A Heuristic Multicast Algorithm to Support QoS Group Communications in Heterogeneous Network

Hui Cheng, Jiannong Cao, *Senior Member, IEEE*, and Xingwei Wang

***Abstract*—This paper examines the problem of quality-of-service group communications in a heterogeneous network, which consists of multiple mobile *ad hoc* networks attached to the backbone Internet. A heuristic multicast algorithm named delay and delay variation multicast algorithm (DDVMA) is proposed. DDVMA is designed for solving the delay- and delay-variation-bounded multicast tree problem, which has been proved to be NP-complete. It can find a multicast tree satisfying the multicast end-to-end delay constraint and minimizing the multicast delay variation. Two concepts, which can help the DDVMA achieve better performance in terms of multicast delay variation than the delay and delay variation constraint algorithm that is known to be the most efficient so far, are proposed, namely, 1) the proprietary second shortest path and 2) the partially proprietary second shortest path. An analysis is given to show the correctness of DDVMA, and simulations are conducted to demonstrate the performance improvement of DDVMA in terms of multicast delay variation. It is also shown that the strategy employed by DDVMA is also applicable to handling the mobility of mobile hosts in a heterogeneous network.**

***Index Terms*—Heterogeneous network, MANET, multicast, QoS.**

I. Introduction

THE EXPLOSIVE growth of mobile communications has attracted interests in the integration of wireless networks with wireline ones and the Internet in particular. Providing mobile users wireless access to the Internet is of major interest in today's research in networking. In addition to wireless Internet [1], this also includes extending Mobile *Ad hoc* NETworks (MANETs) [2] with IP connectivity of the mobile hosts (MHs) to the Internet. An integrated connectivity solution is proposed in [3]. Its prototype is implemented by connecting IP networks and MANETs running the *ad hoc* on-demand distance vector (AODV) routing protocol [4], where a mobile IP [5] is used for mobility management. Mobile IP MANET (MIPMANET) [6] is a solution for connecting a MANET to the Internet. MIPMANET uses on-demand routing and provides Internet access by using mobile IP with foreign agent care of addresses and reverse tunneling. A heterogeneous network architecture is proposed in [7], which extends the typical wireless access points to multiple MANETs, each as a subnet of the Internet, to create an integrated environment that supports both macro-IP and micro-IP mobility. The heterogeneous network architecture will facilitate the current trend of moving to an all-IP wireless environment.

In a heterogeneous network consisting of multiple MANETs attached to the backbone Internet, a gateway is a fixed node connecting a MANET to the Internet and each gateway serves one MANET. Gateways forward data packets and relay them between MANETs and the Internet. When a MANET is connected to the Internet, it is important for the MHs to detect available Internet gateways. Therefore, an efficient gateway discovery mechanism is required. Many efforts have been devoted to the problems of gateway forwarding strategies and Internet gateway discovery [8]–[10]. These works have provided the foundation for our work.

Such an integrated heterogeneous network environment has brought up many new applications. In particular, there is an increasing demand for enhanced services to help users do mobile collaborations, which require the support for mobile group communications. For example, several MANETs, which are distributed in different remote regions, need to coordinate their works through the backbone Internet. However, to the best of our knowledge, no multicast algorithm has been proposed to support the quality-of-service (QoS) group communication [11] in backbone wireline networks attached by MANETs.

There are two important QoS parameters. The first is the end-to-end delay [12] that is used to ensure that the messages transmitted by the source can reach the destination within a certain amount of time. The second is the multicast delay variation [13], defined as the difference between the maximum and the minimum multicast end-to-end delays on the multicast tree. It measures the consistency and fairness of receiving messages among all the destinations.

In this paper, we propose a heuristic multicast algorithm named delay and delay variation multicast algorithm (DDVMA) for QoS group communication in a heterogeneous network. In DDVMA, each MANET can be seen as a team. When one team wants to send messages to multiple remote teams, two steps are needed. First, the AODV routing protocol

Manuscript received November 15, 2005; revised January 21, 2006. This work was supported in part by the University Grant Council of Hong Kong under the CERG Grant PolyU 5170/03E and by the National Natural Science Foundation of China under Grant No. 60473089. The review of this paper was coordinated by Prof. X. (Sherman) Shen.

H. Cheng and J. Cao are with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: cshcheng@comp.polyu.edu.hk; csjcao@comp.polyu.edu.hk).

X. Wang is with the College of Information Science and Engineering, Northeastern University, Shenyang 110004, China (e-mail: wangxw@mail.neu.edu.cn).

Digital Object Identifier 10.1109/TVT.2006.873835

is used to discover routes between the team leader and its gateway. The end-to-end delay values of all the wireless routes are collected for use by the DDVMA. Second, DDVMA constructs a multicast tree from the source gateway to all the destination gateways in the backbone network utilizing both the topology of backbone network and the delay values of wireless routes.

The advantages of DDVMA are as follows. 1) By using AODV to discover wireless routes and collect delay information, the construction of the multicast tree in the backbone network can guarantee the QoS requirements of the group communication involving MHs. 2) Under the multicast end-to-end delay constraint, the proposed DDVMA can achieve better performance in terms of multicast delay variation than the delay and delay variation constraint algorithm (DDVCA) [13] known to be the most efficient so far.

