

# 行政院國家科學委員會專題研究計畫 成果報告

## 應用數學規劃在大型隨意網路上多重通訊之研究(3/3) 研究成果報告(完整版)

計畫類別：個別型  
計畫編號：NSC 95-2221-E-002-449-  
執行期間：95年08月01日至96年07月31日  
執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：陳健輝

計畫參與人員：碩士級-專任助理：蔡秉穎、林士傑、郭育良、陳毓訓、洪韶  
憶、洪俊竹、洪浩舜、黃秋溶

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 96 年 06 月 11 日

# 行政院國家科學委員會專題研究計畫成果報告

## 應用數學規劃在大型隨意網路上多重通訊協定之研究

Mathematical Programming Approaches to Multicasting on  
Large-Scale Mobile Ad-Hoc Networks

計畫編號：NSC 92-2213-E-260-018-

執行期限：93年8月1日至96年7月31日

主持人：陳健輝教授 國立暨南國際大學資訊工程學系

計畫參與人員：蔡秉穎，林士傑，郭育良，陳毓訓，洪韶憶，  
洪俊竹，洪浩舜，黃秋溶

### 中文摘要

為了有效管理封包之傳遞，近年來在大型隨意網路(MANET)上之多重通訊協定(multicast protocols)主要採取兩階層架構(two-tier infrastructures)之設計。但是這些多重通訊協定有以下三個缺點。第一，它們選取有較多相鄰通訊點之通訊點作為主幹通訊點(backbone hosts)來代理封包的遞送。此種選取方式易造成大量封包集中在這些主幹通訊點而形成網路瓶頸。第二，它們在選取主幹通訊點時未考量其移動性，因此易造成大量封包之遺失。第三，它們未能提供不同等級傳輸服務品質(Quality-of-Service)之功能。為滿足多媒體或其它有及時性需求之應用程式，多重通訊協定必須具備此功能。

在本計劃執行的第一年裏，我們針對大型隨意網路設計一個不同的主幹通訊點選取方式，為了能產生較短的多重通訊路徑(multicast routes)，我們選取離其他通訊點最近的通訊點作為主幹通訊點，並將此種主幹通訊點選取方式表成數學規劃之目標函數(objective functions)與限制條件(constraints)。然後，藉由解出數學規劃之一組有效解(feasible solution)，主幹通訊點便可決定。我們將運用這些主幹通訊點來完成一套符合我們需要的多重通訊協定。本計劃執行的第二年，我們進一步考量通訊點的移動性(Mobility)，以作為主幹通訊點選取方式的重要依據。我們選取離其他通訊點較近與有較長的連接時間(Connecting Time)的通訊點作為主幹通訊點。此種選擇主幹通訊點的方式的目標，不僅能建立較短的多重通訊路徑，而且能建立一個具有較少控制封包與較少遺失封包之穩定兩階層架構。

今年(本計劃執行的第三年)，為滿足多媒體或其它有及時性需求之應用程式，我們增加設計的多重通訊協定具備提供不同等級傳輸服務品質(Quality-of-Service)之功能。現有在隨意網路上提供傳輸服務品質之通訊協定與多重通訊協定，必須建立滿足應用程式頻寬需求之路徑，然而他們在設計上因為未考量 two-hop neighbors 的頻寬消耗狀況，所以其路徑會導致兩個違反頻寬需求之問題產生。今年，我們設計一個演算法能避免上述兩個問題之發生，而建立滿足應用程式頻寬需求之多重通訊樹(multicast trees)。另為了減少頻寬與電力之消耗，挑選具有最少傳輸點(forwarders)的多重通訊樹為我們的目標。實驗驗證此演算法能提高整體系統之產能。

**關鍵詞：**隨意網路，數學規劃，多重通訊，最佳化。

Recent multicast protocols for large-scale mobile ad hoc networks (MANETs) adopted two-tier infrastructures to achieve effective flooding schemes. They have three disadvantages. First, hosts with a large number of neighbors were selected as backbone hosts (BHs) for forwarding packets. It is very likely that these BHs will become traffic concentrations or bottlenecks of the networks. Second, they suffered from more lost packets because they did not consider host mobility in selecting BHs. Third, they did not provide quality-of-service (QoS) function. A multicast protocol is desired to provide different levels of QoS for multimedia and real-time applications.

In the first year of this project, a distinct strategy is proposed for constructing a two-tier infrastructure in a large-scale ad-hoc network. Hosts with a minimal number of hops to the other hosts rather than those with a maximal number of neighbors will be adopted as BHs in order to obtain shorter multicast routes. The problem of determining BHs can be formulated with linear programming. Then BHs will be determined by solving the formulated MP for a feasible solution. Finally, the new multicast protocol will be developed based on the selected BHs. In the second year of this project, host mobility is further taken into consideration in selecting BHs. Hosts with fewer hops and longer remaining connection time to the other hosts will be selected as BHs. The objective is not only to obtain short multicast routes, but also to construct a stable two-tier infrastructure with fewer lost packets.

In this year (the third year of this project), the proposed multicast protocol is enhanced to provide different levels of QoS for multimedia and real-time applications. Previous quality-of-service (QoS) routing/multicasting protocols in mobile ad-hoc networks determined bandwidth-satisfied routes for QoS applications. However, they suffer from two bandwidth-violation problems since the bandwidth consumption of two-hop neighbors is not taken into consideration. In this year, a novel algorithm that can avoid the two problems is proposed to construct bandwidth-satisfied multicast trees for QoS applications. Further, it also aims to minimize the number of forwarders so as to reduce bandwidth and power consumption. Simulation results show that the proposed algorithm can improve the network throughput.

**Keywords:** Ad-hoc network, distributed algorithm, linear programming, multicast

## 一、前言

Motivated by increasing importance of real-time and multimedia applications with different quality-of-service (QoS) requirements, e.g., VoIP and video-conference, several QoS-constrained multicast algorithms for multimedia communication in wired networks have been proposed in the literature [3-5]. These algorithms aim to construct least-cost multicast trees with the constraints of delay and/or bandwidth requirement.

A mobile ad-hoc network (MANET) is formed by a group of mobile hosts that can

communicate with one another without the aid of any centralized point or fixed infrastructure. Because of recent provision of high-speed wireless internet services, QoS-guaranteed applications are now crucial to new-generation wireless multimedia communication systems. To meet the QoS requirements of the applications, multicast protocols are required to construct routes with QoS guaranteed. On the other hand, cellular networks have fixed base stations as access points for wireless communication.

In contrast with cellular networks, MANETs do not need any infrastructure and their topologies may dynamically change in an unpredictable manner because mobile hosts are free to move. Besides, each transmission in MANETs is a local broadcast and each mobile host shares the common radio channel with all its neighbors. MANETS have much less available bandwidth than cellular and wired networks. An inadequate bandwidth reservation may degenerate network performance, which is more serious in MANETS because of shared channel and limited bandwidth.

Several routing protocols [6-8] and multicast protocols [9, 10] have been proposed for MANET QoS services. In [6], a QoS routing protocol, named CEDAR, was proposed, which constructed routes by selecting high-bandwidth and stable links. In [7], another QoS routing algorithm was proposed, which constructed routes satisfying delay and/or bandwidth requirements. In [8], another QoS routing protocol, named AQOR, was proposed, which estimated available bandwidth and end-to-end delay for admission control and bandwidth reservation in determining QoS-satisfied routes.

In [9], a QoS multicast protocol, named MCEDAR, was proposed, which is a direct extension of CEDAR. In [10], another QoS multicast protocol, named M-CAMP, was proposed, which adopted a measurement-based approach to estimating the bandwidth availability from the server to the clients. Then, probe packets were sent along the multicast tree to test if the network satisfied the bandwidth requirement. All QoS routing/multicasting protocols mentioned above adopted some admission control mechanisms for determining bandwidth-satisfied routes. However, they may lead to bandwidth violation, as described below.

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

Currently, two QoS routing protocols have been proposed by Yang et al. [19] and Chen et al. [20]. They considered the bandwidth consumption of the one-hop/two-hop hosts when a new flow with bandwidth requirement was permitted. So, HRP can be avoided. However, a QoS multicast protocol should determine multiple bandwidth-satisfied routes from a server to all clients concurrently. Another bandwidth-violation problem is induced if multiple flows are permitted concurrently. The problem is named the hidden multicast route problem (HMRP).

