Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT•

Computer Communications 27 (2004) 27–40

computer communications

www.elsevier.com/locate/comcom

An on-demand, link-state, multi-path QoS routing in a wireless mobile ad-hoc network[☆]

Yuh-Shyan Chen[a,*], Yu-Chee Tseng[b], Jang-Ping Sheu[c], Po-Hsuen Kuo[c]

[a]*Department of Computer Science and Information Engineering, National Chung Cheng University, 160, San-Hsing, Min-Hsiung, Chiayi 621, Taiwan, ROC*
[b]*Department of Computer Science and Information Engineering, National Chao-Tung University, Hsin-Chu, Taiwan, ROC*
[c]*Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan, ROC*

Received 22 February 2002; revised 12 May 2003; accepted 14 May 2003

Abstract

The peer-to-peer multimedia applications have recently generated much interest in wireless network infrastructure with supporting the quality-of-service (QoS) communications. In this paper, we propose a new on-demand QoS routing, namely link-state multi-path QoS routing, protocol in a wireless mobile ad-hoc network (MANET). Our proposed scheme offers a bandwidth routing protocol for QoS support in a multihop mobile network. Our QoS routing protocol determines the end-to-end bandwidth calculation and bandwidth allocation at the destination. Our protocol collects link bandwidth information from source to destination in order to construct a network topology with the information of link bandwidth at the destination. To satisfy a given bandwidth requirement, the bandwidth calculation of the QoS route are determined at the destination to accurately determine the QoS uni-path or multi-path route depending on the status of network bandwidth. Our routing scheme may offer a multi-path route if a MANET with the insufficient network bandwidth, and provide a uni-path route if the network bandwidth is sufficient. The destination eventually determines the QoS multi-path routes and replies to the source host to perform the bandwidth reservation. Our routing scheme obviously improves the success rate of identifying a QoS route, and the simulation result demonstrates this improvement.

Keywords: CDMA-over-TDMA; Mobile ad-hoc network; Mobile computing; Quality-of-service routing; Multi-path; Wireless network

1. Introduction

Personal communications and mobile computing require a wireless network infrastructure that is fast developing, possibly multihop, and capable of multimedia service support [11]. The wireless network mainly focus on the cellular architecture for wireless personal communication networks (PCNs) supported by wired backbone infrastructure. In the cellular architecture, all mobile hosts in a communication cell can reach a base station in one hop. Separately from the single hop cellular model, another type of model, namely the mobile ad-hoc network (MANET) [8], based on radio to radio multihopping, has been evolving to serve a growing number of applications that rely on a fast development, multihop wireless infrastructure [11]. The MANET [8] consists of wireless hosts that communicate with each other in the absence of a fixed infrastructure, multihopping through wireless repeaters strategically located on the public local area, such as campus, permits the reduction of battery power and the increase in network capacity (via spatial reuse). In a MANET, host mobility can cause frequent unpredictable topology changes, thus the design of a MANET routing protocol is more complicated, because it needs more strong fault-tolerant capabilities [4,5].

Connections with quality-of-service (QoS) requirements, such as those for multimedia applications with bandwidth constraint, recently have been intensively studied QoS issues in MANETs. In this paper, the 'bandwidth' is considered as the main factor of the QoS issues by omitting the signal-to-interference ratio (SIR) and packet loss rate, etc. This is because the bandwidth guarantee is one of the most critical requirements for real-time applications. The time division multiple access

---

[☆] A preliminary version of this paper was presented at EW2002: European Wireless 2002, February 2002, Florence, Italy.

[*] Corresponding author. Tel.: +886-527-20411x33115; fax: +886-527-20959.

E-mail address: yschen@cs.ccu.edu.tw (Y.-S. Chen).

0140-3664/03/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0140-3664(03)00176-2

(TDMA) scheme [11] is generally used in the wireless network for the bandwidth reservation. ‘Bandwidth’ in time-slotted network system is measured in terms of the amount of ‘free’ slots. In general, the goal of the QoS routing protocol is searching for one (uni-path) or more paths (multi-paths) from a source to a destination such that the total bandwidth on these available paths is above the minimal requirement. To compute the bandwidth-constrained paths from source to destination, we not only have to know the available bandwidth on each link along all possible paths, but we also have to do a suitable scheduling of free slots.

These problems have been addressed in several studies in the literature [2,3,6,7,9–14]. Initially, the work in Ref. [2] assumed that the bandwidth of a link can be determined independently of its neighboring links. This strong assumption may be realized by a costly multi-antenna model 3 such that a host can send/receive using different antennas independently and simultaneously. In Ref. [2], a ticket-based QoS routing protocol was proposed to identify the QoS uni-path in a MANET. Recently, the CDMA-over-TDMA channel model has been reasonably assumed in Refs. [10,11] to provide a new MAC channel model for solving the hidden-terminal problem. Furthermore, Lin et al. developed a DSDV-based QoS routing result in Ref. [11] and Lin presented an on-demand QoS routing result in Ref. [10] under the CDMA-over-TDMA channel model. These QoS results in Refs. [10,11] both consider MAC sub-layer and network layer to calculate the end-to-end path bandwidth to offer an on-demand QoS route with satisfied the bandwidth requirement.

In this study, we propose a new on-demand QoS multi-path routing protocol in a MANET. This work calculates the end-to-end path bandwidth of a QoS multi-path routing under CDMA-over-TDMA channel model [10,11]. Our scheme offers a bandwidth routing protocol for QoS support in a multihop mobile network. Our scheme contains end-to-end bandwidth calculation and bandwidth allocation. Our protocol may produce a uni-path if the network bandwidth of a MANET is sufficient. Otherwise, our protocol offers a QoS multi-path route. The success rate of identifying the QoS uni-path or multi-path route of our approach is always better than the QoS uni-path routing schemes in Refs. [10, 11]. In short summary, a QoS route existed in our approach should be uni-path or multi-paths.

Our proposed scheme is an on-demand QoS routing protocol. For an on-demand QoS routing, a source node may dynamically request a QoS route under a given bandwidth requirement. The source node floods a path-searching packet into MANET to try to identify a QoS route and calculate the path bandwidth simultaneously. Such phase is called as the path-searching phase. The main idea of our on-demand, link-state routing protocol is to collect the information of the available link bandwidth of all paths from source to destination during the path-searching phase. This operation constructs a partial network topology, or referred to as a flow sub-graph, at the destination. The extra cost of our approach is that the path-searching packet used in our protocol is a long packet to record the link-bandwidth information. The total packet number for searching a QoS route of our approach is approximately equal to the total packet number of existing results in Refs. [10,11], which will be illustrated in detail in Section 4.

To satisfy a given bandwidth requirement, the bandwidth calculation of one or more paths are determined at the destination. It is very interesting to observe that the QoS route is eventually determined at the destination depending on whether the MANET’s network bandwidth is sufficient or insufficient. The bandwidth reservation operation is performed after the destination determines the QoS uni-path or multi-path route depending on the status of the network bandwidth. The experimental results show that it is particularly useful to adopt our proposed scheme. This is because our scheme can provide a uni-path or a multi-path QoS route. Finally, the simulation analysis demonstrates this improvement by furthermore discussing the effect of the extra cost and success rate.

The rest of the paper is organized as follows. Section 2 presents basic ideas and motivation. Our protocol is developed in Section 3 and experimental results are discussed in Section 4. Section 5 presents the conclusions.

2. Basic ideas and challenges

The network model assumptions are as follows. The MAC sub-layer in our model is implemented by using the CDMA-over-TDMA channel model. Each frame is divided into a control phase and a data phase as shown in Fig. 1. The CDMA-over-TDMA channel model is assumed by following

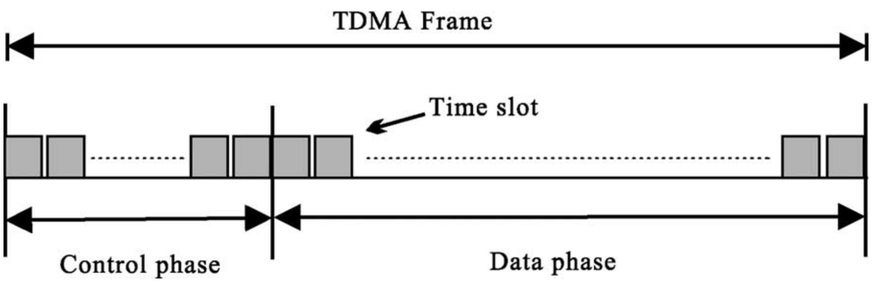
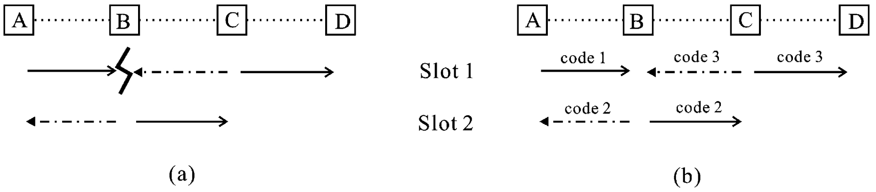


Fig. 1. TDMA frame structure.