The rest of this paper is organized as follows. Section II describes the network model, the problem specification, and related work, and introduces several new concepts. DDVMA is proposed in Section III. Proof of correctness and performance evaluation are described in Sections IV and V, respectively. The strategy to handle the mobility of MHs is proposed in Section VI. Finally, we conclude this paper in Section VII.

II. Preliminaries

A. Network Model and Problem Specification

The backbone wireline network can be modeled as a weighed digraph $G(V,E)$, where V represents the set of nodes including gateways and E represents the set of links between the nodes. For each link $l \in E$, a link delay function $D: l \rightarrow r^+$ is defined. A nonnegative value $D(l)$ represents the transmission delay on link l.

Multicast messages are sent from the leader MH of the source MANET. Messages are first forwarded to the source gateway $v_s \in V$ through the route discovered by AODV, then arrive at a set of destination gateways $M \subseteq V - \{v_s\}$ through the multicast tree T constructed over the backbone network, and finally are forwarded to the leader MHs of the destination MANETs through the wireless routes between each destination gateway and each leader MH. To guarantee the QoS of group communication, the multicast end-to-end delay between the leader MH of the source MANET and the leader MH of each destination MANET should not exceed the multicast end-to-end delay constraint Δ, and the multicast delay variation among the leader MHs of destination MANETs should be minimized.

Let $P_T(v_s, v_w)$ denote the path from the source gateway v_s to a destination gateway $v_w \in M$ on T. Then, the transmission delay between v_s and v_w on T is defined as $\sum_{l \in P_T(v_s,v_w)} D(l)$. We define a gateway delay function $W: g \rightarrow r^+$ for each gateway $g \in \{v_s\} \cup M$. It assigns gateway g a nonnegative value $W(g)$, which represents the delay of the wireless route discovered between gateway g and the leader MH of the MANET g serves.

In this paper, we solve the problem of QoS group communications in the heterogeneous network by finding an optimal multicast tree $T^*(V_{T^*}, E_{T^*})$, $\{v_s\} \cup M \subseteq V_{T^*}$, $E_{T^*} \subseteq E$, satisfying

$$\Delta_{T^*} = W(v_s) + \max_{v_w \in M} \left\{ \sum_{l \in P_{T^*}(v_s,v_w)} D(l) + W(v_w) \right\} \leq \Delta$$

$$\delta_{T^*} = \min_{T} \left\{ \min_{v_u, v_w \in M} \left\{ \left| \left(\sum_{l \in P_T(v_s,v_u)} D(l) + W(v_u) \right) - \left(\sum_{l \in P_T(v_s,v_w)} D(l) + W(v_w) \right) \right| \right\} \right\}$$

where T denotes any multicast tree spanning v_s and M in $G(V,E)$.

If we assume $W(g) = 0$ for each $g \in \{v_s\} \cup M$, the problem turns to be the delay- and delay-variation-bounded multicast tree (DVBMT) problem [14], which has been proved to be NP-complete. Our problem is also NP-complete because it contains, as a special case, the DVBMT problem. Hence, only heuristic algorithms can be developed for it.

B. Related Work

For the DVBMT problem, several heuristic algorithms have been proposed. The delay variation multicast algorithm (DVMA) [14] is a search algorithm that attempts to construct a multicast tree satisfying both the multicast end-to-end delay constraint and the multicast delay variation constraint. Although DVMA demonstrates good average case behavior in terms of the multicast delay variation, its time complexity is very high. DDVCA [13] is a fast and efficient algorithm, which is meant to search as much as possible for a multicast tree with a small multicast delay variation under the multicast end-to-end delay constraint. DDVCA claims to outperform DVMA slightly in the multicast delay variation. However, in contrast to DVMA, the time complexity of DDVCA is lower.

In DDVCA, the minimum delay path algorithm and the shortest path tree (SPT) are used. SPT is constructed by combining all the shortest (i.e., minimum delay) paths from the source node to each destination node. The fundamental strategy of DDVCA comes from the core-based tree (CBT)'s core router concept and the minimum delay path algorithm. The basic idea is described as follows. In DDVCA, for each network node, the SPT from it to all the destination nodes is constructed. The node whose SPT has the minimum multicast delay variation is selected as the central node. Then, a checking process is done to examine whether the sum of the minimum delay between the source node and the current central node and the maximum multicast end-to-end delay of the SPT rooted at the central node satisfies the multicast end-to-end delay constraint. If the central node violates the constraint, it will be abandoned. In this case, the algorithm will go on to pick the node whose SPT has the next minimum multicast delay variation as the next possible central node and apply the same checking process until a central node that satisfies the constraint is found.

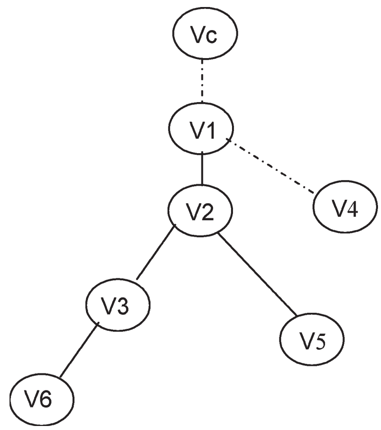


Fig. 1. Illustration of proprietary links and partially proprietary links.

C. Notations and Definitions

SPT has very good performance in terms of multicast end-to-end delay. But selecting the shortest paths may lead to a violation of the delay variation constraint among nodes that are close to the source and nodes that are far away from it. Consequently, it may be necessary to select longer paths for some destination nodes to further reduce the multicast delay variation of the SPT. Therefore, intuitively, if we introduce higher delay paths to replace the minimum delay paths from the source to some destinations on the SPT, more trees with small multicast delay variations can be searched compared to DDVCA.

