protocols [1-7] that intended to provide bandwidth guarantees or delay guarantees in single-rate MANETs are first reviewed. In [1], a bandwidth- guaranteed routing protocol, named CEDAR, was proposed to select routes with stable links and high available bandwidths. In [2], a routing protocol, named AQOR, was proposed by selecting the bandwidth-guaranteed route with the shortest end-to-end delay. Since CEDAR did not estimate the end-to-end delay and AQOR did not take the newly admitted flows into consideration, they failed to provide delay guarantees. Besides, they might suffer from HRP. In [3], a bandwidth-guaranteed routing protocol was proposed. Since the bandwidth consumption of one-hop neighbors and two-hop neighbors was calculated, HRP could be avoided. In [4], a bandwidth-guaranteed multicast protocol (named MCEDAR), which was an extension of CEDAR, was proposed. In [5], a bandwidth-guaranteed multicast protocol, named M-CAMP, was proposed by adopting a measurement-based approach to estimate the available bandwidth of the multicast tree. In [6], a bandwidth-guaranteed multicast protocol, named QoS-ODMRP, was proposed. The three multicast protocols above might suffer from HMRP, as a consequence that they constructed the routes from the source to multiple destinations concurrently. In [7], the authors investigate multicast in MANET based on a general Markovian mobility model in order to guarantee delay performance.

In [8], a mathematical model was proposed for data rate selection and route determination. In [9], a routing protocol was proposed in multi-rate multi-radio mesh networks. In [10], a multicast protocol was proposed in multi-rate MANETs, which dynamically adjusted the data rate based on the channel quality. The multi-rate protocols in [8-10] failed to provide bandwidth guarantees and delay guarantees. In [11], the authors introduce a mobility aware recovery technique for reliable multicast routing in MANETs to enhance the multicast efficiency and lifetime of the network. In [12], the authors study broadcast capacity and minimum delay scaling laws for highly mobile wireless networks, in which each node has to disseminate or broadcast packets to all other nodes in the network. In [13], the authors investigate the multicast capacity for vehicular ad hoc networks (VANETs) with directional antennas under the end-to-end delay constraint.

In [17-18], the behavior of the IEEE 802.11 protocol was explicitly analyzed according to different traffic loads, and the Markov-modulated Poisson process was used to estimate the average delays. However, their analysis did not consider the channel condition, and the parameters, e.g., average on-time and average off-time, and the traffic sources must be aware in advance. In [19-20], an approximation model was proposed to estimate one-hop delay by computing the collision probability while transmitting packets. However, it is hard to estimate the traffic attempt rate for a wireless channel, especially in multi-rate MANETs. In [21], another method for estimating one-hop delay was proposed, which measured the ratio of busy/idle periods of the shared channel. Since the estimation methods in [19-21] took the channel condition into consideration, they estimated one-hop delay more precisely than those in [17-18]. In [15], the ratio of busy/idle periods of the shared radio channel was used to estimate both of the one-hop delay and end-to-end delay.

3 Delay-Guaranteed Multicast Protocol

In this section, a delay-guaranteed multicast protocol in multi-rate MANETs is proposed. The IEEE 802.11 DCF with multiple rates is used as the underlying MAC protocol. A single physical channel is available for packet transmission, and hosts are able to monitor the status of the channel, which is perceived as either idle or busy. In a host, the channel is considered as idle if the host does not sense a busy carrier with a signal strength exceeding the carrier sense threshold. In the proposed protocol, data packets can be transmitted with different rates, but control packets are transmitted with the base (i.e., the lowest) rate.

Suppose that there are n data rates $r_1, r_2, \ldots, r_n$ available, where $r_1 < r_2 < \ldots < r_n$. If a host h_i uses a data rate r_p and it is within the transmission range of host h_j, h_i is defined as a r_p-one-hop neighbor of h_j, where $1 \leq p \leq n$. On the other hand, host h_k is a r_p-two-hop neighbor of h_i if it is a r_p-one-hop neighbor of h_j and is not a r_p-one-hop neighbor of h_i. The transmission range is a distance from a transmitter to the receiver that can receive and decode the packets correctly. Actually, it varies with the data rate used by the transmitter. A higher data rate will result in a smaller transmission range.

When a transmitter is transmitting some packets, all hosts within its carrier sense range are blocked. The carrier sense range is another distance from the transmitter to the hosts that can sense the signal but cannot decode the packets correctly. In [22-24], 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 proposed multicast protocol aims to construct a delay-guaranteed multicast tree for minimizing the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts. The delay violation of the ongoing flows due to newly admitted flows will also be avoided. For this purpose, a method to estimate the one-hop delay for each forwarder along the tree is needed.

3.1 One-hop Delay Estimation

In order to identify the r_p-one-hop and r_p-two-hop neighbors of a host h_i in multi-rate MANETs, each host needs to construct two tables, named one-hop neighbor table and two-hop neighbor table. We use $T_{i,1}$ and $T_{i,2}$ to denote the one-hop neighbor table and two-hop neighbor table of h_i, respectively. All r_1-one-hop neighbors h_j of h_i are stored in $T_{i,1}$, and all r_1-one-hop neighbors h_k of h_j are stored in $T_{i,2}$, where r_1 is the base rate. Associated with h_j, the following parameters are also stored in $T_{i,1}$: the busy/idle ratio (b_j), the expected number of backoff time slots (Eb_j), the expected number of packets in MAC queue (Ep_j), the h_i-to-h_j SINR value ($SINR_{i,j}$), and the h_j-to-h_i SINR value ($SINR_{j,i}$). The h_i-to-h_j (h_j-to-h_i) SINR value is the SINR value of the link from h_i to h_j (from h_j to h_i). Associated with h_k, four parameters: b_k, Eb_k, Ep_k, and $SINR_{j,k}$, are stored in $T_{i,2}$.

Since the receiver perceives the channel quality in a more timely manner than the transmitter, the receiver-based auto rate (RBAR) algorithm [25] (which is a receiver-based approach) can yield significant throughput gains, as compared with the auto rate fallback (ARF) algorithm [26] (which is a transmitter-based approach). Similarly, $SINR_{i,j}$, $SINR_{j,i}$ and $SINR_{j,k}$ are estimated by the receivers h_j, h_i and h_k, respectively. Once h_j (h_i and h_k, respectively) receives a packet from h_i (h_j and h_j, respectively), $SINR_{i,j}$ ($SINR_{j,i}$ and $SINR_{j,k}$, respectively) is estimated based on the received signal quality.