(a) (b)

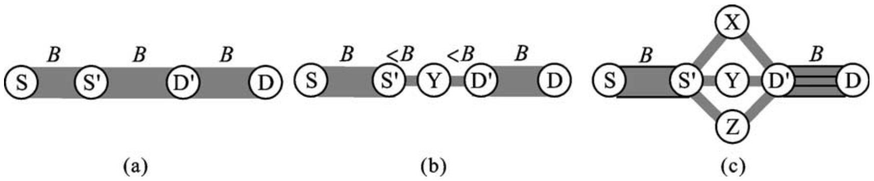
Fig. 2. (a) Hidden terminal problem, (b) CDMA-over-TDMA.

the same model as defined in Refs. [10,11]. The CDMA (code division multiple access) is overlaid on top of the TDMA infrastructure. Multiple sessions can share a common TDMA slot via CDMA. Fig. 2(a) shows a hidden-terminal problem. To overcome the problem, an orthogonal code used by a host should differ from that used by any of its two-hop neighbors as shown in Fig. 2(b). A code assignment protocol should be supported (this can be regarded as an independent problem, which can be found in Refs. [10,11]). The bandwidth requirement is realized by reserving time slots on links. Under such a model, the use of a time slot on a link is only dependent on the status of its one-hop neighboring links. This model can be emulated by wireless LAN cards which follow the IEEE 802.11 standard [1].

Recently, Liao et al. presented a QoS multi-path routing approach [9] based on a ticket-distribution scheme under a quite ideal model which was assumed in Ref. [2]. The multi-path approach was developed under an ideal model, which does not consider the radio interference problem. Therefore, the challenge of this work was in attempting to design a QoS multi-path routing in a stronger model, which is the CDMA-over-TDMA channel model. We first review Liao et al.'s multi-path approach [9]. Their ticket-distribution scheme splits the bandwidth requirement if all neighboring nodes' link bandwidths are smaller than the bandwidth requirement. That is, the ticket is partitioned into some subtickets, and the original bandwidth requirement is split into sub-bandwidth requirements, where each sub-ticket is in charge of one sub-bandwidth requirement. Let B_x^y denote the link bandwidth from node x to node y as B, and denote $(h_1, h_2, \ldots, h_k)$ as a path from node $h_1, h_2, \ldots,$ to node h_k. Given that a source node S initiates a QoS Route REQuest (QRREQ) with a bandwidth requirement B, Fig. 3(a) reveals that a successful QoS route, from S to D, is constructed, with link bandwidths of (S, D') and (D', D) being $B_{S'}^{D'}$ and $B_{D'}^{D}$, respectively. Fig. 3(b) shows a failed QoS route since the sub-path bandwidth of (S', D') is $b_{S'}^{D'}$, where $b < B$. This indicates that a QoS route may fail if we do not know the enough link bandwidth information in a MANET. However, multi-paths (S', X, D'), (S', Y, D'), and (S', Z, D') can be used for the QRREQ with bandwidth requirement B. We assume that multi-paths exist in the network. A ticket herein should be split into sub-tickets to find a QoS route. However, if there are no multi-paths, then the QoS fails. Fig. 3(c) shows that the bandwidth requirement B is divided into three sub-bandwidth requirements, while each sub-ticket is responsible for searching a multi-path to cover its corresponding sub-bandwidth requirement. For instance, (S, S', X, D', D), (S, S', Y, D', D), and (S, S', Z, D', D) are three multi-paths from source S to destination D. Each multi-path is responsible for one sub-bandwidth requirement. Notably, multi-paths are allowed to share the same sub-paths. For instance, multi-paths (S, S', X, D', D), (S, S', Y, D', D), and (S, S', Z, D', D) share (S, S') and (D', D). This indicates that all multi-paths are not necessarily disjointed.

Our proposed scheme is an on-demand routing protocol. Some global link-state information is collected in advance before determining the selected multi-path route at the destination. This information is recorded in a long packet during the QoS multi-path searching. However, as we mentioned previously, the total number of path-searching packets is approximately equal to existing QoS routing schemes in Refs. [10,11], since the packet flooding operation is performed in the QoS multi-path searching phase.

The traditional link-state records all information of network topology, however this is difficult and inefficient. This protocol reactively collects link-state information from source to destination. This forms a flow-network. Based on the flow-network, a better multi-path result than that of Liao et al. [9] is obtained. The overview of the proposed protocol is given now. In a MANET, a mobile host knows the available bandwidth to each of its neighbors. When a source node S needs a route to a destination D of bandwidth B, it will send out some Route REQuest (RREQ) packets, each of which carries the path history and link-state information. Each RREQ packet records all link-state information from source to destination. The destination collects all possible link-state information from different RREQ packets sent



(a) (b) (c)

Fig. 3. Multi-path approach.

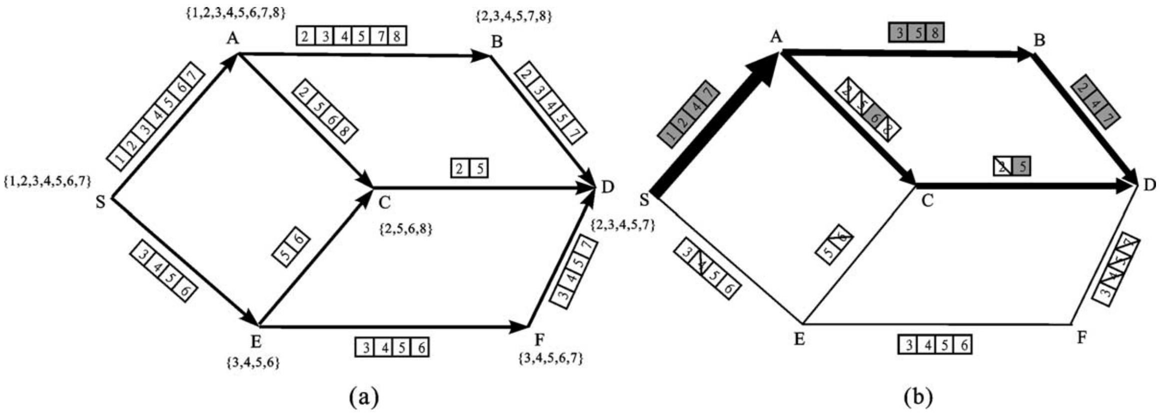


Fig. 4. Example of multi-paths in a CDMA-over-TDMA channel model.

from the source. A partial network, which is a flow-network, is constructed in the destination node after receiving multiple information packets. An algorithm is applied at the destination to determine a better result for QoS multi-path routing. After determining a multi-path route, a reply packet is sent from the destination to the source. On the reply's way back to the source, the bandwidths are confirmed and reserved. Observe that, Liao et al.'s scheme possibly fails to identify the QoS route although multi-paths exist in a MANET, since Liao et al.'s scheme uses the hop-by-hop QoS route discovery operation. This shortcoming can be overcome by using our on-demand, link-state, multi-path QoS routing protocol.

Some notations are defined herein. Let $\{\alpha_1, \alpha_2, \ldots, \alpha_{\kappa-1}\}$ denote the free time-slots list of a node. Let $\overrightarrow{XY}$ denote a link from X to Y. If free time-slot lists of two neighboring nodes A and B are $\{\alpha_1, \alpha_2, \ldots, \alpha_{\kappa_1}\}$ and $\{\beta_1, \beta_2, \ldots, \beta_{\kappa_2}\}$, $\kappa_1 \neq \kappa_2$, we define an intersection function $\cap(\{\alpha_1, \alpha_2, \ldots, \alpha_{\kappa_1}\}, \{\beta_1, \beta_2, \ldots, \beta_{\kappa_2}\}) = [\gamma_1, \gamma_2, \ldots, \gamma_{\kappa_3}]$, where $[\gamma_1, \gamma_2, \ldots, \gamma_{\kappa_3}] \in \{\alpha_1, \alpha_2, \ldots, \alpha_{\kappa_1}\}$, $[\gamma_1, \gamma_2, \ldots, \gamma_{\kappa_3}] \in \{\beta_1, \beta_2, \ldots, \beta_{\kappa_2}\}$, and $\kappa_3 \leq \min\{\kappa_1, \kappa_2\}$. Let $[\gamma_1, \gamma_2, \ldots, \gamma_{\kappa_3}]$ represent shared free time-slots between nodes A and B. This indicates that time slots for communicating between A and B must be selected from $[\gamma_1, \gamma_2, \ldots, \gamma_{\kappa_3}]$. For instance as shown in Fig. 4(a), if the free time-slot lists of S and A are {1, 2, 3, 4, 5, 6, 7} and {1, 2, 3, 4, 5, 6, 7, 8} then $\cap(\{1,2,3,4,5,6,7\}, \{1,2,3,4,5,6,7,8\}) = [1,2,3,4,5,6,7]$. Time slots will be reserved from the shared time-slot list of [1, 2, 3, 4, 5, 6, 7].