We define the concepts of proprietary second shortest path and partially proprietary second shortest path, which will be used as the higher delay path.

We denote the central node being checked as v_c. Let $T(v_c)$ represent the SPT rooted at v_c. For one destination node v_j, we define the following:

Proprietary Links: links that are not shared by other destination nodes on $T(v_c)$.

Proprietary Link Set (PS): all the proprietary links of v_j.

In Fig. 1, suppose that Vc is the central node and V2, V4, V5, and V6 are the destination nodes. For V6, its proprietary links are (V2, V3) and (V3, V6). So its proprietary link set is {(V2, V3), (V3, V6)}.

Proprietary Second Shortest Path: the second shortest path from v_c to v_j, which is obtained by computing the shortest path from v_c to v_j after deleting l from the network topology G, $l \in$ PS. So the number of proprietary second shortest paths equals the number of proprietary links for v_j. The proprietary second shortest path is actually the shortest path on the network topology $G - \{l\}$.

Partially Proprietary Links: links that are only shared by all its child destination nodes on $T(v_c)$.

Partially Proprietary Link Set (PPS): all the partially proprietary links of v_j.

In Fig. 1, for V2, its partial proprietary link is (V1, V2). So its partially proprietary link set is {(V1, V2)}.

Partially Proprietary Second Shortest Path: the second shortest path from v_c to either v_j or a child destination node of v_j, which is obtained by computing the shortest path from v_c to either v_j or the child destination node of v_j after deleting l from the network topology G, $l \in$ PPS. The partially proprietary second shortest path is actually the shortest path on the network topology $G - \{l\}$.

The characteristics of proprietary second shortest paths and partially proprietary second shortest paths guarantee that adding them to the SPT will not create a cycle, which is proved by Theorem 1 in Section IV. Thus, other multicast end-to-end paths on the SPT will not be interfered with. Similarly, it can be proved that using a partially proprietary second shortest path to replace the shortest path will not create a cycle on $T(v_c)$.

For a multicast tree, it is easy to determine the proprietary link set or partially proprietary link set for a destination node; hence, we can compute the proprietary second shortest paths or partially proprietary second shortest paths for a destination node using Dijkstra's algorithm conveniently and quickly.

III. DDVMA: A Heuristic Multicast Algorithm

A. Overview of DDVMA

DDVMA constructs a QoS multicast tree over the backbone network to transmit multicast messages from the source gateway to all the destination gateways. An optimal wireless route between each leader MH and its gateway is discovered by the AODV routing protocol. The delay values of the wireless routes are collected for computation in DDVMA.

Compared with the DDVCA, the improvement of DDVMA is realized by using the proprietary second shortest path or partially proprietary second shortest path to replace the multicast path with the minimum end-to-end delay on the SPT. The improvement procedure can be seen as an optimization procedure, i.e., using a better path to optimize the QoS of the SPT. The optimization objective is to achieve a smaller multicast delay variation under a multicast end-to-end delay constraint.

The optimization procedure will stop when one of the following two cases occurs:

1) multicast delay variation has been decreased to a specified tolerance range or cannot be decreased further;
2) maximum multicast end-to-end delay of the SPT exceeds the given upper bound.

During the optimization procedure, the tree should always keep an SPT structure for the associated network topology. At the beginning, the associated network topology is just the network topology G. After each replacement, the selected proprietary link or partially proprietary link will be excluded from the associated network topology.

For a destination node on the SPT, if its proprietary link set is not NULL, its partially proprietary link set will be NULL, and vice versa. Assume that we are checking the destination node with minimum multicast end-to-end delay on the SPT. If its proprietary link set is not NULL, which means it is a leaf node, we will check whether a proprietary second shortest path can be found for it to optimize the tree; if its partially proprietary link set is not NULL, which means it is a nonleaf node, we will check whether partially proprietary second shortest paths can be found for it to optimize the tree.

***Algorithm* DDVMA:**

begin

For each network node $v \in V$ do
Construct the SPT $T(v)$ from v to all the destinations.
If $T(v)$ satisfies the multicast end-to-end delay constraint, optimize $T(v)$ using procedure P or procedure PP until the multicast delay variation cannot be improved under the multicast end-to-end delay constraint.
end of for v loop
Choose the node with the smallest value of multicast delay variation under the multicast end-to-end delay constraint as the central node.
Construct the multicast tree by connecting the central node with both the source and all destinations.

end of the algorithm

Fig. 2. Formal description of DDVMA.

B. Formal Description of DDVMA

In this section, we will present a formal description of DDVMA as shown in Fig. 2. Two procedures are used: one is to deal with the destination node with at least one proprietary link and the other is to deal with the destination node with at least one partially proprietary link. The former is named as procedure P (Proprietary), the latter is named as procedure PP (Partially Proprietary). Both are described in Section III-C.

As mentioned before, the wireless routing delay between each gateway and the leader MH is used to compute the multicast end-to-end delay and the multicast delay variation in DDVMA. For a path that ends at a destination gateway, the wireless routing delay collected by the destination gateway is added to the path delay. Thus, it is guaranteed that the constructed multicast tree satisfies the QoS requirement of the multicast message transmission among leader MHs in multiple MANETs attached to the backbone Internet.