Let $thrd_p$ be the SINR threshold of the acceptable bit error rate (BER) while using data rate r_p. That is, if h_i uses data rate r_p and $SINR_{i,j} \geq thrd_p$, then h_j can receive packets successfully from h_i, or h_j is an r_p-one-hop neighbor of h_i. Similarly, if h_j uses data rate r_p and $SINR_{j,k} \geq thrd_p$, then h_k is an r_p-one-hop neighbor of h_j. Further, h_k is an r_p-two-hop neighbor of h_i, if it is not an r_p-one-hop neighbor of h_i. In this way, all r_p-one-hop and r_p-two-hop neighbors of h_i can be determined from $T_{i,1}$ and $T_{i,2}$. On the other hand, when packets are transmitted, they may have different one-hop delays, which mainly depend on the number of waiting packets in the MAC queue and the contention time for the shared radio channel. The values of Ep_j and Ep_k are needed for estimating the former, and the values of b_j, b_k, Eb_j, and Eb_k are needed for estimating the latter.

In order to maintain $T_{i,1}$ and $T_{i,2}$, h_i has to get up-to-date one-hop neighbor tables, together with b_j, Eb_j and Ep_j, of all its r_1-one-hop neighbors h_j. This can be done by periodically exchanging *hello* packets. The value of $SINR_{j,i}$ in $T_{i,1}$ can be updated whenever h_i receives a packet from h_j. The proposed estimation method of one-hop delay is similar to that proposed in [21]. In [21], two mechanisms for backoff range adaptation and flow admission control were proposed in order to satisfy the QoS applications with delay requirements in IEEE 802.11 wireless LANs. The *MAC access delay* of a packet is the elapsed time (in number of time slots) from the time when the packet arrives at the head of the MAC queue to the time when it is received by the receiver.

For a given time period, the following parameters are defined with respect to a transmitter h_i, where f is a requesting flow within the carrier sense range of h_i.

d_i: one-hop delay (in number of time slots) of h_i;

$\tilde{d}_i$: MAC access delay of a packet sent by h_i;

$Eslot_i$: expected number of time slots needed for h_i to transmit a packet;

$Etran_i$: expected number of transmission attempts of h_i;

b_i: busy/idle ratio of the channel sensed by h_i;

$\tilde{b}_l$: busy/idle ratio of the channel sensed by h_i after f is admitted;

$bslot_i$ ($islot_i$): number of busy (idle) time slots sensed by h_i;

λ_i: total packet arrival rate of the ongoing flows that are within the carrier sense range of h_i;

$\Delta\lambda$: packet arrival rate of f.

Suppose that h_i uses data rate r_p (in Mbps), and let El be the expected packet length (in bytes). Then, $b_i = \frac{bslot_i}{islot_i}$ and $Eslot_i = \frac{El \times 8}{r_p \times (20 \times 10^{-6})}$, where 20×10^{-6} is the time length (in seconds) of a time slot. The values of $\tilde{d}_l$, d_i, and $\tilde{b}_l$ were estimated in [17] as follows:

$$\tilde{d}_l = (Eb_i \times (1 + b_i) + Eslot_i) \times Etran_i; \quad \textbf{(1)}$$

$$d_i = \tilde{d}_l \times Ep_i; \quad \textbf{(2)}$$

$$\tilde{b}_l = \frac{bslot_i + \beta}{islot_i - \beta}, \quad \textbf{(3)}$$

where β is the number of busy time slots occupied by f.

Let $m_i = \frac{l \times 8}{r_p \times (20 \times 10^{-6})}$ be the number of time slots needed for h_i to transmit a packet, where l is the packet length (in bytes) of f. The value of β was estimated in [17] as $(\lambda_i + \Delta\lambda) \times m_i \times (20 \times 10^{-6}) \times perd$, where $perd$ is the length (in number of time slots) of the given time period. After f is admitted, the MAC access delay $\tilde{d}_l$ can be estimated by (1) as well, if b_i is replaced with $\tilde{b}_l$. Notice that the busy/idle ratio of the channel is estimated as $b_i = \frac{bslot_i}{islot_i}$, if f is not admitted yet. After f is admitted, some busy time slots will be occupied by f, and the busy/idle ratio of the channel is estimated as $b_l = \frac{bslot_i + \beta}{islot_i - \beta}$, , where β represents the effect of f on the busy/idle ratio of the channel. In [17], the value of β was estimated, considering the effects of both the ongoing flows (i.e., λ_i) and f (i.e., $\Delta\lambda$). Differently, we consider the effect of f only, while estimating the value of β. That is, we compute $\beta = \Delta\lambda \times m_i \times (20 \times 10^{-6}) \times perd$.

Our estimation method of one-hop delay for h_i is as follows. First, the MAC access delay $\tilde{d}_l$ is estimated according to (1), where the busy/idle ratio b_i of the channel can be obtained by measuring. Then, the value of d_i is estimated according to (2). In case a new flow f within the carrier sense range of h_i is admitted, the value of d_i must be re-estimated, where b_i in (1) is replaced with $\tilde{b}_l$ (i.e., (3)). Notice that the value of d_i computed by (2) is a mean value. When it is used in our protocol (presented below), it is multiplied by α (≥ 1) so as to satisfy the delay requirement of f within a certain *confidence level*.

3.2 Delay-guaranteed Multicast Tree Determination

The rest of this section is organized as follows. The proposed algorithm is introduced in Section 3.2.1. In order to construct a delay-guaranteed multicast tree, a basic procedure is invoked iteratively, which can construct delay-guaranteed routes from all hosts to a given destination and minimizes the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts for each constructed route. The basic procedure is described in Section 3.2.2.

3.2.1 Construction of Multicast Trees

Given a requesting flow, denoted by Γ, the proposed algorithm aims to construct a delay- guaranteed multicast tree, with the objective of minimizing the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts. Each host is assigned with a weight, which is the sum of its packet transmission time and the total blocking time of its blocked hosts. The weight of a route is defined as the total weight of all hosts contained in it.

More concretely, let $\tilde{w}_i$ denote the weight of a host h_i and $r_{g(i)}$ denote the data rate used by h_i. Then,

$$\tilde{w}_i = \frac{l}{r_{g(i)}} \times (1 + | I_{i,r_{g(i)}} \cup | II_{i,r_{g(i)}} |),$$

where l is the packet size and $I_{i,r_{g(i)}}$ ($II_{i,r_{g(i)}}$) is the set of $r_{g(i)}$-one-hop ($r_{g(i)}$-two-hop) neighboring hosts of h_i. The hosts in $I_{i,r_{g(i)}}$ and $II_{i,r_{g(i)}}$ are blocked when h_i is transmitting packets with data rate $r_{g(i)}$.