Our multi-path routing is constructed using multiple uni-paths. An example of multi-path routing is shown in Fig. 4(b); paths (S, A, B, D) and (S, A, C, D) are established if the path bandwidth requirement is four slots. Before describing the multi-path scheme, we now discuss the basic idea of the uni-path routing scheme adopted in this paper. Under the CDMA-over-TDMA channel model, Lin [10,11] calculated the end-to-end uni-path bandwidth by the hop-by-hop calculation for the time slot reservation. For instance as shown in Fig. 5, a QRREQ, with path requirement = two slots, is sent from source node A to destination node F. The hop-by-hop time slot reservation repeatedly calculates the free time slots between two adjacent nodes. From node A, we initially allocate time slots {1, 4} and {2, 7} to $\overrightarrow{AB}$ and $\overrightarrow{BC}$, respectively. Continuing, the time slot {4, 5} is allocated to $\overrightarrow{CD}$. Unfortunately, this QoS route fails in link $\overrightarrow{DE}$ since there is only one free time slot {8} in $\overrightarrow{DE}$. Under the same network environment, however, it is possible to exploit a QoS uni-path, which satisfies path

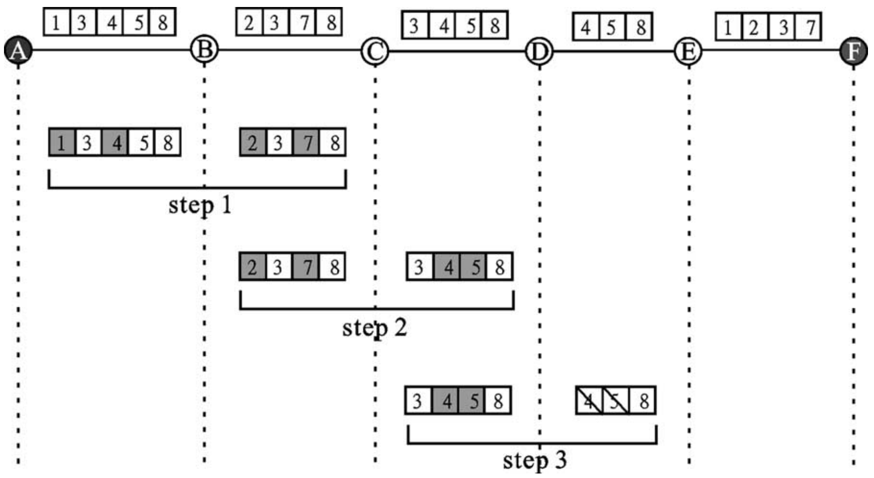


Fig. 5. Example of a hop-by-hop time slot reservation scheme.

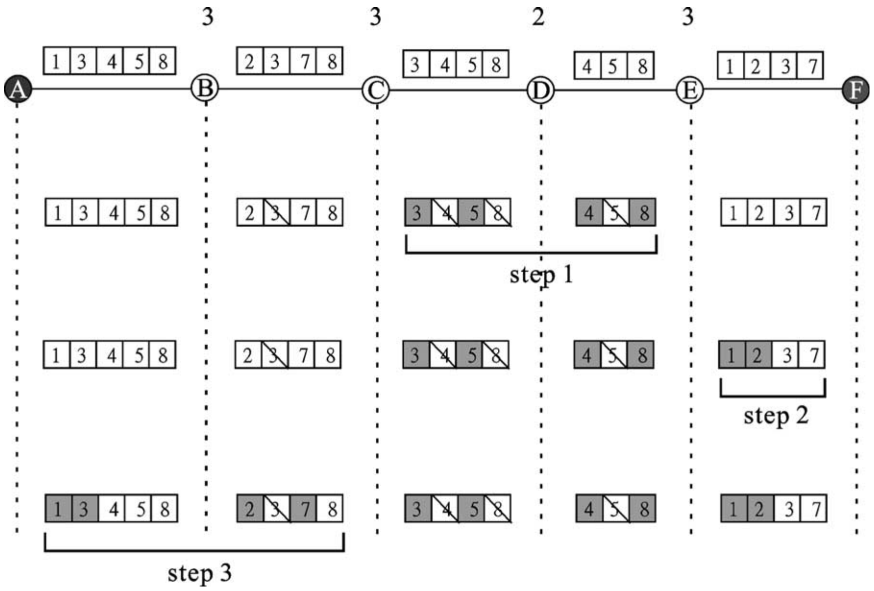


Fig. 6. Example of slot reservation by our QoS uni-path protocol.

requirement = two slots as shown in Fig. 6. We can first allocate time slots {3, 5} and {4, 8} to $\overrightarrow{CD}$ and $\overrightarrow{DE}$, respectively. Continuing, slot {1, 2} is reserved to $\overrightarrow{EF}$, and then slots {1, 3} and {2, 7} are dispatched to $\overrightarrow{AB}$ and $\overrightarrow{BC}$, respectively. Therefore, a uni-path with two slots is established from A to F. Obviously, this is not the hop-by-hop calculation of time slot reservation. Existing on-demand routing protocols [10,11] usually adopt the hop-by-hop calculation for time-slot reservation. The property of the hop-by-hop calculation is that each node only needs to maintain the link bandwidth of its neighbors.

This indicates that our scheme has to reactively collect link-state information from source to destination. After the destination collects the current link-state information, an efficient algorithm of QoS uni-path time slot reservation is developed in Section 3.2. Based on the new uni-path routing result, multi-path routing scheme is presented in Section 3.3. The main contribution of the multi-path routing scheme is to increase the success rate for seeking a QoS route.

3. The on-demand, link-state, multi-path QoS routing protocol

We first give an overview of our on-demand, link-state, multi-path QoS routing protocol. The proposed protocol mainly constructs multi-paths from source to destination. A reactive QoS multi-path routing protocol is achieved by the development of *on-demand, link-state delivery/collection, uni-path discovery, multi-path discovery, multi-path reply,* and *multi-path maintenance* phases. The on-demand, link-state delivery/collection phase describes how to deliver and collect link-state information from source to destination in a MANET. In the uni-path discovery phase, we present two efficient schemes for uni-path time-slot reservation. In the multi-path discovery phase, we further design a multi-path time-slot reservation strategy based on the uni-path results. A route-reply packet, in the multi-path reply phase, is sent back from the destination to the source in order to confirm the reserved time slots for the final multi-paths. It is noted that the multi-path maintenance phase is performed in order to keep a highly stable QoS route since the robust-path capability is strengthened using our approach.

3.1. Phase 1: on-demand, link-state delivery and collection

The source node initiates a QRREQ packet and floods the MANET until arriving at its destination. Each packet records the path history and all link-state information. The link-state information is delivered from the source toward to the destination. The destination possibly collects link-state information from different QRREQ packets, each of which travels along a different paths.

For each bandwidth request, a number of QRREQ packets may be sent. Each QRREQ packet is responsible for searching a path from source to destination nodes. However, final paths are eventually selected from all of the paths which are received by the destination. A QRREQ packet is denoted as *QRREQ* (*S*, *D*, *node_history*, *free_time_slot_list*, *B*, *TTL*), where each field of the packet is defined as follows:

- *S* : the source host;
- *D* : the destination host;
- *node_history*: a list of nodes, which denotes the node history which records the path from source to the current traversed node;
- *free_time_slot_list*: a list of free time slot of links, each of which records free time slots among the current traversed node and the last node recorded in the *node_history*;
- *B* : the bandwidth requirement from *S* to *D*; and

- *TTL*: (Time To Live) the limitation of hop-length of the search path.

We now formally define the on-demand, link-state delivery/collection operation as follows.

(A1) Source node S initiates and floods a *QRREQ* (S, D, *node_history* = $\{S\}$, *link_bandwidth_list* = $\{\ \}$, B, *TTL*) packet into the MANET toward the destination node D if the given bandwidth requirement is B.

(A2) If node e receives a *QRREQ* packet, then we add e into *node_history* and append the free time slots of node e and the last node recorded in the *node_history* into the *free_time_slot list*, and decrease the value of the *TTL*. If e is not the destination and *TTL* is not equal to zero, then we re-forward the packet to all neighboring nodes which do not exist in *node_history*.