C. Procedure P and PP

Procedure P starts out with an SPT and decreases the multicast delay variation by replacing the minimum delay multicast path with the appropriate proprietary second shortest path. If procedure P returns false, it means that the SPT remains unchanged.

After $T(v_c)$ is modified, the network graph G' associated with it needs to exclude the selected proprietary link l (i.e., $G' = G' - \{l\}$) to keep the SPT structure of $T(v_c)$. On the updated network topology G', the proprietary second shortest path keeps to be the shortest path and the improved SPT keeps to be the SPT. The associated network topology will be updated each time the SPT is modified by procedure P.

Different from procedure P, procedure PP uses the partially proprietary second shortest paths. If the nonleaf node v_j is the destination node with the minimum multicast end-to-end delay on $T(v_c)$, some child nodes of v_j will also be destination nodes. Node v_j with all its child destination nodes forms a subset M' of M. $P(v_c, v_j)$ represents the multicast path from v_c to v_j on $T(v_c)$, and $T(v_j)$ represents the sub-SPT rooted at node v_j on $T(v_c)$. $P(v_c, v_j)$ is the common part of each multicast path $P(v_c, j)$, $j \in M' - \{v_j\}$. For each $j \in M' - \{v_j\}$, $P(v_c, j)$ will also be changed when $P(v_c, v_j)$ is replaced by the corresponding partially proprietary second shortest path $P'(v_c, v_j)$.

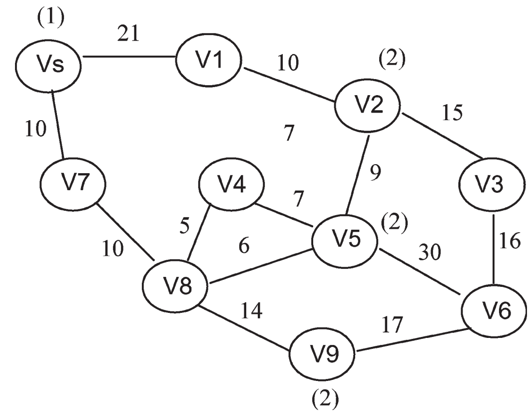


Fig. 3. Given network topology $G = (V, E)$.

There are two strategies to handle the changes of $P(v_c, j)(j \in M' - \{v_j\})$ caused by the change of $P(v_c, v_j)$ in procedure PP: 1) compute the partially proprietary second shortest path $P'(v_c, j)$ as the new multicast path from v_c to j for each $j \in M' - \{v_j\}$; 2) use the corresponding path on $P'(v_c, v_j) \cup T(v_j)$ [i.e., the topology combining the partially proprietary second shortest path $P'(v_c, v_j)$ with the sub-SPT $T(v_j)$] as the new multicast path from v_c to j for each $j \in M' - \{v_j\}$. Procedure PP adopts strategy (1) because it can help improve the multicast delay variation between v_j and any other node in M'. We prove it by Theorem 2 in Section IV.

If procedure PP returns false, it means that the SPT remains unchanged. Similar to procedure P, the associated network topology will be updated each time the SPT is modified in procedure PP.

D. Illustrative Example of DDVMA

In the following, we will illustrate the operation of DDVMA with an example. We will contrast it with DDVCA, so we use the computer network topology given in [13]. The network topology is shown in Fig. 3. For a group communication scenario, we denote Vs as the source gateway, and V2, V5, and V9 as the destination gateways, i.e., $M = \{\mathrm{V2}, \mathrm{V5}, \mathrm{V9}\}$. The number in the parentheses near gateway g (including the source gateway and all the destination gateways) represents the corresponding wireless route delay $W(g)$. Suppose the multicast end-to-end delay constraint is 60. Because the wireless route delay between the source leader MH and the source gateway is 1, the multicast end-to-end delay constraint used in DDVMA will be 59 (i.e., $60 - 1$). Table I shows the procedure of selecting a central node in DDVCA. Table II shows the corresponding procedure in DDVMA.

In Table I, for each network node Vi, the minimum path delay between it and each destination gateway (i.e., the wireline

TABLE I
SELECTING A CENTRAL NODE IN DDVCA

Network node Vi The minimum delay	Destination node Vj	Vs	V1	V2	V3	V4	V5	V6	V7	V8	V9
The multicast path delay between Vi and Vj	V2	33	12	2	17	9	11	33	24	14	28
	V5	28	21	11	26	9	2	32	18	8	22
	V9	36	38	28	35	21	22	19	26	16	2
The maximum delay from Vi to each Vj		36	38	28	35	21	22	33	26	16	28
The minimum delay from Vi to each Vj		28	12	2	17	9	2	19	18	8	2
dv(Vi)		8	26	26	18	12	20	14	8	8	26

TABLE II
SELECTING A CENTRAL NODE IN DDVMA

Network node Vi The minimum delay	Destination node Vj	Vs	V1	V2	V3	V4	V5	V6	V7	V8	V9
The multicast path delay between Vi and Vj	V2	33	12	2	17	22*	11	33	24	14	28
	V5	34*	21	11	26	13*	2	32	24*	14*	22
	V9	36	38	28	35	21	22	19	26	16	2
The maximum delay from Vi to each Vj		36	38	28	35	22	22	33	26	16	28
The minimum delay from Vi to each Vj		33	12	2	17	13	2	19	24	14	2
dv(Vi)		3	26	26	18	9	20	14	2	2	26

transmission path delay in the backbone network plus the corresponding wireless route delay recorded at the destination gateway) is listed in each column. The multicast delay variation of the SPT rooted at Vi, *dv*(Vi), is listed at the bottom line of each column. From Table I, we know that the multicast delay variations of the SPT rooted at Vs, V7, and V8 are the minimum. Assuming that Vs is selected, we obtain the multicast tree that satisfies the multicast end-to-end delay constraint and achieves the multicast delay variation 8.