## 二、研究目的

Since bandwidth and battery power are limited in MANETs, they should be taken into consideration in routing/multicasting protocols. One way to reduce bandwidth and power consumption is to decrease the number of hosts (i.e., forwarders) participating in packet forwarding. The problem of finding a multicast tree with minimum number of forwarders can be modeled as the Steiner tree problem (STP) that is known to be NP-hard (see [11]). If the bandwidth consumption is considered additionally, the resulting problem is also NP-hard. In this paper, we study the problem of determining a bandwidth-satisfied multicast tree with minimum number of forwarders. The problem is referred to as BSMTP in the rest of this report, and we propose to provide a feasible solution to it.

A number of good heuristics for approximate STP [21-26] have been proposed. They can be classified into centralized approach [21-24] and distributed approach [25, 26]. In the centralized approach, a center node that is aware of the state of the whole network computes the tree. In the distributed approach, on the other hand, each node of the network actively contributes to the algorithm computation. However, the approximate Steiner trees determined in [21-26] are insufficient to characterize a good multicast tree for multimedia applications with different QoS requirements (delay and/or bandwidth).

The problem of determining a delay-satisfied multicast tree with minimum number of forwarders (denoted as DSMTP) is a constrained Steiner tree. The most relevant work in the area of DSMTP has been done in [2, 12, 27, 28]. The algorithm proposed in [27, 28] was based on a minimum spanning tree, which encapsulated the minimum delay routes between any two nodes and the corresponding delays, rooted at the server over the entire network. A delay-satisfied multicast tree was derived by removing the unnecessary edges from the spanning tree. On the other hand, the algorithm proposed in [2, 12] modified the cheapest insertion heuristic algorithm proposed in [1] to solve DSMTP. In [1], a tree rooted at the server connected to the nearest client, which was not connected, on each step.

BSMTP is similar to DSMTP in some aspects. They both intend to determine a multicast tree so that the number of forwarders in the tree is minimized. The major differences between BSMTP and DSMTP are the constraints, i.e., bandwidth and delay. In DSMTP, the delay of a route can be obtained easily by summing the delays of the edges along the route, where the delay on each edge is a predefined constant. But in BSMTP, The bandwidth increment to a host is not a constant. In MANETs, a host shares the radio channel with its neighbors so that its bandwidth increment induced by a new flow relies on whether it and its neighbors are selected as the forwarders for the flow. Similarly, bandwidth violation happened to its neighbors should be avoided when it is selected as a forwarder. Therefore, to determine a feasible solution of BSMTP is more difficult.

A novel algorithm that can avoid the two problems is proposed to construct bandwidth-satisfied multicast trees, which is a feasible solution to BSMTP, for QoS applications. Further, it also aims to minimize the number of forwarders so as to reduce bandwidth and power consumption. Simulation results show that the proposed algorithm can improve the network throughput.

### 三、文献探討

To meet the bandwidth requirements of bandwidth-constrained applications, a proper admission control mechanism is needed to judge if a host is allowed to forward the packets for a requested flow. In wired networks, there is a dedicated point-to-point link, denoted by  $l_{ij}$ , between two adjacent nodes  $v_i$  and  $v_j$ . If  $v_i$  ( $v_j$ ) transmits packets to  $v_j$  ( $v_i$ ), the bandwidth consumption is bounded by the maximal available bandwidth of  $l_{ij}$ . When a neighbor of  $v_i$  ( $v_j$ ) transmits packets to  $v_i$  ( $v_j$ ), it does not consume the bandwidth of  $l_{ij}$ . So,  $v_i$  and  $v_j$  are aware of the remaining available bandwidth of  $l_{ij}$ . Suppose that a new flow from the source  $v_s$  to the destination  $v_d$  is initiated and its bandwidth requirement is  $b\_req$ . Let  $b\_r_{ij}$  be the current remaining available bandwidth of  $l_{ij}$ . If  $b\_r_{ij} \geq b\_req$ , then  $v_i$  ( $v_j$ ) is allowed to forward the flow to  $v_j$  ( $v_i$ ). Finally, if  $v_s \rightarrow v_{f_1} \rightarrow v_{f_2} \rightarrow v_{f_3} \rightarrow \dots \rightarrow v_{f_m} \rightarrow v_d$  is determined as the bandwidth-satisfied route for the flow, then it should have  $\min\{b\_r_{s,f_1}, b\_r_{f_1,f_2}, b\_r_{f_2,f_3}, \dots, b\_r_{f_m,d}\} \geq b\_req$ .

The MANET QoS routing/multicasting protocols in [6, 7, 9, 10] borrowed the concept of point-to-point links from the wired networks to construct bandwidth-satisfied routes. However, a host in MANETs shares the radio channel with its neighbors so that the bandwidth is consumed not only by it, but also by its neighbors. For example, suppose that  $v_s \rightarrow v_{f_1} \rightarrow v_{f_2} \rightarrow v_{f_3} \rightarrow \dots \rightarrow v_{f_m} \rightarrow v_d$  is a bandwidth-satisfied route in MANETs. When  $v_{f_1}$  transmits packets to  $v_{f_2}$ , the bandwidths of  $l_{s,f_1}$ ,  $l_{f_1,f_2}$  and  $l_{f_2,f_3}$  are consumed. Therefore, it is more difficult to determine a bandwidth-satisfied route in MANETs, because the computation of  $l_{ij}$  also relies on the neighbors of  $v_i$  and  $v_j$ .

On the other hand, AQOR [8] determined a bandwidth-satisfied route differently. The bandwidth computation is from the viewpoint of a host, instead of a link. Let  $b\_r_i$  be the current remaining available bandwidth of  $v_i$ . With the same example of  $v_s \rightarrow v_{f_1} \rightarrow v_{f_2} \rightarrow v_{f_3} \rightarrow \dots \rightarrow v_{f_m} \rightarrow v_d$  above, the bandwidths of  $v_s$  and  $v_{f_1}$  are consumed when  $v_s$  transmits packets to  $v_{f_1}$ , the bandwidths of  $v_s$ ,  $v_{f_1}$  and  $v_{f_2}$  are consumed when  $v_{f_1}$  transmits packets to  $v_{f_2}$ , and so on. Consequently, the total bandwidth consumptions of  $v_s$ ,  $v_{f_1}$ ,  $v_{f_2}$ ,  $v_{f_3}$ , ...,  $v_{f_m}$  and  $v_d$ , caused by the flow, are  $2 \times b\_req$ ,  $3 \times b\_req$ ,  $3 \times b\_req$ ,  $3 \times b\_req$ , ...,  $3 \times b\_req$  and  $b\_req$ , respectively.

All QoS routing/multicasting protocols described above may suffer from two bandwidth-violation problems, i.e., HRP and HMRP, because the hosts in the neighborhood of ongoing flows fail to compute the bandwidth consumptions of those hosts that are two hops distant from them.

Refer to Figure 1, where an illustrative example 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  $c$ . Suppose that each host has the same available bandwidth, say 11 units, and the bandwidth requirements for the three flows are 2 (from  $e$  to  $f$ ), 7 (from  $g$  to  $h$ ) and 3 (from  $a$  to  $d$ ) units, respectively. If  $c$  serves as a forwarder, then total 9-unit bandwidth of  $c$  will be consumed ( $c$ , the predecessor of  $c$  and the successor of  $c$  each require 3-unit bandwidth). Since the ongoing flow from  $e$  to  $f$  is in the radio coverage of  $c$ , it consumes 2-unit bandwidth of  $c$ . Consequently, the remaining available bandwidth of  $c$  is  $11 - 2 = 9$  units, and so  $c$  is allowed to be a forwarder.

Now we turn our attention to the bandwidth consumption of  $e$ . Since both  $c$  and  $g$  are in the radio coverage of  $e$ , the bandwidth requirement of  $e$  is  $2 + 3 + 7 = 12$  units, which exceeds its available bandwidth (11 units). The reason for the bandwidth violation is that  $c$  was not aware of the ongoing flow from  $g$  to  $h$  when it was determined to be a forwarder. In short, the bandwidth violation happened because the ongoing flow from  $g$  to  $h$  was hidden from the new flow from  $a$  to  $d$ . The problem is henceforth referred to as the hidden route problem (HRP). In contrast to the hidden terminal problem [13], which arises in the MAC layer, the hidden route problem arises in the network layer.