The destination eventually receives many different *QRREQ* packets from the source. The destination will re-configure the network topology and all corresponding free time-slot information. For instance as shown in Fig. 7(a)–(d), *QRREQ* (S, D, $\{S, A, B, D\}$, [$\{1, 2, 3, 4, 5, 6, 7\}$, $(2, 3, 4, 5, 7, 8\}$, $\{2, 3, 4, 5, 7\}$, B, *TTL*), *QRREQ* (S, D, $\{S, A, C, D\}$, [$\{1, 2, 3, 4, 5, 6, 7\}$, $\{2, 5, 6, 8\}$, $\{2, 5\}$, B, *TTL*), *QRREQ* (S, D, $\{S, E, C, D\}$, [$\{3, 4, 5, 6\}$, $\{5, 6\}$, $\{2, 5\}$, B, *TTL*), and *QRREQ* (S, D, $\{S, E, F, D\}$, [$\{3, 4, 5, 6\}$, $\{3, 4, 5, 6\}$, $\{3, 4, 5, 7\}$, B, *TTL*) packets are collected at the destination node D. A partial network topology, as shown in Fig. 4(a), is re-configured at destination node D. Based on the re-constructed partial MANET topology, the uni-path and multi-path discovery operations are described in phase 2.

3.2. Phase 2: uni-path discovery

Before describing the multi-path discovery operation, we first discuss the uni-path discovery operation. Observe that a network topology with link-state information is re-constructed in the destination node. Therefore, all of the uni-path and multi-path discovery operations are identified in the destination. Our uni-path discovery operation is accomplished by constructing a least-cost-first time-slot reservation tree T_{LCF}. Before describing construction of the T_{LCF} tree, the traditional hop-by-hop time-slot reservation is again discussed herein, because our approach does not use hop-by-hop.

Existing on-demand time-slot reservation results mainly calculate the end-to-end path bandwidth from source to destination by the hop-by-hop approach. For instance, Lin's approach [10] is one of them. For example, there is a path ($A, B, C, \ldots, F$) between source node A and destination node F. Let $a, b, c, \ldots, e$ denote free time slots of links $\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CD}, \ldots, \overrightarrow{EF}$, respectively, as shown in Fig. 8(a).

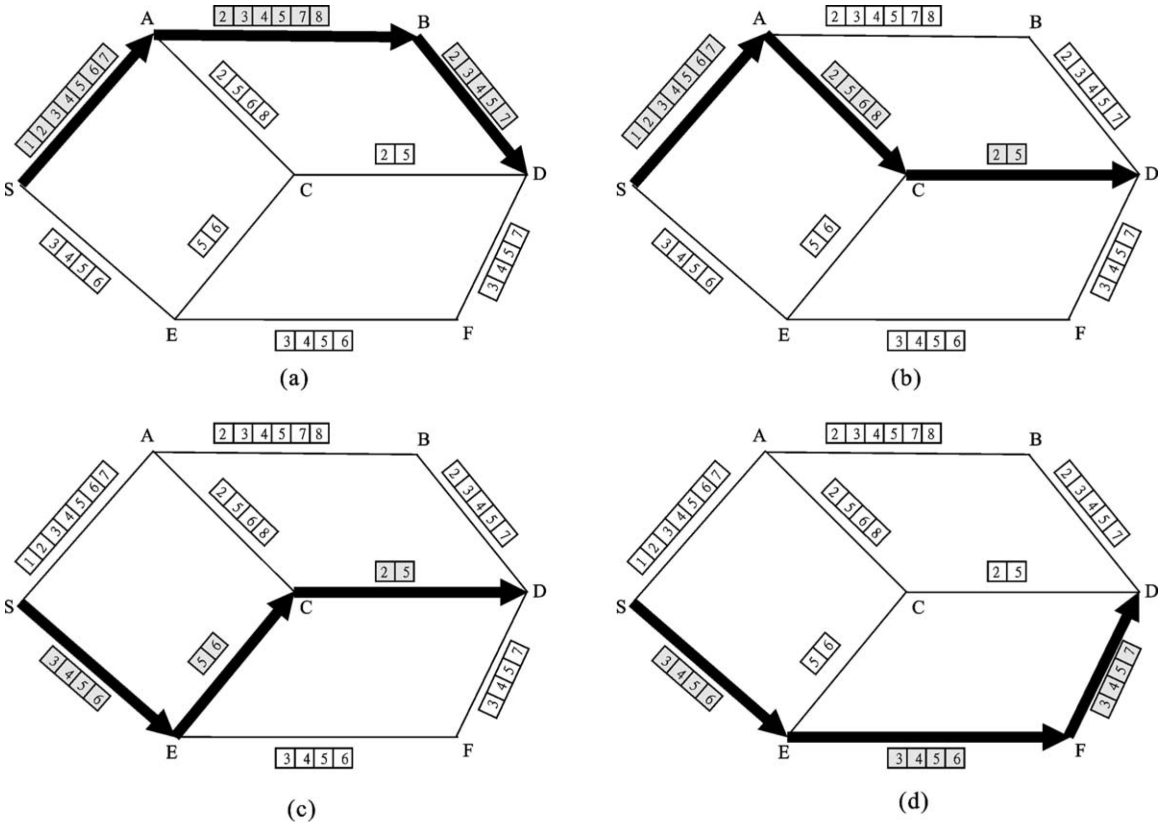


Fig. 7. On-demand, link-state delivery and collection operation.

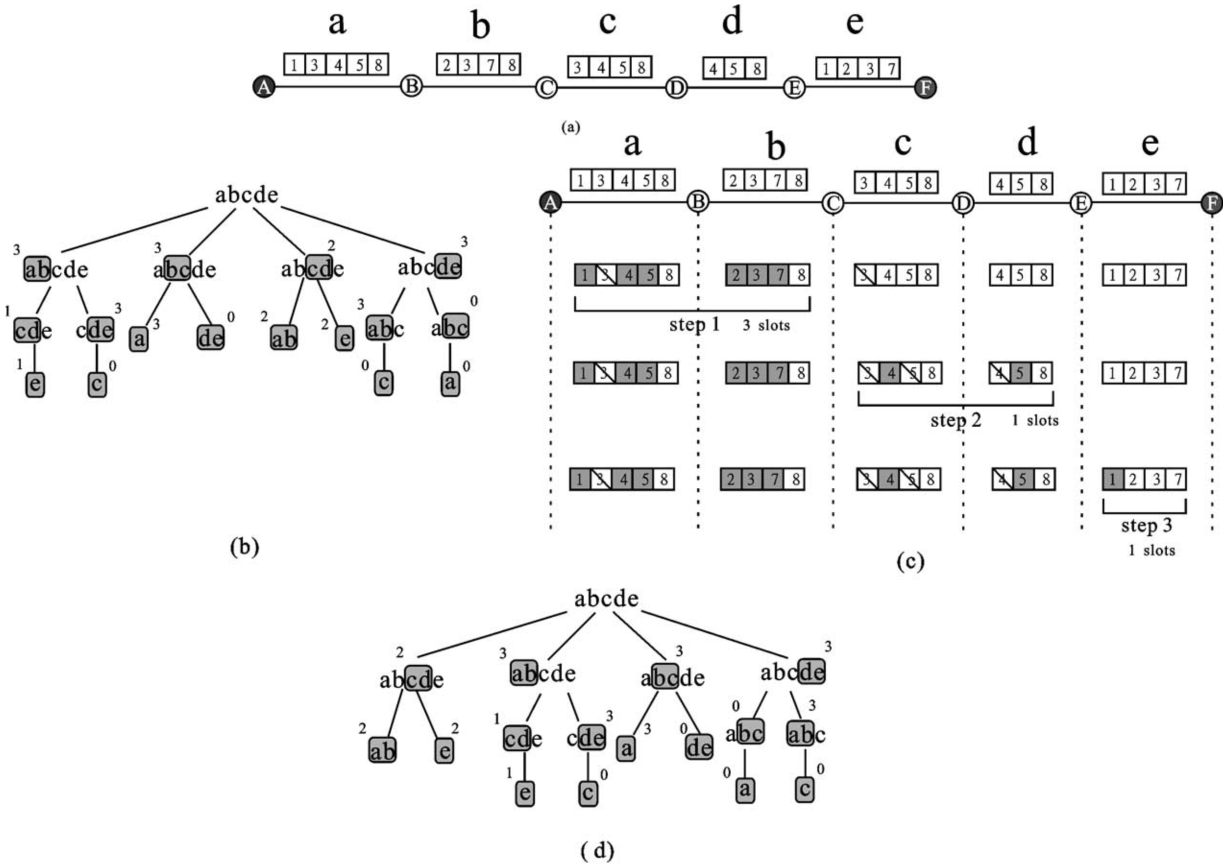


Fig. 8. (a) A path with link bandwidths; (b) T tree; (c) example of uni-path time-slot reservation using T; and (d) T_{LCF} tree.