In Table II, for each network node Vi, in each column we also list the path delay between it and each destination gateway on the improved SPT rooted at Vi. An * next to a delay value indicates that it is the delay of the proprietary second shortest path or partially proprietary second shortest path. It means that the corresponding minimum delay paths on the SPT have been replaced by the proprietary or partially proprietary second shortest path. The multicast delay variation of the improved SPT, $dv(\mathrm{Vi})$, is listed at the bottom line of each column. From Table II, we know that the multicast delay variations of the improved SPT rooted at V7 and V8 are both the minimum. Assuming that V7 is selected, we get the multicast tree that satisfies the multicast end-to-end delay constraint and achieves the multicast delay variation 2.

The example shows that DDVMA can achieve the multicast tree with smaller multicast delay variation than DDVCA.

IV. PROOF OF THE ALGORITHM PROPERTIES

In this section, we prove the algorithm properties claimed in the previous sections.

Theorem 1: Let $T(v_c)$ be the SPT rooted at v_c. For any destination node $v_j \in M$ with $\mathrm{PS}(v_j) \neq \varnothing$, using a proprietary second shortest path to replace the shortest path will not create a cycle on $T(v_c)$.

Proof: The proof is done by contradiction. Let $T^*(v_c)$ represent the SPT from v_c to all nodes in $M - \{v_j\}$. Clearly, $T^*(v_c)$ is one part of $T(v_c)$. According to the definition of proprietary links, all links in $\mathrm{PS}(v_j)$ are not shared by any destination node in $M - \{v_j\}$ on $T(v_c)$. So no link in $\mathrm{PS}(v_j)$ belongs to $T^*(v_c)$. The proprietary second shortest path is obtained by computing the shortest path after deleting the selected proprietary link from the associated network topology. All the shortest paths on $T^*(v_c)$ will remain unchanged for the updated network topology. Hence, $T^*(v_c)$ is still one part of the improved SPT. The improved SPT is constructed by combining the proprietary second shortest path with $T^*(v_c)$. We assume that the replacement creates a cycle. Thus, the proprietary second shortest path must contain at least one node belonging to $T^*(v_c)$ except v_c. But for the proprietary second shortest path, the subpath from v_c to any node belonging to $T^*(v_c)$ is still the shortest path that coincides with the original path on $T^*(v_c)$. It contradicts with the assumption of a cycle being created. Hence, the replacement of a proprietary second shortest path will not create a cycle. ■

Theorem 2: Let v_j be a nonleaf destination node with the minimum multicast end-to-end delay on $T(v_c)$ (the SPT rooted at v_c) and $\mathrm{PPS}(v_j) \neq \varnothing$. Let $j_1, j_2, \ldots, \Lambda, j_t$ represent the child destination nodes of v_j on $T(v_c)$ and $M' = \{v_j, j_1, j_2, \ldots, \Lambda, j_t\}$. By using strategy (1), which is mentioned in Section III-C, the procedure PP can improve the multicast delay variation between v_j and any other node in M'.

Proof: By adopting strategy (1), procedure PP computes the partially proprietary second shortest path $P'(v_c, j)$ for each $j \in M'$ and use them to replace the original shortest paths on $T(v_c)$.

Let $P'(v_c, v_j)$ represent the partially proprietary second shortest path from v_c to v_j. For each $j \in M' - \{v_j\}$, let $P(v_j, j)$ represent the shortest path between v_j and j, and

$P'(v_c, j)$ represent the partially proprietary second shortest path between v_c and j. $P'(v_c, v_j)$, $P'(v_c, j)$, and $P(v_j, j)$ are all the shortest paths for the updated network topology (i.e., deleting the selected partially proprietary link from the network topology associated with the previous SPT). So we have

$$\text{Delay}\,(P'(v_c, v_j)) \leqslant \text{Delay}\,(P'(v_c, j)) + \text{Delay}\,(P(v_j, j)) \quad (1)$$

$$\text{Delay}\,(P'(v_c, v_j)) + \text{Delay}\,(P(v_j, j)) \geqslant \text{Delay}\,(P'(v_c, j)). \quad (2)$$

From the two expressions, we get

$$|\text{Delay}\,(P'(v_c, j)) - \text{Delay}\,(P'(v_c, v_j))| \leqslant \text{Delay}\,(P(v_j, j)). \quad (3)$$

Because

$$\text{Delay}\,(P(v_j, j)) \leqslant \delta \quad (4)$$

where δ is the multicast delay variation of $T(v_c)$, we get

$$|\text{Delay}\,(P'(v_c, j)) - \text{Delay}\,(P'(v_c, v_j))| \leqslant \delta. \quad (5)$$

We can see that by using strategy (1), procedure PP can help improve the multicast delay variation between v_j and each $j \in M' - \{v_j\}$. ■

V. Performance Evaluation

A. Algorithm Analysis

We will show by argument that DDVMA can achieve higher efficiency in terms of the multicast delay variation than DDVCA.