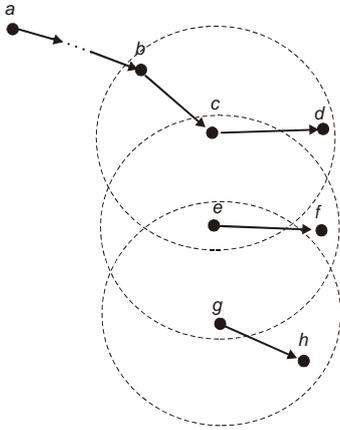


Figure 1. An example of HRP.

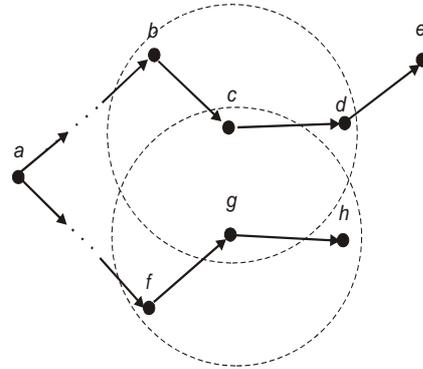


Figure 2. An example of HMRP.

An illustrative example for HMRP is shown in Figure 2, where there is a multicast group and a new flow from  $a$  (server) to  $e$  and  $h$  (clients) is permitted. Suppose that each host has the same available bandwidth, say 11 units, and the bandwidth requirement of the flow is 3 units. Also note that bandwidth reservation will be made for the flow when data flow through the routes. Both  $c$  and  $g$  can be forwarders for the flow from  $a$  to  $e$  and from  $a$  to  $h$ , respectively, because their bandwidth requirement (9 units) is smaller than their available bandwidth (11 units). However, since they are in the radio coverage of each other, 3-unit bandwidth is required additionally when data flow from  $a$  to  $e$  and  $h$ . This increases their bandwidth requirement to 12 units, which causes a bandwidth violation. The bandwidth violation happens because the two multicast routes from  $a$  to  $e$  and from  $a$  to  $h$  are mutually hidden from each other. It is henceforth referred to as the hidden multicast route problem (HMRP).

The influence of HRP and HMRP on network performance is demonstrated by simulations using the Network Simulator 2 package (ns-2) [14], where 50 hosts are randomly distributed over a  $1000\text{m} \times 1000\text{m}$  area. The data transmission capability (available bandwidth) of each host is assumed 800 Kbps. Two bandwidth requirements, i.e., 25 Kbps and 50 Kbps, for each flow are

assumed.

The probabilities of HRP and HMRP are investigated in Figure 3 and Figure 4, where forty runs with different seed numbers are conducted for each scenario and collected data for these runs are averaged. In Figure 3 (Figure 4), the number of source-destination pairs (multicast trees) varies from 1 to 20 (from 1 to 10). The sources and destinations are randomly chosen from 50 hosts and each multicast group has three clients. The bandwidth-satisfied routes (multicast trees) are obtained according to AQOR (MCEDAR). The probability of HRP (HMRP) means the probability that bandwidth violation happens to at least one route.

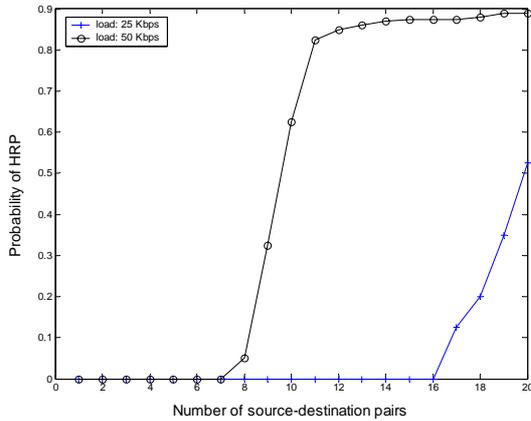


Figure 3. Probability of HRP.

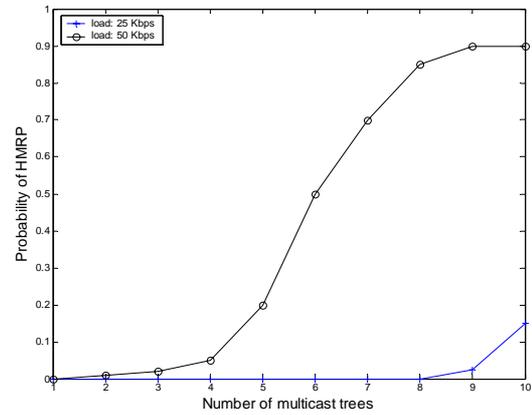


Figure 4. Probability of HMRP.

The simulations exhibit that when the network traffic is saturated, HRP and HMRP will happen very frequently to AQOR and MCEDAR, respectively, or AQOR and MCEDAR will be very likely to make improper bandwidth reservation.

#### 四、研究方法

A multicast group contains a server that is responsible for transmitting data packets to the clients in the same group. In this section, we propose a MANET bandwidth-aware multicasting protocol that can determine a bandwidth-satisfied multicast tree to admit a flow with the requested bandwidth. To offer bandwidth-guaranteed multicasting, the residual bandwidth at a given host must be known. However, how to calculate the residual bandwidth using the IEEE 802.11 MAC is still a challenging problem, because the bandwidth is shared among neighbors, and a host has no knowledge about other neighbors' traffic status [20]. In Section 3.1, we show a method to estimate residual bandwidth, which will act as a soft upper bound on the residual bandwidth of the host for bandwidth-satisfied multicast tree discovery/recovery.

Recall that each host shares the radio channel with its neighbors. So, its bandwidth increment induced by a new flow relies on whether it and its neighbors are selected as the forwarders for the flow. If it is selected as a forwarder, bandwidth violation to it and its neighbors must be avoided. Besides, all bandwidth-satisfied routes (which constitute a bandwidth-satisfied multicast tree) from a server to its clients must be determined concurrently, which is different from QoS routing protocols that determine a bandwidth-satisfied route in one time. In (2), an algorithm that can discover a bandwidth-satisfied multicast tree, which represents a feasible solution to BSMTTP, is proposed. The algorithm collects the entire network information in order to determine the

bandwidth- satisfied multicast routes concurrently.

The bandwidth-satisfied multicast tree determined in (2) may become outdated or disconnected due to host movement. In (3), a distributed algorithm is proposed to recover the tree.

### (1) Residual bandwidth estimation

It is rather difficult to estimate the residual bandwidth of a host, because the bandwidth is shared among neighboring hosts and these neighboring hosts are not aware of the traffic status of one another. Recently, there are two QoS routing protocols, proposed by Yang *et al.* [19] and Chen *et al.* [20], which estimate the residual bandwidth of a host by monitoring the amount of idle channel time. In this section, based on the same concept as [19] and [20], a method for residual bandwidth estimation is proposed.

As shown in Figure 5, each host  $v_i$  maintains two neighbor tables: the one-hop neighbor table and the two-hop neighbor table, where  $v_j$  ( $v_k$ ) denotes a one-hop (two-hop) neighbor of  $v_i$  and  $b_{o_j}$  ( $b_{o_k}$ ) denotes the currently consumed bandwidth for all ongoing flows that are forwarded by  $v_j$  ( $v_k$ ). Besides, there is a link directed from  $v_j$  to  $v_k$  if  $v_k$  is a one-hop neighbor of  $v_j$ .

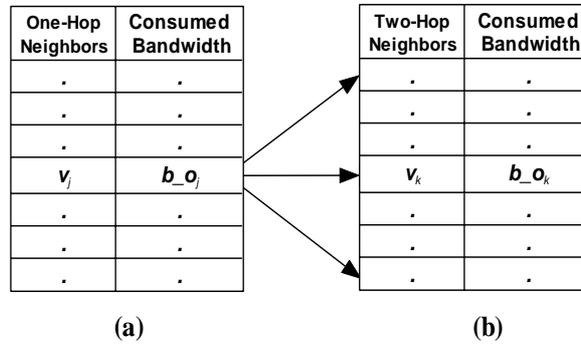


Figure 5. Two neighbor tables. (a) One-hop neighbor table. (b) Two-hop neighbor table.