Let D be a set of destinations, F be a set of forwarders, and $\Re$ be a set of data rates. Initially, let D be empty, F contain the source only, and $\Re$ contain the data rate of the source only. Then, it finds a minimum-weight route connecting some destination not in D with some forwarder (or destination) in $F \cup D$. After that, the destination is added to D and all hosts in the route are added to F. The above process is repeated until all destinations are included in D. At the end of the process, F contains all the forwarders.

Let h_s be the source, D^* be the set of all destinations, and d_req be the delay requirement for the requesting flow Γ. Also, given a route from a host h_i to a destination h_d, define $\Psi_{i,d}$ to be the set of hosts, exclusive of h_d, in the route, $R_{i,d}$ to be the set of data rates used by the hosts in $\Psi_{i,d}$, and $s_{i,d}$ to be the weight of the route (the weight of h_d is equal to zero). The algorithm, named *Delay_Guaranteed_Tree*, is presented below.

Procedure *Delay_Guaranteed_Tree*(h_s, D^*, d_req);
{
  $D \leftarrow \varnothing$; $F \leftarrow \{h_s\}$; $\Re \leftarrow \{r_{g(s)}\}$;
  /* Initially, let $r_{g(s)}$ be the maximal data rate. */
  repeat
    for each $h_d \in D^* - D$ **do**
      {
        Delay_Guaranteed_Route(h_d);
        /* After the procedure is invoked, the values of $\Psi_{i,d}$, $R_{i,d}$, and $s_{i,d}$
        for each host h_i are determined. */
        determine $h_x \in F \cup D$ with $s_{x,d} = \min\{s_{j,d} \mid h_j \in F \cup D\}$;
        $f(d) \leftarrow x$;
      };
    determine $h_{d^*} \in D^* - D$ with $s_{f(d^*),d^*} = \min\{s_{f(d),d} \mid h_d \in D^* - D\}$;
    $D \leftarrow D \cup \{h_{d^*}\}$; $F \leftarrow F \cup \Psi_{f(d^*),d^*}$; $\Re \leftarrow (\Re - \{r_{g(f(d^*))}\}) \cup R_{f(d^*),d^*}$;
  until $D = D^*$;
}.

There are three input parameters: h_s, D^*, and d_req for *Delay_Guaranteed_Tree*. A procedure, named *Delay_Guaranteed_Route*, is invoked to construct delay-guaranteed and minimal-weight routes from all hosts h_i to a specified destination, i.e., h_d. The values of $\Psi_{i,d}$, $R_{i,d}$, and $s_{i,d}$ are determined accordingly. Then, among these minimal-weight routes, the minimum-weight one, i.e., from $h_{f(d)}$ to h_d, is determined. *Delay_Guaranteed_Route* is executed for each $h_d \in D^* - D$ within the for-loop. Whenever the execution of the for-loop ends, the route from $h_{f(d^*)}$ to h_{d^*} is further determined, whose weight is minimum among all $h_{f(d)}$-to-h_d routes. Since the $h_{f(d^*)}$-to-h_{d^*} route is included as a part of the multicast tree, it is not necessary to consider the destination h_{d^*} again in subsequent execution of the for-loop (hence, h_{d^*} is added to D, $\Psi_{f(d^*),d^*}$ is added to F, and $R_{f(d^*),d^*}$ is added to $\Re$).

Throughout the execution of *Delay_Guaranteed_Tree*, the set D collects the destinations (i.e., h_{d^*}'s) that were connected to the current multicast tree, and the set $D^* - D$ collects the destinations that were not yet connected to the current multicast tree. The set F collects the forwarders in the current multicast tree, and the set $\Re$ collects the data rates used by the forwarders in F. Also notice that the data rate of a forwarder may be changed (hence, $r_{g(f(d^*))}$ is removed from $\Re$ before $R_{f(d^*),d^*}$ is added to $\Re$), whenever it is connected to another destination. The values of D, F, and $\Re$ need to be updated whenever a new h_{d^*} is found. When $D = D^*$, *Delay_Guaranteed_Tree* terminates with F being the set of the forwarders in the multicast tree. If no delay-guaranteed route from any host to h_d is found after invoking *Delay_Guaranteed_ Route*, then *Delay_Guaranteed_Tree* terminates without delay-guaranteed multicast trees. *Delay_Guaranteed_Route* is detailed in Section 3.3.2.

3.2.2 A Basic Procedure

Delay_Guaranteed_Route intends to construct delay-guaranteed minimal-weight routes from all hosts to h_d, by the aid of Dijkstra's shortest path algorithm [27]. Given a source vertex in a weighted graph, Dijkstra's algorithm can construct shortest (i.e., minimal-weight) paths from the source vertex to all other vertices. A multi-rate MANET is conveniently represented by a directed graph $G = (V, A)$, where each vertex in V uniquely corresponds to a host and each arc from u to v, denoted by $<u, v>$, in A means that v is within the transmission range of u when u uses the base rate. Clearly, $<u, v> \in A$ if and only if $<v, u> \in A$.

Each arc $<u, v>$ in A is assigned with a weight, denoted by $w_{u,v}$, which is calculated as follows.

$$w_{u,v} = \frac{l}{\hat{r}_{u,v}} \times (1 + | I_{u,\hat{r}_{u,v}} \cup_{u,\hat{r}_{u,v}} |), \tag{4}$$

where l is the packet size, $\hat{r}_{u,v}$ is a data rate, and $I_{u,\hat{r}_{u,v}}$ ($II_{u,\hat{r}_{u,v}}$) is the set of $\hat{r}_{u,v}$-one-hop ($\hat{r}_{u,v}$- two-hop) neighboring vertices of u. The weight of a (directed) path in G is defined as the total weight of the arcs contained in it. Notice that $w_{u,v}$ differs from $\tilde{w}_i$ in the data rate used.

Initially, $\hat{r}_{u,v}$ is set to the maximally available data rate with respect to h_j, where it is assumed that u and v correspond to hosts h_i and h_j, respectively. Here, *the maximally available data rate with respect to* h_j is the maximal data rate used by h_i such that h_j can successfully decode packets received from h_i, and it can be determined as $\max\{r_p \mid thrd_p \le SINR_{i,j} \text{ and } 1 \le p \le n\}$, where the $SINR_{i,j}$ is available in $T_{i,1}$. Then, after h_i becomes a forwarder (i.e., $h_i \in F$), $\hat{r}_{u,v}$ must be maintained with the data rate of $\min\{\hat{r}_{u,v}, r_{g(i)}\}$. Whenever $\hat{r}_{u,v}$ is changed, $w_{u,v}$ needs to be changed accordingly. Further, since *Delay_Guaranteed_Route* intends to construct minimal- weight routes from all hosts to h_d, we need to swap $w_{u,v}$ and $w_{v,u}$ for each pair of arcs $<u, v>$ and $<v, u>$ in A, in order to apply Dijkstra's algorithm which starts at h_d.