In DDVMA, since each network node is checked, the source gateway v_s is also likely to be selected as the central node. Then the multicast tree is constructed by connecting v_s to each destination gateway through the minimum delay path. Such a multicast tree is the SPT from v_s to all destination gateways. If it does not satisfy the multicast end-to-end delay constraint, clearly there does not exist any multicast tree that can satisfy the multicast end-to-end delay constraint regulated by the input. This characteristic has also been stated in [13].

For each network node being checked, when the SPT is constructed, the multicast delay variation of the SPT is used in DDVCA. But in DDVMA, we will execute procedure P and procedure PP to further reduce the multicast delay variation of the SPT. Procedure P and procedure PP can keep the SPT unchanged or return an improved SPT with smaller multicast delay variation. They are called repeatedly until the SPT cannot be improved. So DDVMA can search more possible multicast trees and achieve higher efficiency in terms of the multicast delay variation than DDVCA. The illustration in Section III-D shows this characteristic.

The time complexity of DDVCA is $O(mn^2)$, where m is the number of destination nodes and n is the number of network nodes. Since the time complexity of procedure P and procedure PP is $O(n^2)$, the same as Dijkstra's algorithm, the time complexity of DDVMA remains the same as DDVCA.

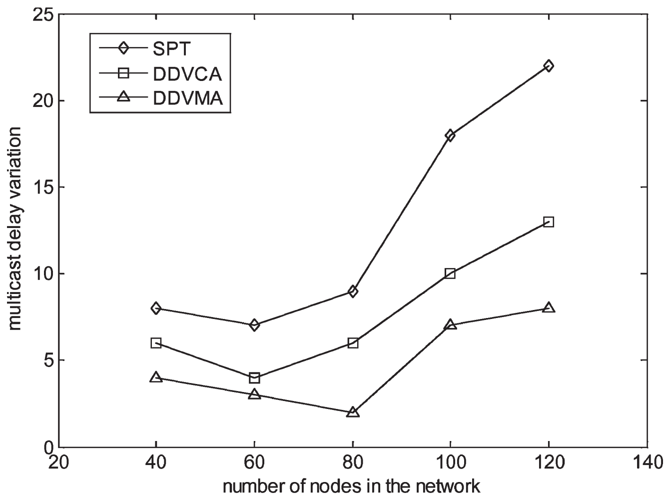
B. Simulation

Simulation experiments were conducted to examine the efficiency of DDVMA. Given two integers n and $m(n-1 \leq m \leq n(n-1)/2)$, an interval [LD, UD], and an integer d, our random graph generator will generate a connected network topology graph with n nodes and m links. The delay on each link is an integer value in [LD, UD], which is in direct proportion to the length of the link. The degree of each node does not exceed d. The random graph generator first generates the n nodes. It then picks out two different nodes randomly. For the two nodes, if no direct link connects them and both of their node degrees are less than d, a new link between them will be added to the graph. This process is continued until m links are added to the graph. A similar random graph generation approach is introduced in [15].

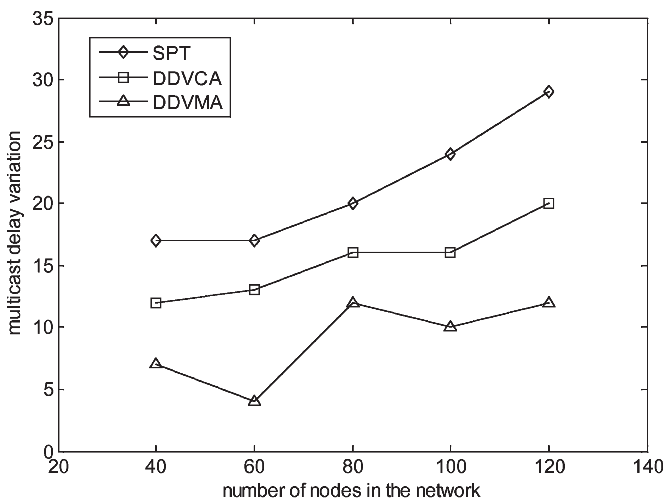
In our simulation experiments, we generate five different network topology graphs. Their sizes range from 40 to 60, 80, and up to 120 nodes. The delay on each link is drawn from the interval [1, 10]. For a specified multicast group, the upper bound on the maximum multicast end-to-end delay Δ is set to be 1.5 times the minimum delay between the source node and the farthest destination node. In the simulation, we compared DDVMA with DDVCA and the SPT algorithm produced from Dijkstra's algorithm. We evaluated the multicast delay variations and multicast end-to-end delays of the three algorithms. For each network, we investigated two cases: one is that the destination nodes in the multicast group occupy 5% of the total nodes in the network and the other is 20%. For each case, we generate 20 different multicast groups randomly. Then, 20 multicast trees are obtained by each algorithm. We calculated the average over the multicast delay variations of the 20 multicast trees for each algorithm and used the average value to evaluate the efficiency of the algorithm in terms of multicast delay variation.