To follow the hop-by-hop reserving sequence, our reservation scheme follows the order of $\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CD}, \ldots, \overrightarrow{EF}$. The example shown in Fig. 8(c) is a hop-by-hop reservation scheme. A maximal reserved time-slot number is denoted to reserve the largest number of time slots of a link in a path. For example in Fig. 8(c), if $a = \{1, 3, 4, 5, 8\}$ and $b = \{2, 3, 7, 8\}$, then [1, 4, 5] is reserved to link $\overrightarrow{AB}$ and [2, 3, 7] is reserved to link $\overrightarrow{BC}$. The maximal reserved timeslot numbers of $\overrightarrow{AB}$ and $\overrightarrow{BC}$ are 3. Additionally, slot [2, 3, 7] is allocated to $\overrightarrow{BC}$, and the free time slot of c is thus updated from [3, 4, 5, 8] to [4, 5, 8]. Further, [4] is reserved to link $\overrightarrow{CD}$ and [5] is reserved to $\overrightarrow{DE}$, so [1] is then given to $\overrightarrow{EF}$. Therefore, the path bandwidth of path (A, B, C, D, E, F) is 1 using hop-by-hop reservation. Observe that, our approach does not use traditional hop-by-hop slot reservation. Efforts are made to acquire greater path bandwidth than that acquired using hop-by-hop reservation. For example as shown in Fig. 6, a path with bandwidth = two slots exists by not adopting the hop-by-hop slot reservation.

The purpose of constructing tree T_{LCF} is to identify a path with maximal path bandwidth. Before formally defining tree T_{LCF}, a time-slot reservation tree T is constructed. Observe that the T_{LCF} and T trees are used to efficiently reserve time slots for a uni-path. This indicates that our time-slot reservation scheme does not follow the order of $\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CD}, \ldots, \overrightarrow{EF}$. Observe that trees T and T_{LCF} are constructed to represent all possible conditions of time-slot reservation. Given a path $(A, B, C, D, E, \ldots, Y, Z)$, let $abcd\cdots yz$ denote free time slots of links $\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CD}, \ldots, \overrightarrow{YZ}$. Assume that x and x' are free time lists of links $\overrightarrow{XX'}$ and $\overrightarrow{X'X''}$, let $ab\cdots\underline{xx'}\cdots y$ denote the reserved time slots to links $\overrightarrow{XX'}$ and $\overrightarrow{X'X''}$ in the first order, and the time slot is selected from x and x'. A time-slot reservation tree T is constructed by the breadth-first-searching approach, which is formally defined below.

Definition 1. *Time-slot reservation tree T.* Given a path $(A, B, C, D, E, \ldots, Y, Z)$, let the root of T be represented as $abcd\cdots yz$. Children nodes of the root are $\underline{ab}cd\cdots yz$, $a\underline{bc}d\cdots yz$, $ab\underline{cd}\cdots yz$, and $abcd\cdots\underline{yz}$, which form the first level of tree T. The tree recursively expands all children nodes of each node on each level of tree T, and follows the same rules of the first level of tree T until reaching the leaf nodes. Observe that leaf nodes only exist as one component. Therefore, time-slot reservation tree T is constructed.

For instance, a tree T is constructed as shown in Fig. 8(b). Observe that, in Fig. 8(b), the maximal reserved time-slot numbers are 3, 3, 2, 3 from left to right in the first level of tree T. Our uni-path time-slot reservation utilizes tree traversal by the depth-first-searching order, which is formally given below.

(B1) Given a time-slot reservation tree T and path bandwidth B, tree T is traversed by the depth-first-searching order. Each path from root to leaf nodes forms a time-slot reservation pattern. This pattern is used to reserve time slots from source to destination.

- For instance, as shown in Fig. 8(b), the first reservation pattern is $\underline{ab}$, $\underline{cd}$, and $\underline{e}$, whose reserved time slot is 1, and the second reservation pattern is $\underline{ab}$, $\underline{de}$, and $\underline{c}$, whose reserved time slot is 0.

(B2) If a new reservation pattern exists to reserve a path bandwidth B', and $B' < B$, then we proceed to traverse tree T until identifying other reservation patterns, and then go to step B2. Otherwise, if tree traversal is finished or $B' \geq B$, then exits the procedure.

All possible reservation patterns are identified, and their corresponding path bandwidths are exploited; therefore, a maximal path bandwidth is exploited. To reduce the time needed to search a path while satisfying a given bandwidth requirement B, tree T is modified to be the least-cost-first time-slot reservation tree T_{LCF} as follows.

Definition 2. *Least-cost-first time-slot reservation tree* T_{LCF}. A time-slot reservation tree T is said to be a least-cost time-slot reservation tree T_{LCF} if the children nodes on each level of tree T are sorted by the maximal reserved time-slot number from left to right in ascending order.

For instance as shown in Fig. 8(d), tree T_{LCF} is obtained from tree T as follows. Children nodes of the root of tree T are $\underline{ab}cde$, $a\underline{bc}de$, $ab\underline{cd}e$, and $abc\underline{de}$, but children nodes of the root of tree T_{LCF} are $ab\underline{cd}e$, $\underline{ab}cde$, $a\underline{bc}de$, and $abc\underline{de}$, since the maximal reserved time-slot numbers on the first level of tree T_{LCF}, from left to right, are 2, 3, 3, and 3. The uni-path time-slot reservation algorithm is the same for steps B1 and B2.

(C1) This part is the same as step B1, except for providing the least-cost-first time-slot reservation tree T_{LCF} and path bandwidth B.

(C2) This part is the same as step B2, except for traversing the least-cost-first time-slot reservation tree T_{LCF}.

For instance, as shown in Fig. 8(d), the first reservation pattern is $\underline{cd}$, $\underline{ab}$, and $\underline{e}$, whose reserved time slot is 2, and the second reservation pattern is $\underline{ab}$, $\underline{cd}$, and $\underline{e}$, whose reserved time slot is 1. Comparing T-tree traversal with T_{LCF}-tree traversal schemes, the T_{LCF}-tree traversal scheme is more efficient than the T-tree traversal scheme.

A simple result of searching a uni-path time-slot reservation can be used without constructing and traversing trees T and T_{LCF}, which is stated as follows. The result is the same as the first reservation pattern in the T_{LCF} tree. This reservation pattern is easily obtained as follows. Given a path $(A, B, C, D, E, \ldots, Y, Z)$, and let $abcd\cdots yz$ denote free time slots of links $\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CD}, \ldots, \overrightarrow{YZ}$. Assume that x and x' are free time lists of links XX' and $X'X''$, and let $ab\cdots\underline{xx'}\cdots yz$ denote reserved time slots of links XX' and $X'X''$ in the first order if $\underline{xx'}$ has the smallest value of maximal reserved time-slot number, where time slots are selected from x and x'. If $\underline{xx'}$ is determined, then a second-largest value of maximal reserved-time slot number of $\underline{tt'}$, $\underline{tt'} \in ab\cdots yz - \underline{xx'}$, is reserved as time slots over and over again, until a simple reservation pattern is obtained. An example is given in Fig. 6.

3.3. Phase 3: multi-path discovery and reply

A uni-path discovery operation is presented in phase 2. Our multi-path discovery operation sequentially exploits multiple uni-paths such that the total sum of path bandwidths fulfills the original path bandwidth B. A centralized algorithm is proposed at the destination to determine the multi-paths. Given a path bandwidth of B, the multi-path discovery algorithm is formally given as follows.

(D1) Let *Bandwidth_Sum* denote the total sum of multiple uni-paths. Initially, we set $Bandwidth\_Sum = 0$.

(D2) The destination waits for a period of time to obtain a possible uni-path from the source node, while $Bandwidth\_Sum < B$. A uni-path discovery procedure is applied to such a uni-path to acquire its maximal path bandwidth b. Let $Bandwidth\_Sum = Bandwidth\_Sum + b$. If $Bandwidth\_Sum < B$, then modify all link-state information of the network topology according to the current constructed uni-path and then go to step D2. Observe that all of the modifying operations are carried out at the destination node. Otherwise, if $Bandwidth\_Sum \geq B$, then exits the procedure.

For example, a flow network with link-state information is re-configurated at destination node D, which is shown in Fig. 9(a). A uni-path (S, A, B, D) with three slots is identified if the bandwidth requirement is three slots. Therefore, our protocol produces a uni-path. This case is occurred if the network bandwidth is sufficient.