In order to maintain the two neighbor tables for all hosts, every two neighboring hosts, say  $v_x$  and  $v_y$ , have to exchange their one-hop neighbor tables together with  $b_{o_x}$  and  $b_{o_y}$  periodically via control packets  $b\_hello$ . Whenever  $v_x$  ( $v_y$ ) receives a  $b\_hello$  packet from  $v_y$  ( $v_x$ ),  $v_x$  ( $v_y$ ) uses the one-hop neighbor tables of  $v_y$  ( $v_x$ ) that are carried by the  $b\_hello$  packet to update its two neighbor tables and the associated links.

When  $v_i$  has the currently consumed bandwidths, i.e.,  $b_{o_j}$ s, of all its one-hop neighboring hosts  $v_j$ s, its residual bandwidth can be estimated as the maximal available bandwidth minus  $b_{o_i} + \sum b_{o_j}$ , divided by a weighted factor  $\alpha$ . Also,  $v_i$  can estimate the residual bandwidths of all  $v_j$ s similarly. We need to divide these residual bandwidths by  $\alpha$  due to the IEEE802.11 MAC's nature and some overheads induced by the MAC. Since the control packets (i.e., RTS, CTS and ACK packets) consume bandwidth, the back-off scheme fails to use the entire bandwidth and the data/control packets may collide,  $\alpha$  can be calculated as follows.

$$\alpha = \frac{RTS + CTS + (Data + MACHdr + IPHdr) + ACK}{Data}$$

where  $RTS$ ,  $CTS$ ,  $Data$ ,  $MACHdr$ ,  $IPHdr$  and  $ACK$  denote the sizes of RTS, CTS, Data, MAC

header, IP header and ACK, respectively. Since fading errors may cause retransmission of control/data packets, the value of  $\alpha$  used in the algorithm (presented below) should be greater than the value calculated above.

## (2) Discovery of bandwidth-satisfied multicast trees

A heuristic algorithm that can provide a feasible solution to BSMTTP is proposed. The algorithm can admit a new flow with bandwidth requirement for some multicast service by constructing a bandwidth-satisfied multicast tree. Besides, the algorithm attempts to minimize the number of forwarders for the new flow. The execution of the algorithm involves a basic procedure, named *Shortest\_Routes*, which can establish shortest (minimum number of hops) routes to a particular destination.

*Shortest\_Routes* has four input parameters:  $v_d$ ,  $b\_req$ ,  $\Delta$  and  $\Phi$ , where  $v_d$  is a destination of the new flow,  $b\_req$  is the bandwidth requirement for the new flow, and  $\Delta$ ,  $\Phi$  are two sets of hosts. *Shortest\_Routes* also has two output parameters:  $H$  and  $P$  (explained later). The execution of *Shortest\_Routes* invokes another procedure, named *B\_Violation*, which returns *true* if bandwidth violation happens and *false* else. *B\_Violation* uses  $\Delta$  and  $\Phi$  for checking bandwidth violation, which will become clear later.

*Shortest\_Routes* is a modification of the well-known Dijkstra's shortest-path algorithm [15], which can compute the shortest paths from a node to the other nodes. Differently, *Shortest\_Routes* establishes shortest routes in a reverse direction from all hosts to  $v_d$ . For each host  $v_i$  in the shortest routes, let  $h_{i,d}$  denote the minimum number of hops from  $v_i$  to  $v_d$ , and  $P_{i,d}$  denote the set of forwarders along the shortest  $v_i$ - $v_d$  route. Initially, set  $h_{i,d} = 1$  and  $P_{i,d} = \{v_i\}$  if  $v_i$  is neighboring to  $v_d$  and the inclusion of  $v_i$  in the shortest routes will not cause bandwidth violation. Otherwise, set  $h_{i,d} = \infty$  and  $P_{i,d} = \{\}$  (the empty set). Host  $v_i$  will (will not) cause bandwidth violation if *B\_Violation*( $b\_req$ ,  $\Delta$ ,  $\{v_i\} \cup \Phi$ ) returns *true* (*false*).

The shortest routes are established iteratively by a repeat-until loop. In each iteration, a host  $v_x$  that is closest to  $v_d$  (i.e.,  $h_{x,d}$  is minimum) and was not selected before is determined. Then, for each neighbor  $v_j$  of  $v_x$ , the  $v_j$ - $v_d$  route (i.e.,  $P_{j,d}$ ) is replaced with the  $v_x$ - $v_d$  route augmented with the  $v_j$ - $v_x$  hop (i.e.,  $P_{x,d} \cup \{v_j\}$ ), if the new  $v_j$ - $v_d$  route is shorter and does not cause bandwidth violation. The latter can be checked by invoking *B\_Violation*( $b\_req$ ,  $\Delta$ ,  $\{v_j\} \cup P_{x,d} \cup \Phi$ ). Finally, the shortest routes are represented by means of  $H$  and  $P$ , which are two sets containing all  $h_{i,d}$ s and  $P_{i,d}$ s, respectively. *Shortest\_Routes* is detailed below.

**Procedure** *Shortest\_Routes*( $v_d$ ,  $b\_req$ ,  $\Delta$ ,  $\Phi$ ,  $H$ ,  $P$ );

/\*  $V$  is the set of all hosts. \*/

$X \leftarrow V - \{v_d\}$ ;

**for** each  $v_i \in X$  **do**

**if**  $v_i$  and  $v_d$  are neighboring and *B\_Violation*( $b\_req$ ,  $\Delta$ ,  $\{v_i\} \cup \Phi$ ) = *false*

**then**  $h_{i,d} \leftarrow 1$  and  $P_{i,d} \leftarrow \{v_i\}$

```

    else  $h_{i,d} \leftarrow \infty$  and  $P_{i,d} \leftarrow \{\}$ ;
repeat
    determine  $v_x \in X$  so that  $h_{x,d} = \min\{h_{i,d} \mid v_i \in X\}$  and delete  $v_x$  from  $X$ ;
    if  $h_{x,d} \neq \infty$ 
        then for each neighbor  $v_j$  of  $v_x$  do
            if  $h_{j,d} > h_{x,d} + 1$  and  $B\_Violation(b\_req, \Delta, \{v_j\} \cup P_{x,d} \cup \Phi) = false$ 
                then  $h_{j,d} \leftarrow h_{x,d} + 1$  and  $P_{j,d} \leftarrow P_{x,d} \cup \{v_j\}$ 
        until  $h_{x,d} = \infty$  or  $X = \{\}$ ;
     $H \leftarrow \{\}$  and  $P \leftarrow \{\}$ ;
    for each  $v_i \in V - \{v_d\}$  do
         $H \leftarrow H \cup \{h_{i,d}\}$  and  $P \leftarrow P \cup P_{i,d}$ .

```

Next, we explain the execution of  $B\_Violation$ .  $B\_Violation$  has three input parameters:  $b\_req$ ,  $\Delta$  and  $\Phi'$ , where  $b\_req$  and  $\Delta$  are inherited from  $Shortest\_Routes$  and  $\Phi'$  is a superset of  $\Phi$ . The hosts in  $\Delta$  ( $\Phi'$ ) are destinations (forwarders) of the new flow. Let  $N_i$  be the set of hosts that are neighboring to  $v_i$  and  $b\_r_i$  be the residual bandwidth of  $v_i$ . Also, let  $M = \Delta \cup \Phi' \cup \bigcup_{v_j \in \Phi'} N_j$ .

$B\_Violation$  checks if the forwarders in  $\Phi'$  cause bandwidth violation to every host in  $M$ . By means of checking the hosts in  $\bigcup_{v_i \in \Phi'} N_i$ , HRP can be avoided to the ongoing flows that pass through the one-hop neighbors of the forwarders in  $\Phi'$ . Bandwidth violation happens to a host  $v_i$  in  $M$  if the bandwidth increment induced by the forwarders in  $\Phi'$  exceeds  $v_i$ 's residual bandwidth, i.e.,  $|(N_i \cup v_i) \cap \Phi'| \times b\_req > b\_r_i$ . If bandwidth violation happens, then  $B\_Violation$  returns *true*. Finally,  $B\_Violation$  returns *false*, if no bandwidth violation happens.  $B\_Violation$  is detailed below.