Fig. 4 shows the simulation results of multicast delay variations. Fig. 4(a) corresponds to the multicast groups of sizes equal to 5% of the number of network nodes. It can be regarded as the scenario that multicast nodes are distributed sparsely over the network. Fig. 4(b) corresponds to multicast groups of sizes equal to 20% of the number of network nodes. It represents the scenario that multicast nodes are distributed densely over the network. We observe that the trees constructed by DDVMA have an average multicast delay variation that is always smaller than that of SPT and DDVCA trees. With the ratio of the multicast group size to the number of network nodes increasing from 5% to 20%, it is apparent that the multicast delay variation of DDVMA performs much better than that of DDVCA. The performance of the SPT algorithm is the worst in terms of the multicast delay variation among the three algorithms.

We also calculated the average over the maximum multicast end-to-end delays of the obtained multicast trees for each algorithm. Fig. 5 shows the simulation results on the multicast end-to-end delays of different algorithms. It corresponds to the case where the destination nodes in a multicast group occupy



(a)



(b)

Fig. 4. Comparison on the multicast delay variations of the three different algorithms. (a) Multicast group sizes equal to 5% of the number of network nodes. (b) Multicast group sizes equal to 20% of the number of network nodes.

5% of the total network nodes. The simulation result of the 20% case is similar, so we only present and discuss the 5% case. We observe that the multicast end-to-end delay of DDVCA performs better than that of DDVMA, but not much. It can be explained by the design of DDVMA. In DDVMA, we improve the multicast delay variations of the SPTs by introducing higher delay paths. If the delay of the accepted new path exceeds the maximum multicast end-to-end delay of the SPT, the maximum multicast end-to-end delay of the multicast tree will be increased. But if the delay of the new path is so high that the multicast delay variation of the SPT is increased, the path will not be accepted. So in average, we can see that DDVMA and DDVCA have competing performances on multicast end-to-end delays. The SPT algorithm inherently has the best performance in terms of the multicast end-to-end delay.

VI. Handling Mobility

For mobile communications, the handling mobility of the MHs is an important issue. For QoS mobile group communications in a heterogeneous network, the multicast tree needs to support host mobility by reconstructing the multicast path to the MH's new location adaptively. We confine the movement of an MH within its local MANET. If one MH moves to a new location, the MANET routing protocol will discover the new wireless route between the MH and its gateway. The delay of the new wireless route will be collected again. The delay of the new wireless route may decrease or increase. This will lead to the decrease or increase of the multicast end-to-end delay between the source MH and the destination MH because the delay of the new wireless route is one part of the multicast end-to-end delay. If the decrease or increase of the multicast end-to-end delay does not make the multicast delay variation of the mobile multicast tree intolerable, the multicast tree in the backbone network can still be used; otherwise, it needs to be reconstructed locally.

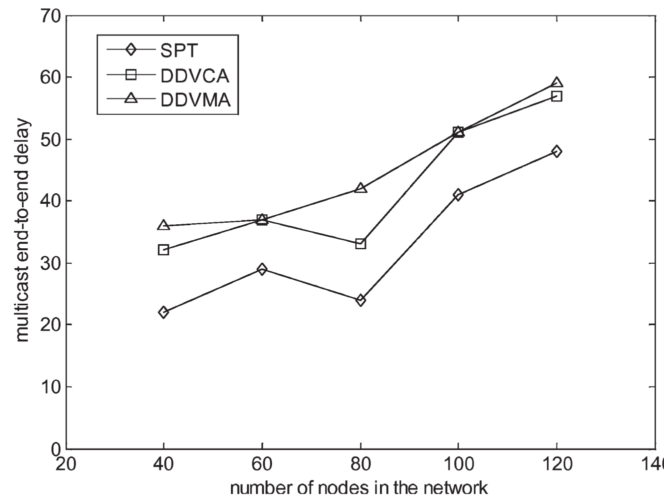


Fig. 5. Comparison on the multicast end-to-end delays of the three algorithms.

As we know, the multicast tree obtained by DDVMA is the combination of the shortest path between the source gateway and a central node and a tree from the central node to all the destination gateways. The tree is an SPT based on the associated network topology. To handle mobility, the associated network topology needs to be recorded after the multicast tree is determined in the backbone network. The following operations can be conducted based on the associated network topology.

For the case that the movement of a destination MH leads to the decrease of the multicast end-to-end delay, we first compute the proprietary second shortest paths or partially proprietary second shortest paths between the central node and the corresponding destination gateway. Then, we select the one whose replacement on the SPT will mostly improve the multicast delay variation under the multicast end-to-end delay constraint and use it as the new multicast path.

For the case that the movement of a destination MH leads to the increase of the multicast end-to-end delay, the increased multicast end-to-end delay will become the maximum multicast end-to-end delay of the multicast tree. Then, for the destination gateway with the minimum multicast end-to-end delay on the SPT, we compute the proprietary second shortest paths or partially proprietary second shortest paths for it. We also select the one whose replacement will mostly improve the multicast delay variation under the multicast end-to-end delay constraint to be the new multicast path. For the updated SPT, repeat this process until the multicast delay variation cannot be improved. Finally, use the new SPT to replace the old SPT on the multicast tree.

VII. Conclusion

In this paper, a heuristic multicast algorithm called DDVMA is developed for constructing the multicast tree spanning the source gateway and all the destination gateways in the heterogeneous network. DDVMA uses higher delay paths to replace the corresponding shortest paths on the SPT for further reducing the multicast delay variation of the SPT rooted at the stand-by central node.