If the original bandwidth requirement is four slots, we continue to modify the link-state information as shown in Fig. 9(b). The second uni-path with two slots and third uni-path with two slots are constructed and shown in Fig. 9(c) and (d), respectively. All identified uni-paths are also displayed in Fig. 10(a). If the bandwidth requirement is four slots, then we may select a first and a second uni-path, but the second uni-path only reserves one slot as shown in Fig. 10(b) and (c). Therefore, multi-paths from source node S to destination node D are constructed as shown in

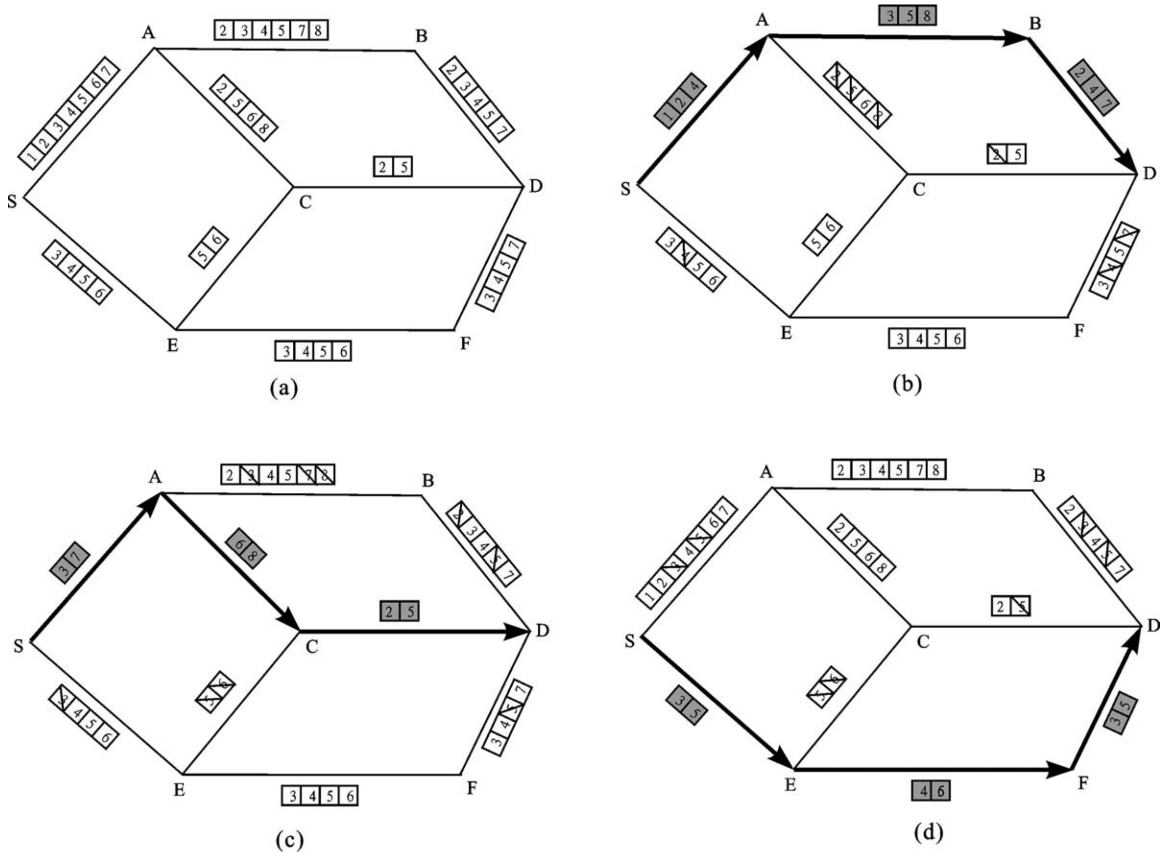


Fig. 9. The original network and found multi-paths.

Fig. 10(d). Eventually, when the destination determines the multi-paths, destination initiates a reply packet from destination to source to reserve the time-slots in the multi-paths and release the time-slots for other no used paths.

3.4. Phase 4: multi-path maintenance

The multi-path maintenance presented herein provides single-path tolerance capability. This multi-path maintenance is achieved by adding a backup path into the multi-path. Observe that this backup path must be a fully disjointed path from the identified multi-path. The path bandwidth of the backup path must be large enough such that if any uni-path of our multi-path fails, the backup path can be used immediately. For example, if all paths of the identified multi-path are disjointed, we can search a backup path with maximum path bandwidth among all paths. For instance as shown in Fig. 11(a), a multi-path is found by paths (S, B, T) and (S, C, E, T); whose path bandwidths are three and two, respectively. A backup and disjointed path (S, A, D, T); whose path bandwidth $= 3$, is constructed. Fig. 11(b) shows that if path (S, B, T) fails, then backup path is used. Fig. 11(c) shows that if path (S, C, E, T) fails, then a backup path is used. Observe that the total path bandwidth is the same even if path (S, B, T) or (S, C, E, T) fails.

4. Experimental results

To examine the effectiveness of our approach, we compare our protocol with an existing on-demand QoS routing scheme presented in Ref. [10], which is termed the LIN. Our scheme is denoted MP1 if the multi-path discovery operation adopts the constructed *TLCF* tree and denoted MP2 if the multi-path discovery operation does not use the constructed *TLCF* tree, which is a simplified version (as described in Section 3.2). The simulation parameters in the simulator are given below.

- The number of mobile hosts ranges from 20 to 40.
- The number of time slots in the data phase of a frame is assumed to be 16 slots.
- Three different bandwidth requirements are 1, 2 and 4 slots. We compare the number with LIN-x, MP1-x, and MP2-x, $x = 1, 2, 4$, where x denotes the bandwidth requirement B of each QoS request.
- The network density is set from 25 to 75%. The low or high value of network density indicates that each mobile node has the greater or less number of free time slots to be used.
- The mobility ranges from 2 to 10 ft/s.
- Each simulated result is obtained by average values through 5000 runs.

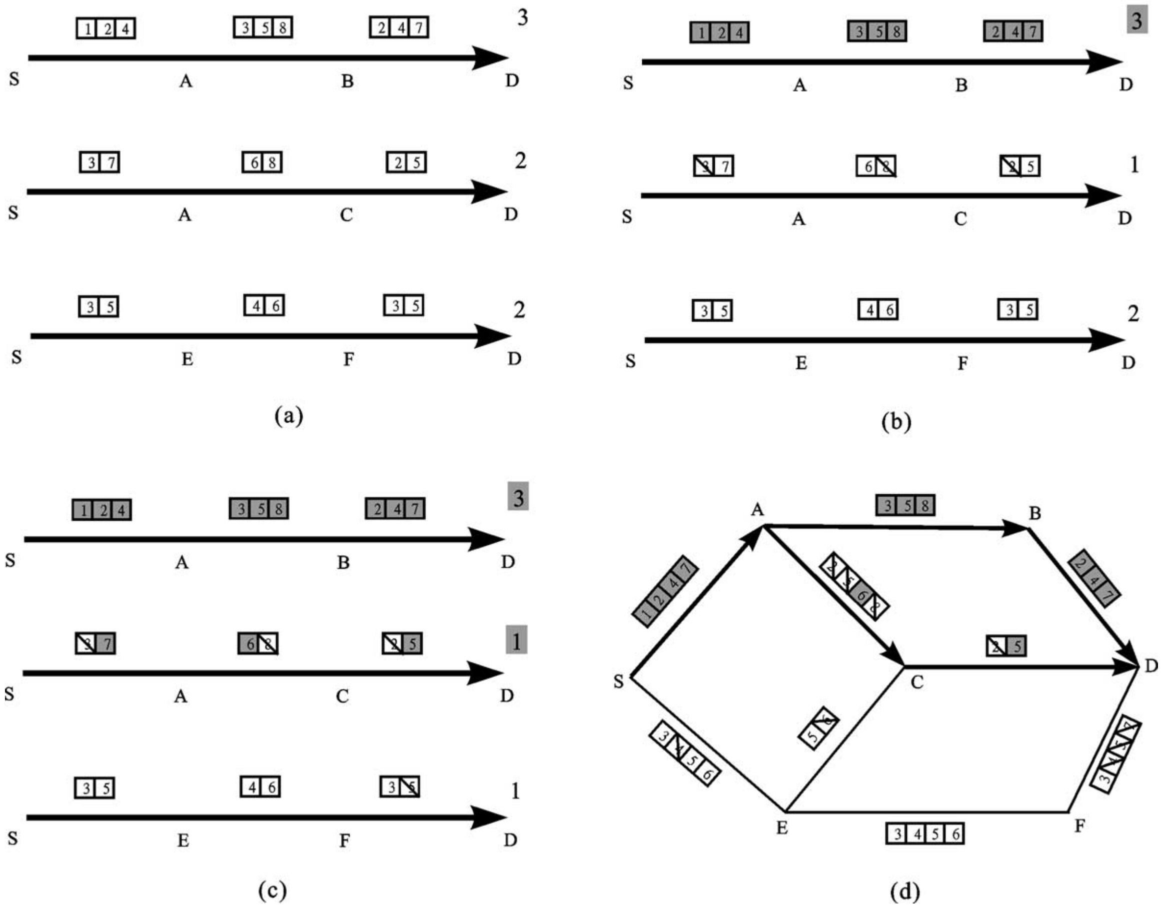


Fig. 10. Example of multi-path discovery.