**Procedure**  $B\_Violation(b\_req, \Delta, \Phi')$ ;

$$M = \Delta \cup \Phi' \cup \bigcup_{v_j \in \Phi'} N_j ;$$

**for** each  $v_i \in M$  **do**

**begin**

**if**  $|(N_i \cup v_i) \cap \Phi'| \times b\_req > b\_r_i$

**then return true;**

**end;**

**return false.**

The proposed heuristic algorithm, named *B\_Satisfied\_Multicast\_Tree*, has three input parameters:  $v_s$ ,  $D$  and  $b\_req$ , where  $v_s$  is the source (i.e., the server) of the new flow,  $D$  is the set of the destinations (i.e., the clients) of the new flow, and  $b\_req$  is the bandwidth requirement for the new flow. *B\_Satisfied\_Multicast\_Tree* intends to establish a bandwidth-satisfied multicast tree for the new flow. Let  $\Delta \subseteq D$  be a subset of the destinations and  $F$  be the set of forwarders in the multicast tree. Initially, set  $\Delta = \{\}$  and  $F = \{v_s\}$ .

The bandwidth-satisfied multicast tree is established iteratively by a repeat-until loop. Without loss of generality, assume  $D = \{v_{d_1}, v_{d_2}, \dots, v_{d_c}\}$ , where  $c = |D|$  and  $v_{d_i}$  is the  $i$ th client in  $D$ . In each iteration, a client  $v_y$  who was not selected before is determined and is connected to the server  $v_s$ . In the first iteration, the shortest bandwidth-satisfied routes from  $v_s$  to every  $v_{d_i}$  are selected from  $P_s$  (computed by *Shortest\_Routes*).  $P_{s,d_i}$ s are selected, which are the shortest routes from  $v_s$  to  $v_{d_i}$ s. Then, the client  $v_y$  with the minimum distance to  $v_s$  among the all clients is determined.  $F(\Delta)$  is replaced with  $F \cup P'_{s,y}(\Delta \cup \{v_y\})$  and  $v_y$  is removed from  $D$ . It should be noted that  $P'_{s,y}$  is a bandwidth-satisfied route from  $v_s$  to  $v_y$ .

In the second iteration, a bandwidth-satisfied multicast tree will be determined. For every client  $v_{d_i}$  in  $D$ , a forwarder  $v_x \in F$  that is closest to  $v_{d_i}$  is first determined. Then,  $P_{x,d_i} \in P$  is selected as the shortest route to  $v_{d_i}$ . Then, the client  $v_y$  with the minimum distance to the forwarders in  $F$  is determined and  $F(\Delta)$  is updated. Now  $F$  represents a tree that connects  $v_s$  with two clients. The tree is bandwidth-satisfied (hence can avoid HMRP), as a consequence of executing *Shortest\_Routes*( $v_{d_i}, b\_req, \Delta, F, H, P$ ) in the first iteration. The execution for the other iterations is very similar. Finally, when the execution of *B\_Satisfied\_Multicast\_Tree* terminates,  $F$  represents a bandwidth-satisfied multicast tree that connects  $v_s$  with  $v_{d_1}, v_{d_2}, \dots, v_{d_c}$ . *B\_Satisfied\_Multicast\_Tree* is detailed below.

**Procedure** *B\_Satisfied\_Multicast\_Tree*( $v_s, D, b\_req$ );

$\Delta \leftarrow \{\}$  and  $F \leftarrow \{v_s\}$ ;

```

repeat

     $D' \leftarrow \{\}$ ;

    for  $i \leftarrow 1$  to  $|D|$  do

        begin

            Shortest_Routes( $v_{d_i}, b\_req, \Delta, F, H, P$ );

            determine  $v_x \in F$  so that  $h_{x,d_i} = \min\{h_{j,d_i} \mid v_j \in F\}$ ;

            if  $h_x \neq \infty$ 

                then  $D' \leftarrow D' \cup \{v_x\}$ ,  $h'_{x,d_i} \leftarrow h_{x,d_i}$  and  $P'_{x,d_i} \leftarrow P_{x,d_i}$ 

                else exit

            end

            determine  $v_y \in D'$  so that  $h'_{k,y} = \min\{h'_{k,y} \mid v_k \in F\}$ ;

             $F \leftarrow F \cup P'_{k,y}$  and  $\Delta \leftarrow \Delta \cup \{v_y\}$  and delete  $v_y$  from  $D$ ;

until  $D = \infty$ .

```

Recall that the adjacency of hosts is necessary to *Shortest\_Routes*. Also, all  $b\_r_i$ s are necessary to *Shortest\_Routes* and *B\_Violation*. Hence, they should be made available before *B\_Satisfied\_Multicast\_Tree* is invoked.

The new proposed algorithm is responsible to collect the two items of data. The adjacency of hosts is conveniently represented by an adjacency 0/1 matrix, where a 1 (0) in the  $(v_x, v_y)$ - entry denotes that  $v_x$  and  $v_y$  are (are not) neighboring. Two new control packets named *b\_join\_query* and *b\_join\_reply* are used, instead of *join\_query* and *join\_reply*. Let  $v_s$  denote the server and  $v_i$  denote an arbitrary host. Five variables:  $t$ , *hopcount*,  $N_i$ ,  $\check{N}_i$  and  $B\_r_i$  are introduced, where

- $t$ : the elapsed time required to transmit a packet between two neighboring hosts;
- hopcount*: the number of hops distant from  $v_s$ ;
- $N_i$ : the set of all neighbors of  $v_i$ ;
- $\check{N}_i$ : the set of  $N_j$ s collected by  $v_i$ ;
- $B\_r_i$ : the set of  $b\_r_j$ s collected by  $v_i$ .

When  $v_s$  intends to initiate a flow with bandwidth requirement  $b\_req$  to the clients, it broadcasts a *b\_join\_query*, which carries *hopcount*, over a region with a radius of *maxhopcount* hops centered at  $v_s$ , where *maxhopcount* is a predefined integer. Upon receiving a non-redundant *b\_join\_query* from a host, say  $v_\alpha$ ,  $v_i$  relays it to its neighbors and then waits for a time period to receive *b\_join\_replies* from its neighbors. If timeout happens,  $v_i$  replies a *b\_join\_reply* to  $v_\alpha$ , which carries  $\check{N}_i$  and  $B\_r_i$ .

Initially, set  $\check{N}_i = \{N_i\}$  and  $B_{r_i} = \{b_{r_i}\}$ . Upon receiving a  $b\_join\_reply$  from a host, say  $v_\beta$ ,  $\check{N}_i$  ( $B_{r_i}$ ) is replaced with  $\check{N}_i \cup \check{N}_\beta$ , ( $B_{r_i} \cup B_{r_\beta}$ ). The new algorithm is elaborated below.

(1) The server  $v_s$  performs the following.

(1.1) Set  $hopcount = 0$ .

(1.2) Transmit a  $b\_join\_query(hopcount)$  to its neighbors.

(1.3) Wait for  $b\_join\_replies$  from its neighbors for a time period of  $2 \times (maxhopcount - 1) \times t$ .

(2) Upon receiving a non-redundant  $b\_join\_query(hopcount)$  from a host, say  $v_\alpha$ , a host  $v_i$  performs the following.

(2.1) Set  $\check{N}_i = \{N_i\}$  and  $B_{r_i} = \{b_{r_i}\}$ .

(2.2) If  $hopcount < maxhopcount - 1$ , perform the following.

(2.2.1) Set  $hopcount = hopcount + 1$ .

(2.2.2) Transmit  $b\_join\_query(hopcount)$  to its neighbors.

(2.2.3) Wait for  $b\_join\_replies$  from its neighbors for a time period of  $2 \times (maxhopcount - 1 - hopcount) \times t$ .

(2.3) If  $hopcount = maxhopcount - 1$ , reply a  $b\_join\_reply(\check{N}_i, B_{r_i})$  to  $v_\alpha$ .

(3) Upon receiving a  $b\_join\_reply(\check{N}_\beta, B_{r_\beta})$  from a host, say  $v_\beta$ , a host  $v_i$  performs the following.

(3.1) Set  $\check{N}_i = \check{N}_i \cup \check{N}_\beta$  and  $B_{r_i} = B_{r_i} \cup B_{r_\beta}$ .