We use the AODV routing protocol to discover the routes between the leader MH and its gateway. Each gateway collects and updates the delay information of the wireless route. Combined with wireless routes between each leader MH and its gateway, the QoS multicast tree obtained by DDVMA can support communications among leader MHs in multiple MANETs attached to the backbone Internet. Furthermore, the multicast tree can satisfy the multicast end-to-end delay constraint and achieve smaller multicast delay variations than the multicast tree obtained by DDVCA known to be the best algorithm for the DVBMT problem. DDVMA can be implemented in an IP routing protocol. This makes our solution simple and feasible to QoS group communications in a heterogeneous network.

For future work, in a MANET, we intend to allow multiple MHs to participate in the same group communication gateway forwarding in a heterogeneous network. Thus, each gateway will collect multiple wireless delay values, and a new multicast tree construction algorithm needs to be investigated to guarantee the QoS.

References

[1] R. S. Chang, W. Y. Chen, and Y. F. Wen, "Hybrid wireless network protocols," *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 1099–1109, Jul. 2003.

[2] X. Z. Cheng, X. Huang, and D.-Z. Du, Eds., *Ad hoc Wireless Networking*. Norwell, MA: Kluwer, 2004.

[3] C. Åhlund and A. Zaslavsky, "Extending global IP connectivity for *ad hoc* networks," *Telecommun. Syst.*, vol. 24, no. 2–4, pp. 221–250, Oct.–Dec. 2003.

[4] C. E. Perkins and E. M. Royer, "*Ad hoc* on-demand distance vector routing," in *Proc. IEEE WMCSA*, Feb. 1999, pp. 90–100.

[5] C. E. Perkins, "Mobile IP," *IEEE Commun. Mag.*, vol. 35, no. 5, pp. 84–99, May 1997.

[6] U. Jönsson, F. Alriksson, T. Larsson, P. Johansson, and G. Q. Maguire, Jr., "MIPMANET—Mobile IP for mobile *ad hoc* networks," in *Proc. IEEE MobiHoc*, 2000, pp. 75–85.

[7] Y.-C. Tseng, C.-C. Shen, and W.-T. Chen, "Integrating mobile IP with *ad hoc* networks," *IEEE Comput.*, vol. 36, no. 5, pp. 48–55, May 2003.

[8] M. Ghassemian, P. Hofmann, C. Prehofer, V. Friderikos, and H. Aghvami, "Performance analysis of Internet gateway discovery protocols in *ad hoc* networks," in *Proc. IEEE WCNC*, 2004, pp. 120–125.

[9] P. M. Ruiz and A. F. Gómez-Skarmeta, "Enhanced Internet connectivity for hybrid *ad hoc* networks through adaptive gateway discovery," in *Proc. IEEE LCN*, 2004, pp. 370–377.

[10] E. Nordström, P. Gunningberg, and C. Tschudin, "Comparison of forwarding strategies in Internet connected MANETs," *ACM Mobile Comput. Commun. Rev.*, vol. 8, no. 4, pp. 72–76, Oct. 2004.

[11] P. Mohapatra, C. Gui, and J. Li, "Group communications in mobile *ad hoc* networks," *IEEE Comput.*, vol. 37, no. 2, pp. 52–59, Feb. 2004.

[12] S. I. Maniatis, E. G. Nikolouzou, and I. S. Venieris, "End-to-end QoS specification issues in the converged all-IP wired and wireless environment," *IEEE Commun. Mag.*, vol. 42, no. 6, pp. 80–86, Jun. 2004.

[13] P.-R. Sheu and S.-T. Chen, "A fast and efficient heuristic algorithm for the delay- and delay variation-bounded multicast tree problem," *Comput. Commun.*, vol. 25, no. 8, pp. 825–833, May 2002.

[14] G. N. Rouskas and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 3, pp. 346–356, Apr. 1997.

[15] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

Hui Cheng received the B.Sc. and M.Sc. degrees in computer science from the Northeastern University, Shenyang, China, in 2001 and 2004, respectively, and is currently working toward the Ph.D. degree at the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong.

His research interests include multicast routing and QoS mobile group communication.

Jiannong Cao (M'93–SM'04) received the B.Sc. degree in computer science from Nanjing University, Nanjing, China, in 1982 and the M.Sc. and Ph.D. degrees in computer science from Washington State University, Pullman, in 1986 and 1990, respectively.

He was on the faculty of computer science at the James Cook University, the University of Adelaide, Australia, and the City University of Hong Kong. He is currently a Professor at the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong. He is also the Director of the Internet and Mobile Computing Lab at the Department of Computing. He has published over 160 technical papers. His research interests include parallel and distributed computing, networking, mobile computing, fault tolerance, and distributed software architecture and tools.

Prof. Cao is a member of the IEEE Computer Society, the IEEE Communication Society, and the Association for Computing Machinery. He is also a member of the IEEE Technical Committee on Distributed Processing, IEEE Technical Committee on Parallel Processing, IEEE Technical Committee on Fault Tolerant Computing, and Technical Committee on Computer Architecture of the China Computer Federation. He has served as a member of the editorial boards of several international journals, a Reviewer for international journals and conference proceedings, and as an organizing/program committee member for many international conferences.

Xingwei Wang received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China, in 1989, 1992, and 1998, respectively.

He is currently a Professor at the College of Information Science and Engineering, Northeastern University. He has published over 100 technical papers. His research interests include multimedia group communication, NGI, IP/DWDM optical Internet, and mobile wireless Internet.