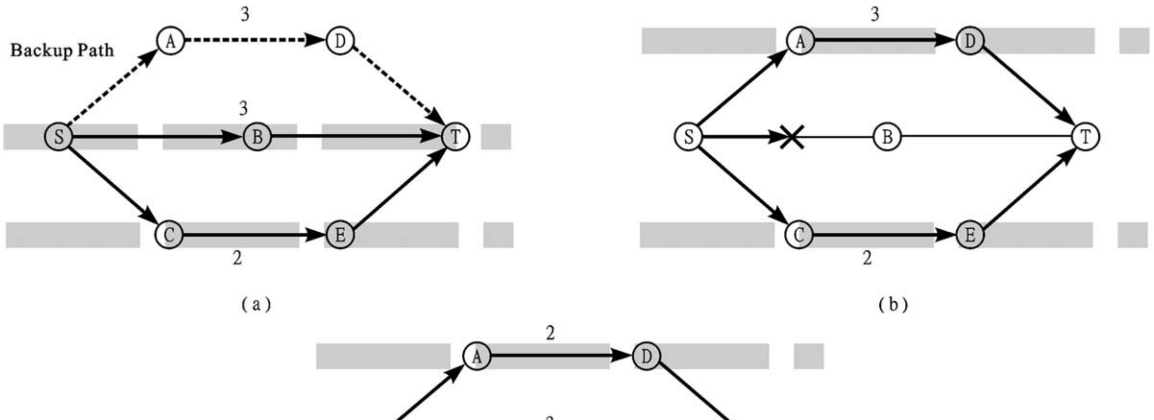


Fig. 11. Example of multi-path maintenance.

The simulator is simulated in a $1000 \times 1000\text{-m}^2$ area. The radio transmission range is 400 m. The data rate is 4 Mb/s. The duration of each time slot of a time frame is assumed to be 5 ms, and the duration of a control slot is 0.1 ms. The source and destination are selected randomly.

Once a QoS request is successful, a time slot is reserved for all subsequent packets. The reservation is released when either the data transmission process is finished or the link is broken. A packet is dropped if the packet residing in a node exceeds the maximal queuing delay time, which is set to four frame lengths (328 ms). The performance metrics of our simulation are given below.

- *Success_Rate* (*SR*): the number of successful QRREQs divided by the total number of QRREQs from source to destination.
- *Slot_Utilization* (*SU*): the average slot utilization of every link in all QoS routes.
- *OverHead* (*OH*): the hop count of routing packets being transmitted divided by the total number of QoS requests.
- *Incomplete Rate* (*IR*): the number of broken connections divided by the number of successful QoS requests.

It is worth mentioning that an efficient QoS routing protocol is achieved with a high success rate (SR), high slot utilization (SU), low overhead (OH), and low incomplete rate (IR). In the following, we illustrate our simulation results of SR, SU, OH, and IR from several prospects.

4.1. Performance of Success Rate

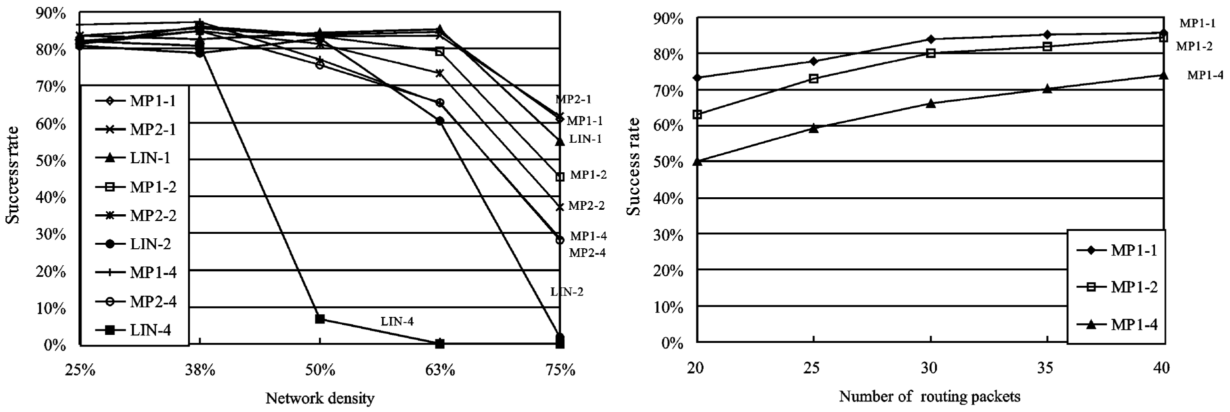
In the following, we illustrate the performance of success rate vs. effect of network density and number of routing packets.

(1A) *Effect of network density*. Each value in Fig. 12(a) is obtained by assuming that the number of mobile hosts is 30 and the number of routing packets is 30. Fig. 12(a) shows the success rate of searching a QoS route vs. network density. Observe that our approach acquires a higher success rate than does the Lin approach under a network density from low to high. This indicates that our proposed protocol outperforms the Lin protocol, especially if the network density is high. This is because our multi-path scheme can indeed improve the success rate, and our uni-path scheme has a higher success rate than the Lin scheme. Fig. 12(a) shows that if the bandwidth requirement is 4 and the network density is larger than 63%, the Lin protocol cannot find any QoS route because there is no uni-path which satisfies the bandwidth requirement. However, our protocol still has a success rate of 65 (density = 63%) and 28% (density = 75%) due to the multi-path routing.

(1B) *Effect of number of routing packets*. Every value in Fig. 12(b) is obtained by assuming that the number of mobile hosts is 30 and the network density is 63%. The number of routing packets reflects the total number of *QRREQ* packets in a MANET. Each *QRREQ* packet represents a possible path from source to destination if this *QQREQ* packet can reach the destination. The greater the number of routing packets is, the more-accurate the network topology will be re-configured at the destination. Fig. 12(b) shows that a high success rate is obtained when using a greater number of *QRREQ* packets. Fig. 12(b) shows that our approach has a greater than 70% success rate if the source sends more than 40 *QRREQ* packets. This simulation result uses and controls the number of *QRREQ* packets to represent the flooding scheme in our multi-path scheme.

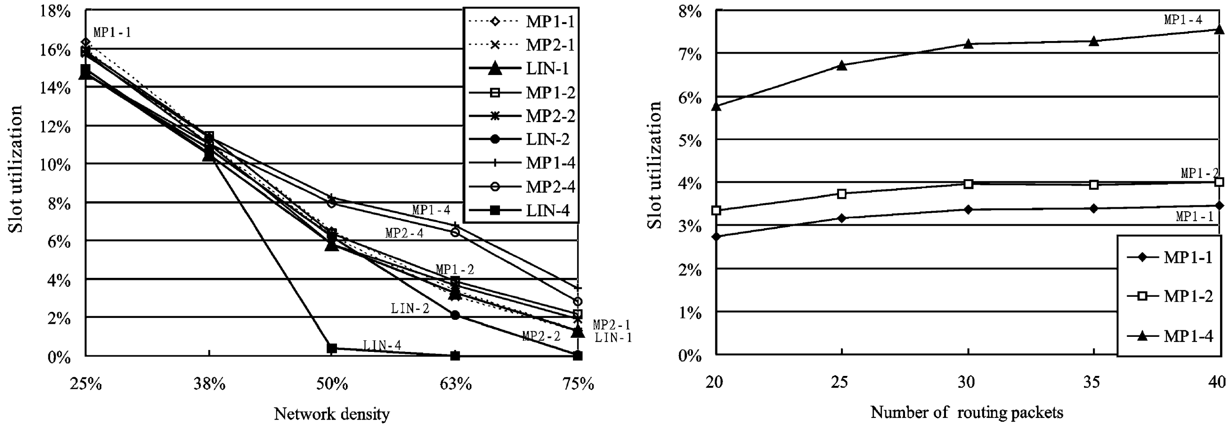
4.2. Performance of Slot Utilization

Fig. 13 shows the performance of slot utilization vs. effect of network density and number of routing packets as follows. Two types of effects are discussed.



(a) (b)

Fig. 12. Success rate vs. (a) network density (number of mobile hosts = 30 and number of routing packets = 30), (b) number of routing packets (number of mobile hosts = 30 and network density = 63).



(a) (b)

Fig. 13. Slot utilization vs. (a) network density (number of mobile hosts = 30 and number of routing packets = 30), (b) number of routing packets (number of mobile hosts = 30 and network density = 63).