Also notice that during our construction of a delay-guaranteed multicast tree, the current multicast tree is maintained delay-guaranteed, i.e., the end-to-end delay from the source to each destination in the current multicast tree is not greater than *d_req*. Besides, the current multicast tree should not cause delay violation to any ongoing flow. As a consequence, the constructed multicast tree can avoid HRP/HMRP. *Delay_Guaranteed_Route* constructs delay- guaranteed routes by iteratively adding hosts to them. A host can be added to a route, only if there is no delay violation to the route, the current multicast tree, and all ongoing flows. Delay violation happens to a route (or a flow), if its end-to-end delay exceeds its delay requirement. The following paragraphs explain the execution of *Delay_Guaranteed_Route*, while it intends to add a host h_j to a route $h_d \to \ldots \to h_i$. Since the flow direction is reverse, we use ($h_j \to$) $h_i \to \ldots \to h_d$, instead of $h_d \to \ldots \to h_i \to h_j$, if necessary.

$$\frac{1}{\hat{r}_{u,v}} \times (1 + | I_{u,\hat{r}_{u,v}} \cup II_{u,\hat{r}_{u,v}} |)$$

Define $\tilde{F} = \{h_j\} \cup \Psi_{i,d} \cup F$ and $H = D \cup \tilde{F} \cup \cup_{h_k \in \tilde{F}} I_{k,r_{g(k)}} \cup II_{k,r_{g(k)}}$, where $\Psi_{i,d}$ is the set of hosts in $h_i \to \ldots \to h_d$ and $I_{k,r_{g(k)}}$ ($II_{k,r_{g(k)}}$) is the set of $r_{g(k)}$-one-hop ($r_{g(k)}$-two-hop) neighboring hosts of h_k. Intuitively, if $h_j \to h_i \to \ldots \to h_d$ is included in the multicast tree, then the hosts in $\tilde{F}$ will be forwarders (i.e., $F = \tilde{F}$). Besides, the one-hop delay of the hosts in H will be increased. It is necessary to check if delay violation happens to those routes that contain hosts in H.

For each $h_x \in H$, its one-hop delay is recalculated as follows. Define $M_x = \{h_k \mid h_k \in \tilde{F}$ and $h_x \in I_{k,r_{g(k)}} \cup II_{k,r_{g(k)}} \cup \{h_k\}\}$, where the data rate, i.e., $r_{g(j)}$, of h_j ($\in \tilde{F}$) is assigned with $\hat{r}_{u,v}$, assuming that u and v correspond to hosts h_i and h_j, respectively. That is, h_x will be blocked when h_k is transmitting packets.

The busy/idle ratio of the channel sensed by h_x can be estimated by the equation (3) in Section 3.1, where $\beta = \Sigma_{h_k \in M_x} \Delta\lambda \times m_k \times (20 \times 10^{-6}) \times perd)$ is the total number of busy time slots needed for the hosts in M_x to transmit the requesting flow Γ ($\Delta\lambda$ is the packet arrival rate of Γ and m_k is the number of time slots needed for h_k to transmit a packet with data rate $r_{g(k)}$). Then, the one-hop delay of h_x can be estimated by equations (1) and (2) in Section 3.1, where b_i in (1) is replaced by (3).

The hosts in $h_j \rightarrow h_i \rightarrow \ldots \rightarrow h_d$ are all contained in H, and the end-to-end delay of $h_j \rightarrow h_i \rightarrow \ldots \rightarrow h_d$ can be estimated by accumulating the one-hop delays of these hosts. There is a delay violation to $h_j \rightarrow h_i \rightarrow \ldots \rightarrow h_d$, if the end-to-end delay exceeds *d_req*. Similarly, the forwarders in the routes from the source h_s to the destinations in D are in H, and whether or not delay violation happens to these routes can be decided. For an ongoing flow over a route, its hosts are not necessarily contained in H, and there is a delay violation to it, if the total increment of the one-hop delays of the hosts in H that it passes exceeds its residual delay. The *residual delay* of a route is the amount of the delay requirement minus its end-to-end delay [28]. For an ongoing flow over a multicast tree, each source-to-destination route is considered individually, and the discussion is similar. If the addition of h_j to $h_d \rightarrow \ldots \rightarrow h_i$ does not cause delay violation, then $r_{g(j)}$ is set to $\hat{r}_{u,v}$, $\tilde{w}_j$ is set to $w_{u,v}$ (after swapping $w_{u,v}$ with $w_{v,u}$), and $s_{j,d}$ is set to $s_{i,d} + \tilde{w}_j$, where it is assumed that u and v correspond to h_i and h_j, respectively. The following is a description of *Delay_Guaranteed_Route*.

Procedure *Delay_Guaranteed_Route*(h_d);
{
/* V is the set of all hosts. */
set $\Psi_{i,d}$ and $R_{i,d}$ to be empty for each $h_i \in V$;
set $s_{i,d}$ to be infinity for each $h_i \in V - \{h_d\}$; $s_{d,d} \leftarrow 0$;
apply Dijkstra's algorithm to construct delay-guaranteed minimal-weight routes from all hosts to h_d, with the following modifications:

- $<h_i, h_j>$ is selected only if the addition of h_j to $h_d \rightarrow \ldots \rightarrow h_i$ does not cause delay violation, and
- if $<h_i, h_j>$ is selected, assign $\Psi_{j,d}$, $r_{g(j)}$, $R_{j,d}$, and $s_{j,d}$ with $\Psi_{i,d} \cup \{h_j\}$, $\hat{r}_{u,v}$, $R_{i,d} \cup \{r_{g(j)}\}$, and $s_{i,d} + w_{u,v}$, respectively, where it is assumed that u and v correspond to h_i and h_j, respectively;

/* Recall that arc weight $w_{u,v}$ is calculated according to the equation (4), and $w_{u,v}$ is swapped with $w_{v,u}$, in order to apply Dijkstra's algorithm. An additionally necessary condition for $<h_i, h_j>$ to be selected by Dijkstra's algorithm is that $h_d \rightarrow \ldots \rightarrow h_i \rightarrow h_j$ does not cause delay violation to those routes that contain hosts in H (please refer to the last two paragraphs for more details). */
}.