When timeout happens at step (2.2.3),  $v_i$  replies a  $b\_join\_reply(\check{N}_i, B_{r_i})$  to  $v_\alpha$ . Finally, the algorithm terminates when timeout happens at step (1.3), and  $v_s$  can obtain the adjacency 0/1 matrix for all hosts and all  $b_{r_i}$ s from  $\check{N}_s$  and  $B_{r_s}$ , respectively. Recall that the adjacency of hosts and all  $b_{r_i}$ s are necessary to  $B\_Violation$  and  $Shortest\_Routes$ , which are two procedures in  $B\_Satisfied\_Multicast\_Tree$ . Then,  $v_s$  can discover a bandwidth-satisfied multicast tree by invoking  $B\_Satisfied\_Multicast\_Tree$ .

### (3) Recovery of bandwidth-satisfied multicast trees

Two control packets, i.e.,  $b\_join\_query$  and  $b\_join\_reply$ , are used to collect the network information in order to discover bandwidth-satisfied multicast trees. An inadequate multicast tree is likely to be established if some control packets are lost. Besides, routes may become outdated or disconnected due to host movement. In this section, a distributed algorithm is proposed to recover inadequate multicast trees

A client can monitor the receiving packets delivered from the server in order to check that the receiving rate is bandwidth-satisfied. Once it detects an un-satisfied route, it initiates a distributed algorithm to find a new bandwidth-satisfied route to the server. The found route reconnects the client to the server. In the case, the client broadcasts a control packet  $b\_recovery\_query$ , which carries  $F = \{\}$  ( $F$  is the set of forwarders in the found route), to its neighbors. When a host  $v_i$  (i.e., a neighbor) receives the packet,  $v_i$  checks if the requested bandwidth causes bandwidth violation to  $v_i$ , i.e.,  $|(N_i \cup v_i) \cap F| \times b\_req > b\_r_i$ , or  $v_j$ s (neighbors of  $v_i$ ), i.e.,  $|(N_j \cup v_j) \cap F| \times b\_req > b\_r_j$ . Recall that  $v_i$  can estimate  $b\_r_i$  and  $b\_r_j$  by the method of Section 3.1. If not,  $v_i$  replaces  $F$  with  $F \cup \{v_i\}$  and then forwards the packet. Otherwise, it discards the received packet.

A host who receives the forwarding packet executes the same checking and forwarding procedure. Once a host along the existing multicast tree receives the packet, it replies another control packet  $b\_recovery\_reply$  to the client along the reverse route. The client may receive multiple replied packets from multiple hosts. Since we attempt to minimize the number of forwarders along the recovery tree, the client selects the host with the minimum number of hops to it and connects to the server by the corresponding reverse route to this host. Then, the multicast tree can be recovered.

## 五、結果與討論

Simulations are implemented using the Network Simulator 2 package (ns-2, version 2.27) [14]. IEEE 802.11 distribution coordination function (PCF) is used as the MAC layer protocol. Packets are sent using the un-slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Each host is equipped with a radio transceiver whose transmission range is up to 250 meters over a wireless channel. We use Two Ray Ground model to predict the signal power received by the receiver. In this model, reflection of ground is considered and the power of a signal attenuates as  $1/d^2$ , where  $d$  is the distance between the two hosts.

In the simulations, CBR data traffic flows are injected into the network from the servers and the size of data payload is 512 bytes. Each host has a MAC layer FIFO transmission queue of 64 packets at maximum.

Three performance measures: receiving rate, admission rate and number of control bytes are adopted. The receiving rate is the ratio of the number of data packets received by clients to the number of data packets delivered from servers. If bandwidth violation happens, it will drop drastically. The admission rate is the ratio of the number of multicast groups admitted to the number of multicast groups requested. When the admission rate goes up, the network performance increases. On the other hand, the number of control bytes can reflect the overheads incurred for constructing/maintaining the multicast routes.

For convenience, we use Heu to denote our bandwidth-satisfied multicasting protocol. Since  $B\_Satisfied\_Multicast\_Tree$  is heuristic, an optimal algorithm for BSMTP is necessary in order to evaluate the performance of  $B\_Satisfied\_Multicast\_Tree$ . In [16], the authors formulated BSMTP as a 0/1 integer linear programming, which can be well solved by a branch-and-bound algorithm (see [17]). We use Opt to denote the 0/1 integer linear programming solved by such a branch-and-bound algorithm. Opt can serve as a benchmark for evaluating the performance of

### *B\_Satisfied\_Multicast\_Tree.*

The simulations are performed with four aspects. First, we use two examples to show the effectiveness of Heu in avoiding HRP and HMRP. Second, performance comparison is made among Heu, Opt and MCEDAR under the assumption of static hosts. Third, the same performance comparison is made for mobile hosts. Fourth, the performance of Heu is investigated by assigning different values of network size and group size. Forty runs with different seed numbers are conducted for each scenario and collected data for these runs are averaged.

#### **(—) Avoidance of HRP and HMRP**

We first present a simple simulation in a 1000m×1000m area with 50 randomly positioned hosts. Figure 6 shows the three routes constructed by AQOR for three flows, i.e., Flow 1, Flow 2 and Flow 3. Each flow requires 140 Kbps. Flow 1 starts transmission first, Flow 2 starts transmission after 2000 seconds elapsed, and Flow 3 starts transmission after 4000 seconds elapsed. Their receiving rates, which are collected every 10 seconds, are exhibited in Figure 7. During the first 4000 seconds, the receiving rates of Flow 1 and Flow 2 approach one. However, after 4000 seconds elapsed, the receiving rate of Flow 2 drops drastically, as a consequence of HRP triggered by Flow 3. Bandwidth violation occurs at two forwarders for Flow 2.

Figure 8 shows the multicast tree constructed by MCEDAR for a multicast group, which carries a flow from the server to three clients, i.e., Client 1, Client 2 and Client 3. The flow requires 140 Kbps. Although each server-to-client route is self-bandwidth-satisfied, the multicast routes as a whole are not bandwidth-satisfied. When the flow goes over the multicast tree, HMRP is triggered and bandwidth violation occurs at two forwarders of the route to Client 2. The receiving rates are exhibited in Figure 9.

Figure 10 and Figure 11 (Figure 12 and Figure 13) demonstrate the effectiveness of Heu in avoiding HRP (HMRP) that was triggered in Figure 6 and Figure 7 (Figure 8 and Figure 9), respectively. Two additional QoS routing protocols, proposed by Yang *et al.* [19] and Chen *et al.* [20], are also simulated and evaluated in Figure 10 and Figure 11.

As observed from Figure 10, the three protocols (i.e., Heu, protocol of [19] and protocol of [20]) construct the same bandwidth-satisfied routes for Flow 1, Flow 2 and Flow 3, because they all try to determine shorter routes. The receiving rates for the three flows are exhibited in Figure 11, which are all equal to one, because the bandwidth consumption of the two-hop hosts is considered in these protocols.

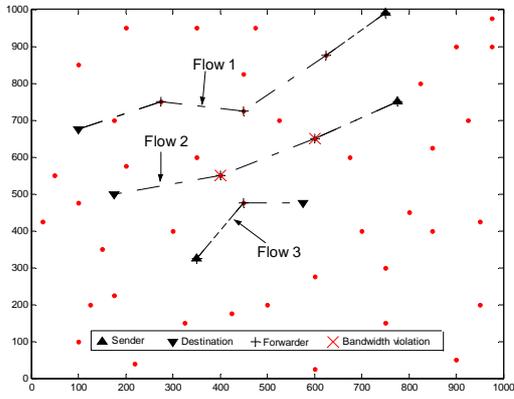


Figure 6. Three routes constructed by AQOR.

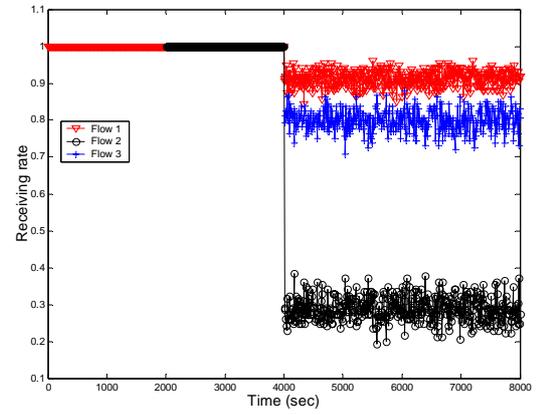


Figure 7. Receiving rates for the three flows of Figure 6.