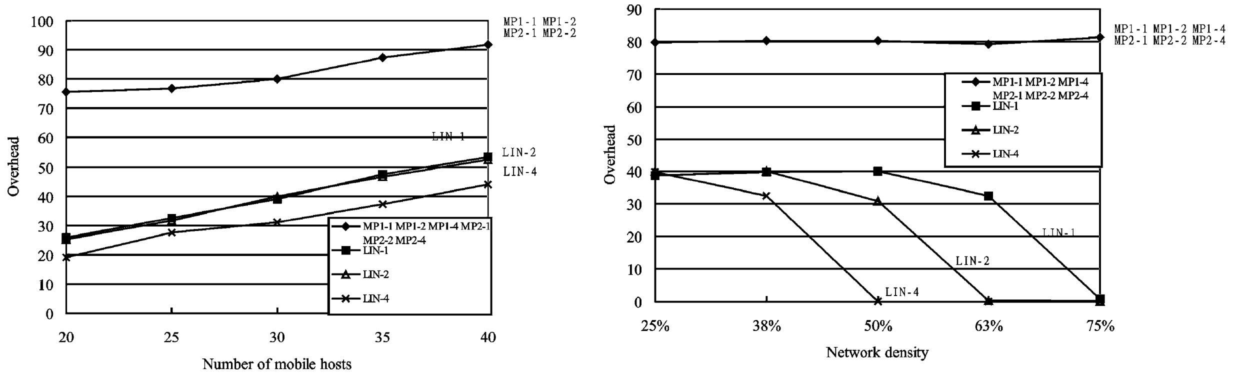
(2A) *Effect of network density*. The simulation assumption is the same for case 1A. Fig. 13(a) shows that if the bandwidth requirement is 4 slots, our MP protocol provides at least 7% more slot utilization than does the Lin protocol. This is because Lin's protocol finds a route with difficulty if the bandwidth requirement is high, therefore the slot utilization is low. This reveals the advantages of our approach.

(2B) *Effect of number of routing packets*. The simulation assumption is the same as for case 1B. Fig. 13(b) compares slot utilization under various numbers of routing packets. Our approach produces at least a 6% increase in slot utilization if a greater number of routing packets is used. This also reflects the result from Fig. 13(b), that the higher success rate obtains better slot utilization.

4.3. Performance of Overhead

Fig. 14 shows the performance of overhead vs. effect of network density and the number of routing packets. Two kinds of effect are described.

(3A) *Effect of number of mobile hosts*. Each value in Fig. 14(a) is obtained by assuming that the number of mobile hosts is 30 and the network density is 38%. Fig. 14(a) shows the performance of overhead under various numbers of mobile hosts. Observe that our MP protocol has the same overhead for MP1-x and MP2-x, where $x = 1$, 2, and 4, since the overhead of our approach mainly concerns how many QRREQ packets are delivered and collected in the on-demand link-state delivery/collection phase. That is, our overhead is independent of the bandwidth requirement. However, in Lin's approach, overhead is dependent on the bandwidth requirement. For instance, as shown in Fig. 14(a), the average overhead of MP1-x and MP2-x ranges from 75 to 91. However, Lin's protocol has a lower overhead. Observe that the routing packet of our approach will be re-forwarded to the destination in an intermediate node even if this node does not have enough free time slots. Under the same condition, the routing packets in Lin's protocol are dropped. This property of our protocol is helpful to reduce the overhead.



(a) (b)

Fig. 14. Routing overhead vs. (a) network size (number of routing packets = 30 and network density = 38%), (b) network density (number of routing packets = 30 and number of mobile hosts = 30).

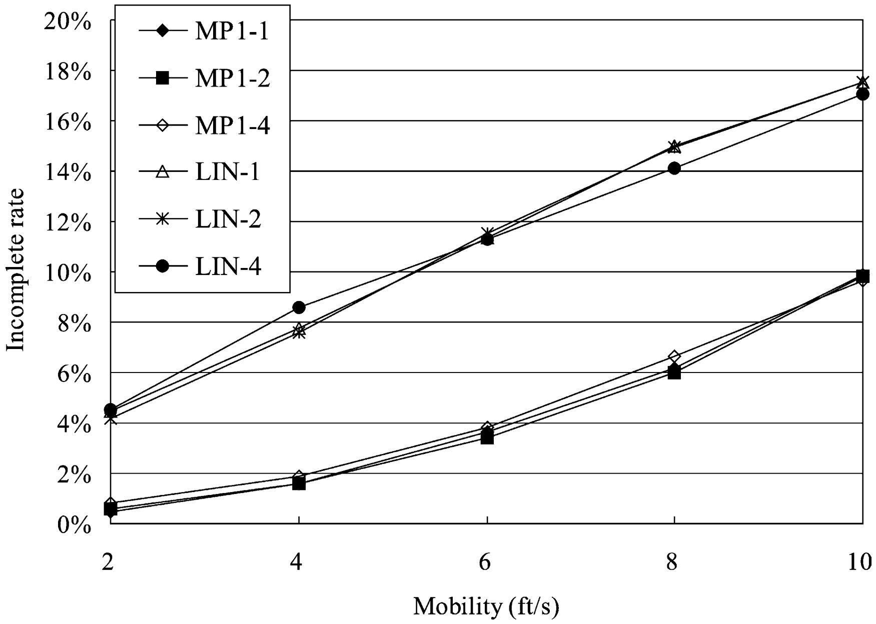


Fig. 15. Incomplete rate vs. mobility (number of mobile hosts = 30, number of routing packets = 30, and network density = 38%).

(3B) *Effect of network density*. The simulation assumption is the same for case 1A. Fig. 14(b) compares the routing overhead vs. network density. Lin's protocol has a very low overhead for high network density. The reason is that when the density is high, the success rate is very low, therefore few routing packets can reach the destination host. Many routing packets are dropped very soon after leaving the source. Therefore the overhead of Lin's approach is quite low for high network density. On the contrary, our approach still maintains a high value of overhead with high network density.

4.4. Performance of Incomplete Rate

Fig. 15 displays the performance of the incomplete rate vs. the effect of mobility.

(4A) *Effect of mobility*. Each value in Fig. 15 is obtained by assuming that the number of mobile hosts is 30, the number of routing packets is 30, and the network density is 38%. A transmission connection might not be completely finished due to link breakage. Fig. 15 shows that our MP protocol has an incomplete rate under 1% if the mobility is low, and has an incomplete rate under 10% if the mobility is high. These correspond to incomplete rates of Lin's protocol of 4–17%.

This is because the backup path scheme improves the route robustness. This verifies that our multi-path approach, which utilizes backup paths, has a lower incomplete rate.

In conclusion, our on-demand, link-state, multi-path routing approach has a higher success rate, higher slot utilization, higher overhead, and a lower incomplete rate, and outperforms existing on-demand routing protocol [10].

5. Conclusions

This paper presents a new QoS routing protocol, namely on-demand, link-state, multi-path QoS routing protocol, in a MANET. Our proposed protocol determines the QoS multi-path route at the destination by constructing a flow graph. Our protocol is developed by searching uni-path or multi-path route at the destination depending on the network bandwidth is sufficient or not. Our approach greatly improves the success rate by means of searching for the QoS multi-path routing. Performance analysis results demonstrate that our proposed protocol outperforms existing QoS routing protocol.

Acknowledgements

This work is supported by the National Science Council of the Republic of China under Grant #NSC90-2213-E-305-001, and the Ministry of Education, the Republic of China, under Grant 91A-H-FA07-1-4 and 92A-H-FA07-1-4 (Learning Technology).

References

[1] IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Institute of Electrical and Electronics Engineers, Inc.
[2] S. Chen, K. Nahrstedt, Distributed quality-of-service routing in ad hoc networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1488–1505.
[3] S. Chen, K. Nahrstedt, Y. Shavitt, A QoS-aware multicast routing protocol, Proceedings of the IEEE INFOCOM (2000) 1594–1603.

[4] Y.-S. Chen, T.-S. Chen, C.-J. Huang, SOM: spiral-fat-tree-based on-demand multicast protocol in a wireless ad-hoc network, Computer Communications 15 (17) (2002) 1684–1695.

[5] Y.-S. Chen, K.-C. Lai, MESH: multi-eye spiral-hopping protocol in a wireless ad hoc network, IEICE Transactions on Communications E84-B (8) (2001) 2237–2248.

[6] Y.-K. Ho, R.-S. Liu, On-demand QoS-based routing protocol for ad hoc mobile wireless networks, Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC) (2000) 560–565.

[7] Y.-C. Hu, D.B. Johnson, Caching strategies in on-demand routing protocols for wireless ad hoc networks, Proceedings of the MOBICOM (2000) 231–242.

[8] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, 1996, pp. 81–153, Chapter 5.

[9] W.-H. Liao, Y.-C. Tseng, J.-P. Sheu, S.-L. Wang, A multi-path QoS routing protocol in a wireless mobile ad hoc network, Telecommunication Systems 19 (3–4) (2002) 329–347.

[10] C.-R. Lin, On-demand QoS routing in multihop mobile networks, Proceedings of the IEEE INFOCOM April (2001) 1735–1744.

[11] C.-R. Lin, J.-S. Liu, QoS routing in ad hoc wireless networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1426–1438.

[12] H.C. Lin, P.C. Fung, Finding available bandwidth in multihop mobile wireless networks, Proceedings of the IEEE Vehicular Technology Conference May (2000) 15–18.

[13] A.K. Talukdar, B.R. Badrinath, Rate adaptation schemes in networks with mobile hosts, Proceedings of the MOBICOM (1998) 169–180.

[14] Y.-C. Tseng, J.-P. Sheu, C.-Y. Lin, A new multi-channel MAC protocol with on-demand channel assignment for mobile ad hoc networks, International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN) (2000) 232–237.