4 Simulation Results

The simulations are implemented using the Network Simulator 2 package [29]. The IEEE 802.11 distribution coordination function (DCF) with CSMA/CA was used as the MAC layer protocol. Each host was equipped with a radio transceiver. The two-ray ground model [30] was adopted to predict the signal power received by the receiver. With this model, the reflection from the ground was considered and the signal power attenuated as $1/d^2$, where d was the distance between the transmitter and the receiver. In our simulation, CBR traffic flows were injected into the network from the sources. Each host has a MAC FIFO transmission queue of size 64 packets.

The proposed delay-guaranteed multicast protocol was denoted as DGM. Further, we used DGM-M (DGM-S) to stand for the DGM using multiple rates (single rate). The simulation was conducted to compare the proposed one-hop delay estimation method with that of [21], verify the effectiveness of DGM-S in avoiding HRP/HMRP, and compare the performance of DGM-M and RAM [10], where RAM was a multicast protocol using multiple rates. Besides, performance comparison was made between DGM-M and DGM-S.

4.1 Avoidance of HRP and HMRP

The effectiveness of the proposed multicast tree construction algorithm (i.e., *Delay_Guaranteed_ Tree* and *Delay_Guaranteed_Route*) in avoiding HRP and HMRP was verified by means of end-to-end delay and success ratio. Given a unicast flow, let P be the set of data packets delivered by the source, and $P' \subseteq P$ be the set of data packets successfully received by the destination without delay. Then, the *success ratio* for the flow was defined as $|P'|/|P|$. The simulation environment was as follows. There were 50 hosts randomly positioned in an area of size $1{,}000 \times 1{,}000$ m^2, and three 64-Kbps unicast flows f_1, f_2, f_3 were generated at $t = 0$, 50, 100 (seconds), respectively, whose delay requirements were set to 0.05 seconds.

The three routes could be obtained by AQOR. Their average end-to-end delays were exhibited in Fig. 1, and their success ratios measured for a period of one second were exhibited in Fig. 2. Before $t = 100$, the delays of f_1, f_2 are lower than 0.03 seconds. However, after $t = 100$, the delays increased drastically, as a consequence that f_3 trigged HRP. Similarly, the success ratios dropped drastically after $t = 100$.

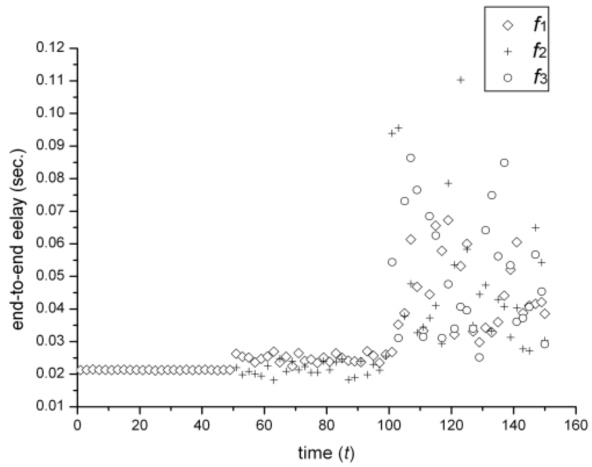

Fig. 1. Average end-to-end delays

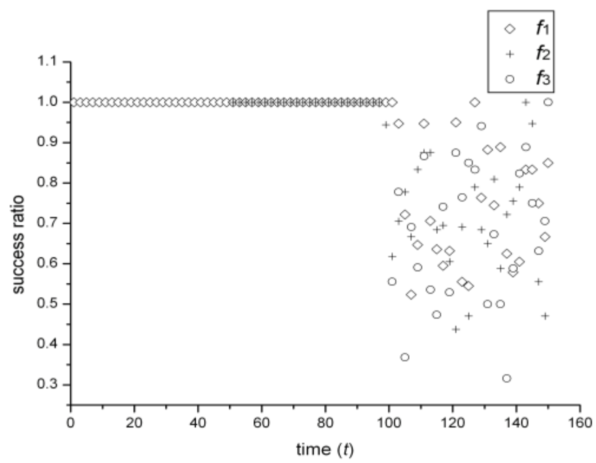

Fig. 2. Success ratios

On the other hand, given a multicast flow with three destinations h_d, $h_{d'}$, and $h_{d''}$, the multicast tree is obtained by [15] (with the base rate). It was assumed that the multicast flow is of 128-Kbps and its delay requirement is 0.1 second. The simulation proceeded for 150 seconds. The average end-to-end delays and success ratios for h_d, $h_{d'}$, and $h_{d''}$ were exhibited in Fig. 3 and Fig. 4, respectively. Observe that the average end-to-end delays for h_d and $h_{d''}$ exceeded the delay requirement, as a consequence of HMRP. Also, most of the success ratios for h_d and $h_{d''}$ were below 0.5 for the same reason.