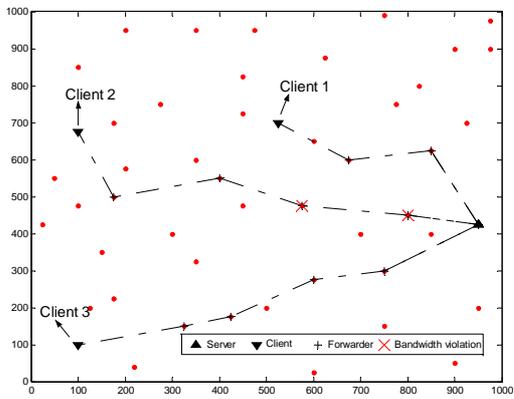


Figure 8. The multicast tree constructed by MCEDAR.

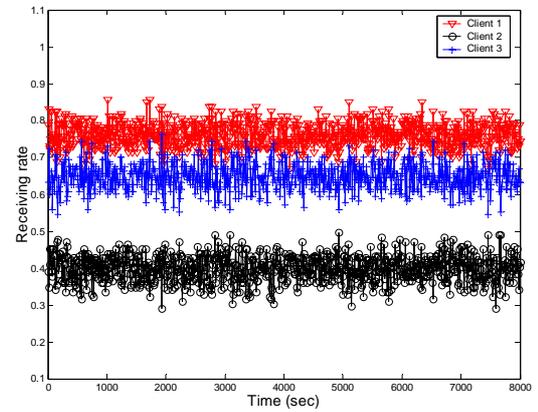


Figure 9. Receiving rates for the three clients of Figure 9.

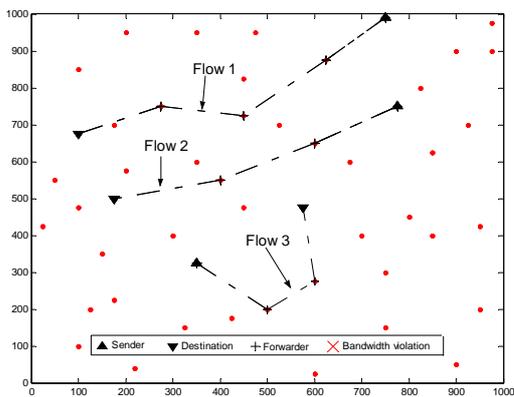


Figure 10. Three bandwidth-satisfied routes constructed by Heu, protocol of [19] and protocol of [20].

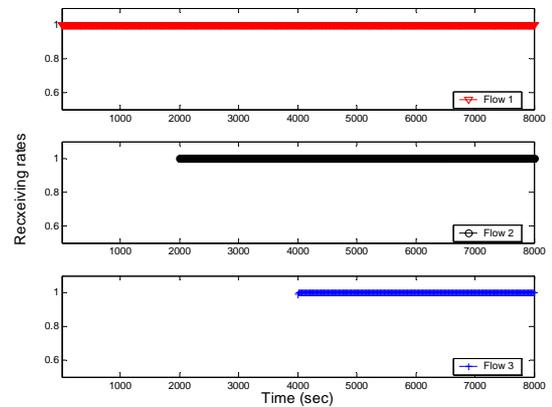


Figure 11. Receiving rates for the three flows of Figure 10.

Figure 12 shows the bandwidth-satisfied multicast tree constructed by Heu for Client 1,

Client 2 and Client 3. The receiving rates for the three clients are exhibited in Figure 13, which are all equal to one. For the two illustrative examples, Heu can avoid HRP and HMRP, while AQOR induces HRP and MCEDAR induces HMRP.

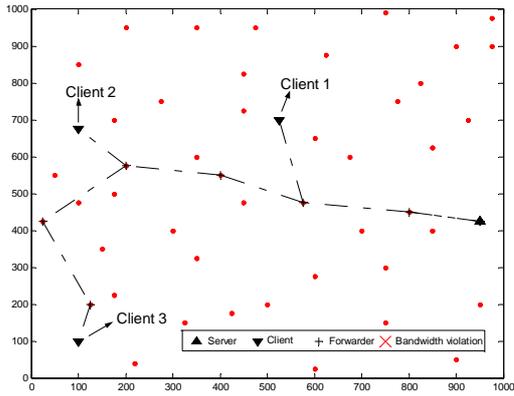


Figure 12. The bandwidth-satisfied multicast tree constructed by Heu.

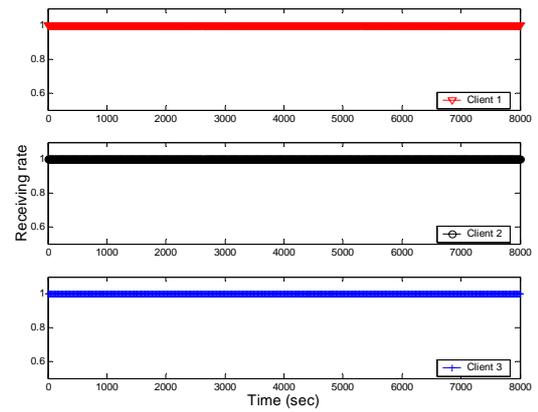


Figure 13. Receiving rates for the three clients of Figure 12.

## (2) Performance comparison: static hosts

The simulation environment is the same as that adopted in Section 2.4, where 50 hosts are randomly distributed over a 1000m×1000m area. Ten multicast groups, denoted by  $G_1, G_2, \dots, G_{10}$ , are randomly created. They are identical with those created in Section 2.4. Each multicast group consists of one server and three clients. A flow requiring 50 Kbps is sent from the server to the clients. The simulations proceed for 1000 seconds;  $G_1$  starts first,  $G_2$  starts after 100 seconds elapsed,  $G_3$  starts after 200 seconds elapsed, and so on. Figure 14 and Figure 15 compare the admission rates and receiving rates of Heu, Opt and MCEDAR under the assumption of static hosts.

Refer to Figure 14. When the number of multicast groups is smaller than 5, Heu has higher admission rates than MCEDAR, as a consequence that the multicast trees constructed by it with fewer forwarders. On the other hand, when the number of multicast groups exceeds 5, Heu has lower admission rates than MCEDAR. Recall that the probability that MCEDAR induces HMRP jumps up when the number of multicast groups exceeds 5 (refer to Figure 4). Also, Figure 15 demonstrates that the receiving rate of MCEDAR drops drastically when the number of multicast groups exceeds 5. When the number of multicast groups increases to 10, the receiving rate of Heu is still high ( $> 0.9$ ), but the receiving rate of MCEDAR drops to 0.65.

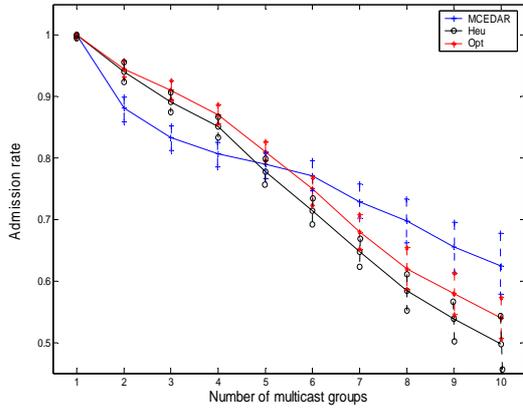


Figure 14. Admission rate for static hosts.

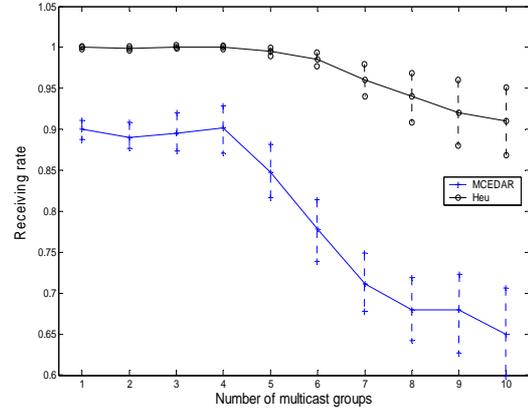


Figure 15. Receiving rate for static hosts.