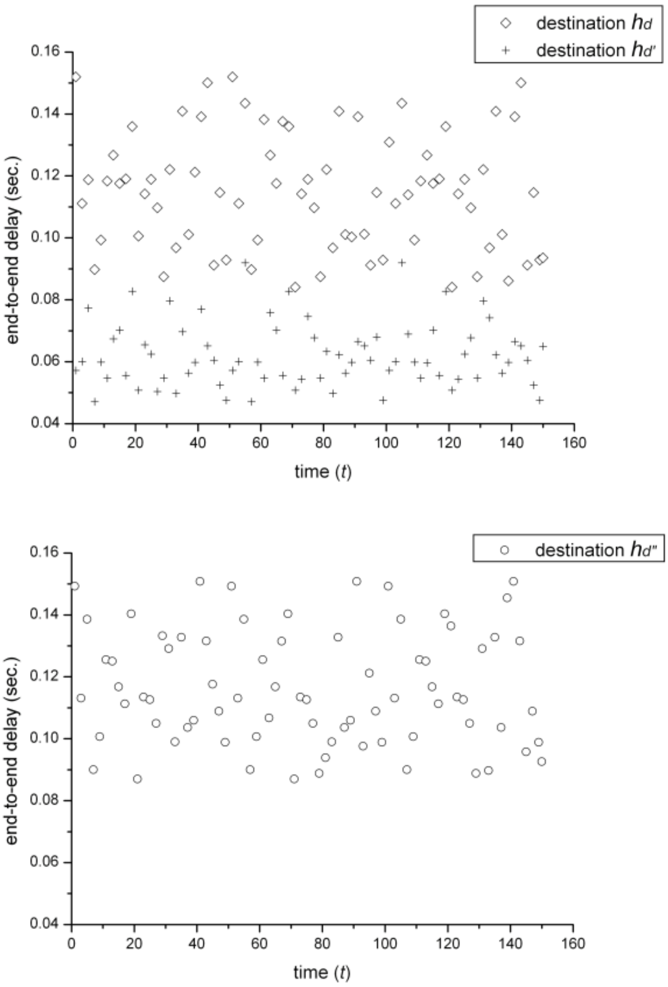

Fig. 3. Average end-to-end delays

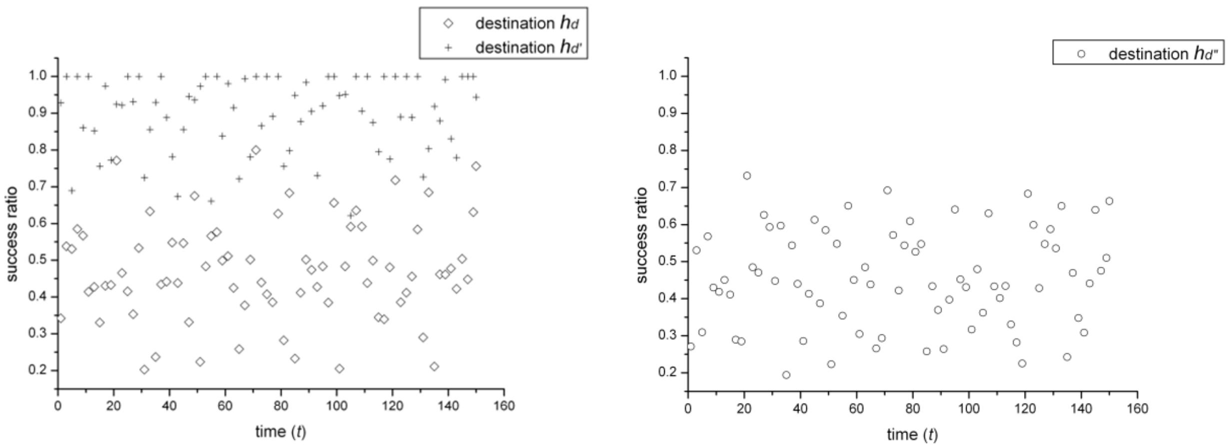

Fig. 4. Success ratios

Fig. 5 and Fig. 6 were obtained by repeating the simulations of Fig. 1 and Fig. 2 for DGM-S. The average end-to-end delays were smaller than the delay requirement (= 0.05 seconds), and the success ratios were higher than 0.95, in almost all simulation cases. Similarly, Fig. 7 and Fig. 8 were obtained by repeating the simulations of Fig. 3 and Fig. 4 for DGM-S. In all simulation cases, the average end-to-end delays were smaller than the delay requirement (= 0.1 second) and the success ratios were higher than 0.95.

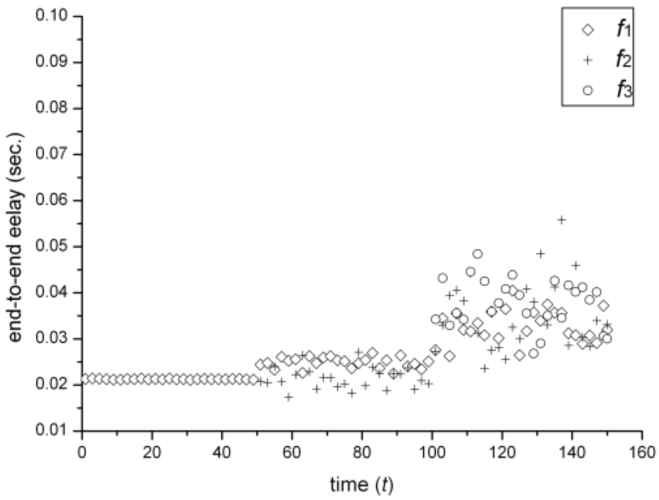

Fig. 5. Average end-to-end delays

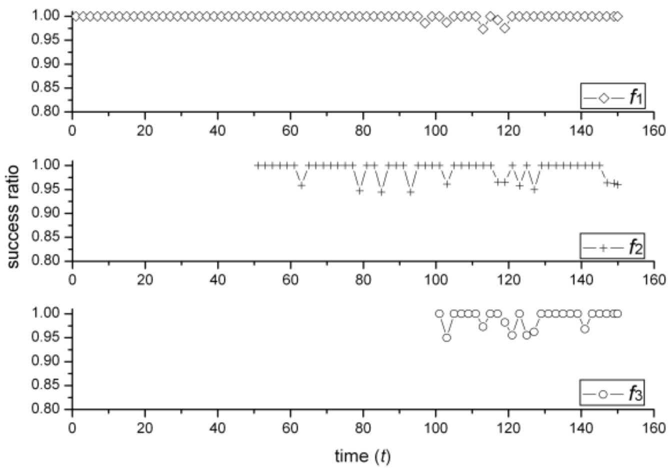

Fig. 6. Success ratios

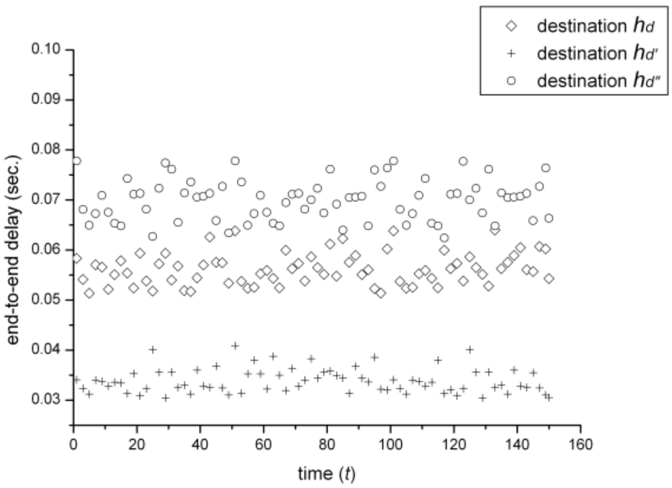

Fig. 7. Average end-to-end delays

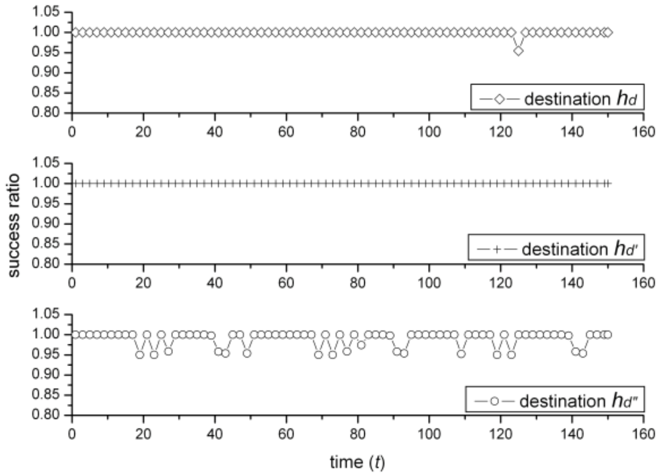

Fig. 8. Success ratios

4.2 Comparison of DGM-M and RAM

The multicast trees obtained by DGM-M were compared with the multicast trees obtained by RAM by means of end-to-end delay. DGM-M intended to construct multicast trees that minimized the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts, whereas RAM intended to construct multicast trees that minimized the total transmission time of the forwarders. It was assumed that there were 50 hosts randomly positioned in an area of size 1,000 × 1,000 m^2. Each multicast flow had three destinations, and the source and destinations were selected randomly from these 50 hosts.

In Fig. 9 and Fig. 10, the simulation results show the average end-to-end delays of the multicast trees constructed by DGM-M and RAM for different numbers of multicast flows, where each multicast flow was of 128-Kbps (200-Kbps). Forty instances were run for each simulation case, and their end-to-end delays were averaged. It was observed that RAM induced higher delays than DGM-M when the number of multicast flows exceeded 8 (Fig. 9) or 4 (Fig. 10), as a consequence of different construction strategies. By selecting forwarders with less transmission time and less total blocking time, DGM-M could mitigate congestion. In other words, DGM-M could accommodate more delay-guaranteed multicast flows (i.e., enhance the network capacity).

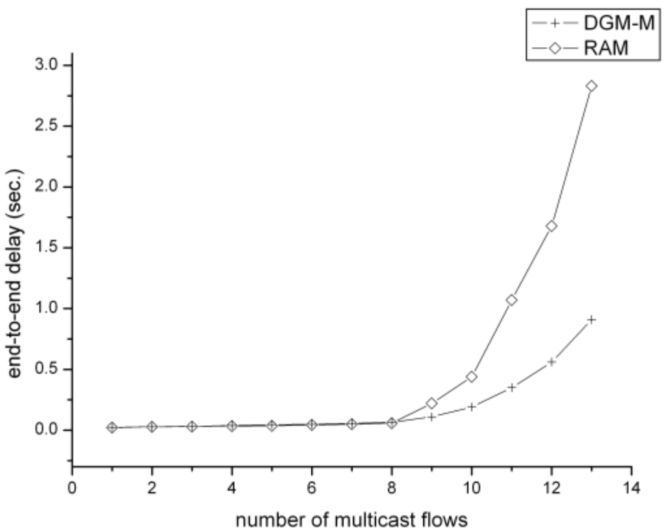

Fig. 9. Average end-to-end delays of 128-Kbps multicast flows

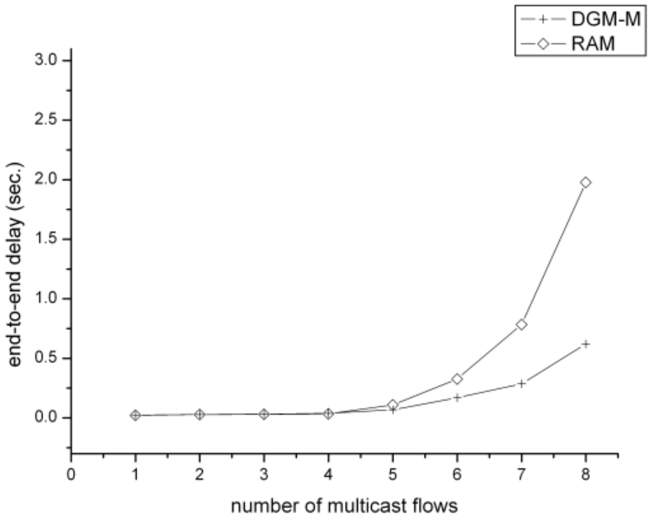

Fig. 10. Average end-to-end delays of 200-Kbps multicast flows

4.3 Comparison of DGM-M and DGM-S

It was assumed that the MANET was deployed over an area of size 1,000 × 1,000 m^2. Besides, each multicast flow had three destinations and was of 64-Kbps. The performances of DGM-M and DGM-S were compared. For each simulation case, forty instances were run and the average was taken. In Fig. 11, the simulation results show the average end-to-end delays of the multicast trees induced by DGM-M and DGM-S for different numbers of multicast flows, where there are 50 hosts randomly positioned in the MANET. In Fig. 12, the simulation results show the average admission ratio for different numbers of requesting multicast flows whose delay requirements is 0.05 seconds. The admission ratio is the ratio of the number of admitted multicast flows to the number of requesting multicast flows. It is observed that DGM-M had shorter end-to-end delays and higher admission ratios than DGM-S.