When there are feasible solutions to BSMTP, Opt can always determine the optimal one. However, it is likely that Heu fails to find a feasible one. Figure 14 also reveals the success rate of Heu in finding a feasible solution to BSMTP. The success rate can be estimated by means of the ratio of the admission rate of Heu to the admission rate of Opt. For one extreme case in which there is one multicast group, the success rate is estimated as  $1/1 = 1$ . For the other extreme case in which there are ten multicast groups, the success rate is estimated as  $0.497/0.54 = 0.92$ . For the other cases, the success rate ranges from 0.92 to 1.

### (3) Performance comparison: mobile hosts

The simulation environment models a network of 50 hosts distributed randomly over a  $1000\text{m} \times 1000\text{m}$  area. Host mobility is based on the random waypoint model [18], in which a host's movement consists of a sequence of random length intervals, called *mobility epochs*. During each epoch, a host moves in a constant direction and at a constant speed. The simulations proceed for 1000 seconds and the speed varies from 5 to 20 meters per second (or 18 to 72 kilometers per hour). The first seven multicast groups, i.e.,  $G_1, G_2, \dots, G_7$ , that were created in Section 4.2 are used in the simulations. A flow requiring 50 Kbps is sent from the server to the clients.

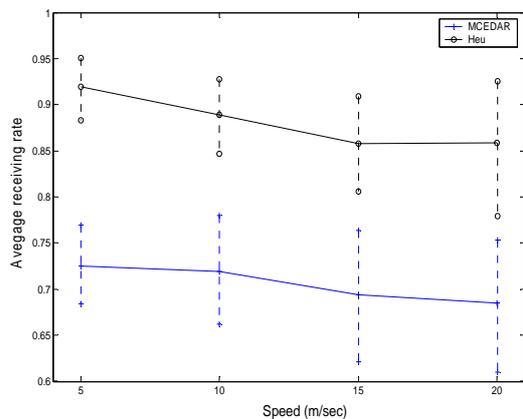


Figure 16. Receiving rate for mobile hosts.

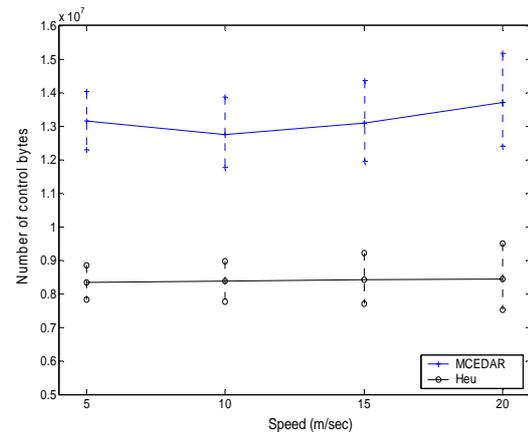


Figure 17. Number of control bytes for mobile hosts.

Figure 16 compares the receiving rates of Heu and MCEDAR under the assumption of mobile hosts. Like the previous situation of static hosts, Heu has higher receiving rates than

MCEDAR. Figure 17 shows that Heu generates fewer control bytes than MCEDAR. Control bytes are generated for constructing a new bandwidth-satisfied multicast tree if the current multicast tree cannot satisfy the bandwidth requirement.

## 六、参考文献

- [1] S. Voss, "Steiner's problem in graphs: heuristic methods," *Discrete Applied Mathematics*, vol. 40, no. 1, pp. 45-72, 1992.
- [2] R. Novak and J. Rugelj, "Distribution of constrained Steiner tree computation in point-to-point networks," *Proceedings of the Mediterranean Electrotechnical Conference on Industrial Applications in Power Systems, Computer Science and Telecommunications*, 1996, pp. 959-962.
- [3] C. P. Low and X. Song, "On finding feasible solutions for the delay constrained group multicast routing problem," *IEEE Transactions on Computers*, vol. 51, pp. 581-588, 2002.
- [4] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," *IEEE Transactions on Computers*, vol. 51, pp. 581-588, 2002.
- [5] Q. Sun and H. Langendoerfer, "Multicast routing for multimedia communication," *Proceedings of the Second Workshop on Protocols for Multimedia Systems*, 1995, pp. 452-458.
- [6] 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.
- [7] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, pp. 120-124, 1999.
- [8] 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.
- [9] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: multicast core-extraction distributed ad-hoc routing," *Proceedings of the IEEE Wireless Communications and Networking Conference*, 1999, pp. 1313-1317.
- [10] E. Pagani and G.P. Rossi, "A framework for the admission control of QoS multicast traffic in mobile ad hoc networks," *Proceedings of the ACM International Workshop on Wireless Mobile Multimedia*, 2001, pp. 3-12.
- [11] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2000, pp. 61-68.
- [12] R. Novak and J. Rugelj, "Distribution of constrained Steiner tree computation in point-to-point networks," *Proceedings of the IASTED International Conference Applied Informatics*, 1996, pp. 279-282.
- [13] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," *Proceedings of ACM SIGCOMM*, 1994, pp. 212-225.
- [14] Network Simulator (Version 2), <http://www-mash.cs.berkeley.edu/ns/>.

- [15] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.
- [16] C. C. Hu, E. H. K. Wu, and G. H. Chen, "Feasible bandwidth-satisfied multicast tree determination in MANETs," *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2005, to appear.
- [17] A. Geoffrion and R. Marsten, "Integer programming algorithms: a framework and state-of-the-art survey," *Management Science*, vol. 18, pp. 465-491, 1972.
- [18] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility for wireless ad hoc," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257-269, 2003.
- [19] Y. Yang and R. Kravets, "Content-aware admission control for ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 363-377, 2005.
- [20] L. Chen and W. Heinzelman, "Qos-aware routing based on bandwidth estimation for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 561-572, 2005.
- [21] L. Kou, G. Markowsky, and L. Bermain, "A fast algorithm for Steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141-145, 1981.
- [22] S. Ramanathan, "Multicast tree generation in networks with asymmetric links," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 558-568, 1996.
- [23] M. Smith and P. Winter, "Path distance heuristics for the Steiner problem in undirected networks," *Algorithmica*, vol. 7, no. 23, pp. 309-327, 1992.
- [24] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," *Mathematica Japonica*, vol. 24, no. 6, pp. 573-577, 1980.
- [25] F. Bauer and A. Varma, "Distributed algorithms for multicast path setup in data networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 181-191, 1996.
- [26] L. Gatani, G. L. Re, and S. Gaglio, "An efficient distributed algorithm for generating multicast distribution trees," *Proceedings of the International Conference on Parallel Processing Workshops*, 2005, pp. 477-484.
- [27] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 286-292, 1993.
- [28] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Two distributed algorithms for multicasting multimedia information," *Proceedings of the International Conference on Computer Communications and Networks*, 1993, pp. 343-349.
- [29] S. J. Lee and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 7, pp. 441-453, 2002.

## 七、計畫結果自評

HRP and HMRP, which are two bandwidth-violation problems that may occur to previous QoS routing/multicasting protocols, were introduced in this paper. The simulation showed that when the network traffic is saturated (the number of multicast groups exceeds 5), AQOR will induce HRP and MCEDAR will induce HMRP. The network performance will drop

drastically if they are induced.

Since bandwidth and power are limited in MANETs, they should be taken into consideration in routing/multicasting protocols. The problem (i.e., BSMTP) of determining a bandwidth-satisfied multicast tree with minimum number of forwarders was studied in this paper. Two algorithms that can generate bandwidth-satisfied multicast trees were proposed. They can construct bandwidth-satisfied multicast trees without inducing HRP and HMRP.

Performance comparison was made among Heu, Opt and MCEDAR, where Opt served as a benchmark. Heu has higher receiving rates than MCEDAR, and it maintains high receiving rates (between 0.859 and 1) even if the network traffic is saturated. If the network traffic is not saturated (the number of multicast groups is smaller than 5), Heu can admit more multicast groups than MCEDAR, which is a consequence that the multicast trees constructed by it with fewer forwarders. Moreover, the admission rate of Heu is close to the benchmark. The ratio of the admission rate of Heu to the admission rate of Opt ranges from 0.92 to 1.

很高興有這次的機會可以參加在布拉格舉行的 ACM MSWIM 會議，藉由這次的會議，不僅僅有技術上的許多收穫，也增廣了我做研究的多方面思考方向