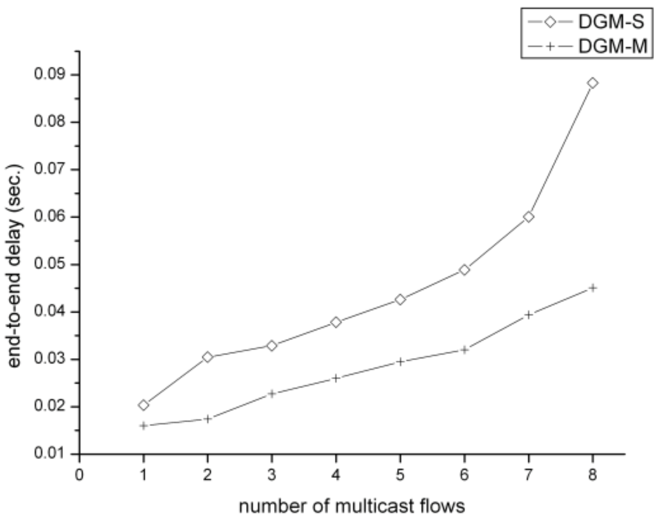

Fig. 11. Average end-to-end delays

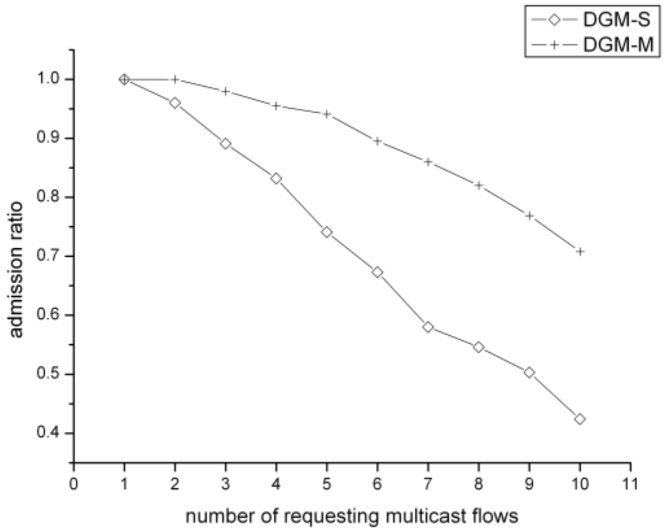

Fig. 12. Average admission ratios

5 Conclusion

In order to exploit wireless resources efficiently and provide QoS for real-time multicast services, a delay-guaranteed multicast protocol for multi-rate MANETs was proposed in this paper. The proposed multicast protocol can avoid HRP/HMRP, and provide delay guarantees for the requesting flow and ongoing flows. The proposed multicast protocol aimed to construct delay-guaranteed multicast trees for real-time multicast services in multi-rate MANETs. its objective is to minimize the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts. Both data rates and the numbers of neighbors of forwarders were taken into account in the construction of each source-to-destination route. Simulation results showed that our estimation method was more accurate than the estimation method in [21]. They also showed that when the network traffic was saturated, additional (30% ~ 40%) requesting flows could be admitted, if multiple data rates were provided.

Acknowledgments

This work was supported in part by the Fujian Provincial Key Laboratory of Cloud Computing and Internet-of-Thing Technology, China.

References

[1] R. Sivakumar, P. Sinha, V. Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm, IEEE Journal on Selected Areas in Communications 17(1999)1454-1465.

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

[3] L. Chen, W.B. Heinzelman, QoS-aware routing based on bandwidth estimation for mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 23(2005) 561-572.

[4] P. Sinha, R. Sivakumar, V. Bhanghavan, MCEDAR: multicast core-extraction distributed ad hoc routing, in: Proc. the IEEE Wireless Communications and Networking Conference, 1999.

[5] E. Pagani, G.P. Rossi, A framework for the admission control of QoS multicast traffic in mobile ad hoc networks, in: Proc. the Ninth ACM International Workshop Wireless Mobile Multimedia, 2001.

[6] A. Darehshoorzadeh, M. Dehghan, G.R.J. Motlagh, Quality of service support for ODMRP multicast routing in ad hoc networks, Lecture Notes in Computer Science 4686(2007) 237-247.

[7] X.Y. Gan, C. Feng, Z.D. Qin, G. Zhang, H.Y. Wu, L.Y. Fu, X.B. Wang, H.D. Ma, Relay assisted multicast with markov mobility, IEEE Transactions on Network Science and Engineering, 2019. DOI:10.1109/TNSE.2019.2902801.

[8] C.C. Hu, E.H.K. Wu, G.H. Chen, Bandwidth-satisfied multicast trees in MANETs, IEEE Transactions on Mobile Computing 7(2008) 712-723.

[9] J.L. Hsu, I. Rubin, Cross-layer multi-rate routing strategies in wireless multi-hop random access networks, in: Proc. the IEEE Global Telecommunications Conference, 2007.

[10] M. Genetzakis, V.A. Siris, A contention-aware routing metric for multi-rate multi- radio mesh networks, in: Proc. the IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2008.

[11] S.V. Mallapur, S.R. Patil, J. Agarkhed, Multicast routing protocol with mobility prediction for mobile ad hoc networks, in: TENCON 2018 - 2018 IEEE Region 10 Conference, 2018.

[12] R. Talak, S. Karaman, E. Modiano, Capacity and delay scaling for broadcast transmission in highly mobile wireless networks, in: Proc. the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2017.

[13] J.J. Ren, G.L. Zhang, D.M. Li, Multicast capacity for vanets with directional antenna and delay constraint under random walk mobility model, IEEE Access 5(2017) 3958-3970.

[14] U.T. Nguyen, X. Xiong, Rate-adaptive multicast in mobile ad-hoc networks, in: Proc. the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2005.

[15] Y.H. Chen, C.C. Hu, E.H.K. Wu, G.H. Chen, Delay-guaranteed multicast routing in multi-rate MANETs, in: Proc. the IEEE Global Telecommunications Conference, 2009.

[16] S.J. Lee, M. Gerla, On-demand multicast routing protocol in multihop wireless mobile networks, ACM/Kluwer Mobile Networks and Applications 7(2002) 441-453.

[17] A. Abdrabou, W. Zhuang, Stochastic delay guarantees and statistical call admission control for IEEE 802.11 single-hop ad hoc networks, IEEE Transactions on Wireless Communications 7(2008) 3972-3981.

[18] A. Abdrabou, W. Zhuang, Statistical QoS routing for IEEE 802.11 multihop ad hoc networks, IEEE Transaction on Wireless Communications 8(2009) 1542-1552.

[19] T. Sakurai, H.L. Vu, MAC access delay of IEEE 802.11 DCF, IEEE Transactions on Wireless Communications 6(5)(2007). DOI:10.1109/TWC.2007.360372

[20] Q. Zhao, D.H.K. Tsang, T. Sakurai, A simple and approximate model for nonsaturated IEEE 802.11 DCF, IEEE Transactions on Mobile Computing 8(2009) 1539-1553.

[21] A. Nafaa, A. Ksentini, On sustained QoS guarantees in operated IEEE 802.11 wireless LANs, IEEE Transactions on Parallel and Distributed Systems 19(2008) 1020-1033.

[22] H. Lim, D. Jung, J. Hwang, K.J. Park, J.C. Hou, Contention control for high end-to-end throughput performance of wireless mesh networks, in: Proc. the Third Annual International Wireless Internet Conference, 2007.

[23] A. Muqattash, M. Krunz, POWMAC: a single-channel power-control protocol for throughput enhancement in wireless ad hoc networks, IEEE Journal on Selected Areas in Communications 23(2005) 1067-1084.

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

[25] G. Holland, N. Vaidya, P. Bahl, A rate-adaptive MAC protocol for multi-hop wireless networks, in: Proc. the International Conference on Mobile Computing and Networking, 2001.

[26] Y. Xi, B.S. Kim, J.B. Wei, Q.Y. Huang, Adaptive multirate auto rate fallback protocol for IEEE 802.11 WLANs, in: Proc. Military Communications Conference, 2006.

[27] E.W. Dijkstra, A note on two problems in connection with graphs, Numerische Math. 1(1959) 269-271.

[28] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, Multicasting for multimedia applications, in: Proc. the IEEE INFOCOM, 1992.

[29] Network Simulator (Version 2). <http://www.isi.edu/nsnam/ns/>, 2018.

[30] C. Bettstetter, G. Resta, P. Santi, The node distribution of the random waypoint mobility model for wireless ad hoc networks, IEEE Transactions on Mobile Computing 2(2003) 257-269.