On-Demand, Link-State, Multi-Path QoS Routing in a Wireless Mobile Ad-Hoc Network[*]

Yuh-Shyan Chen[†], Yu-Chee Tseng[‡], Jang-Ping Sheu[¶], and Po-Hsuen Kuo[¶]
[†]Department of Statistics, National Taipei University
Taipei, 10433, Taiwan, R.O.C.
[‡]Department of Computer Science and Information Engineering
National Chao-Tung University, Hsin-Chu, Taiwan, R.O.C.
[¶]Department of Computer Science and Information Engineering
National Central University, Chung-Li, Taiwan, R.O.C.

Abstract

In this paper, we investigate an on-demand, link-state, multi-path QoS (quality-of-service) routing protocol in a MANET, which is important for many real-time multimedia applications. The basic idea of the on-demand, link-state, multi-path routing protocol is to reactively collect link-state information from source to destination in order to dynamically construct a partial network topology which is only sketched from source to destination in a MANET. Therefore, a multi-path QoS routing protocol is proposed by finding multi-path routes at the destination under the CDMA-over-TDMA channel model, which satisfies a given bandwidth requirement. Existing work mainly aims to exploit a uni-path from source to destination. The destination accurately determines a QoS multi-path routing and replies to the source host. This greatly improves the success rate by means of searching the QoS multi-path routing. Performance analysis results demonstrate that our proposed protocol outperforms other protocols.

1. Introduction

A mobile ad-hoc networks (MANET) [5] consists of wireless hosts that communicate with each other in the absence of a fixed infrastructure. In a MANET, host mobility can cause frequent unpredictable topology changes, thus the design of a MANET QoS routing protocol is more complicated than that of traditional networks. Extensive research efforts have been devoted to the design of routing protocols for MANETs [9]. However, these protocols, when searching for a route to a destination, are only concerned with shortest-path routing and the availability of multiple routes in the MANET's dynamically changing environment. Connections with quality-of-service (QoS) requirements, such as those for multimedia applications with delay and bandwidth constraints, are less frequently addressed.

Some work recently has been intensively studied QoS issues in MANETs. These problems have been addressed in several studies [2, 3, 4, 6, 7, 8]. Initially, in a quite ideal model, it is assumed that the bandwidth of a link can be determined independently of its neighboring links [2]. Under such a model, a ticket-based QoS routing protocol was proposed in [2]. Using the same model, Liao *et al.* recently proposed a QoS multi-path routing protocol [6] based on a ticket-distribution scheme. Observe that Liao *et al.*'s routing protocol presents a multi-path concept to satisfy bandwidth constraints. Unfortunately, Liao *et al.*'s scheme does not consider radio interference problems. Efforts will be made to develop a QoS multi-path routing protocol with considering the radio interference problem. Observe that, a CDMA-over-TDMA channel model was recently assumed in [7, 8] to develop a QoS routing protocol in a MANET, where the use of a time slot on a link is only dependent on the status of its one-hop neighboring links. A code assignment protocol should be supported. Therefore, Lin and Liu calculated the end-to-end path bandwidth to develop DSDV-based QoS routing [8] and on-demand QoS routing [7] in a MANET.

This paper calculates the end-to-end path bandwidth of a QoS multi-path routing following the CDMA-over-TDMA channel model, as defined in [8]. This paper presents an on-demand, link-state, multi-path QoS (quality-of-service) routing protocol in a MANET. The basic idea of the on-demand, link-state, multi-path, routing protocol is to reactively collect link-state information from source to destination. The purpose is to dynamically construct a flow network, which is a network topology sketched from source to destination in a MANET. Therefore, a multi-path QoS routing protocol is proposed by searching multi-path routes at the destination which satisfy the bandwidth requirement. The destination accurately determines a QoS multi-path routing and replies to the source host. This greatly improves the success rate by means of searching the QoS multi-path routing. Performance analysis results demonstrate that our proposed protocol outperforms other protocols.

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.

[*]This work was supported by learning technology, sponsored by the Ministry of Education, Taiwan: 90A-H-FA07-1-4.

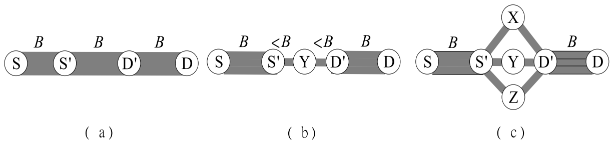

Figure 1: The multi-path approach.

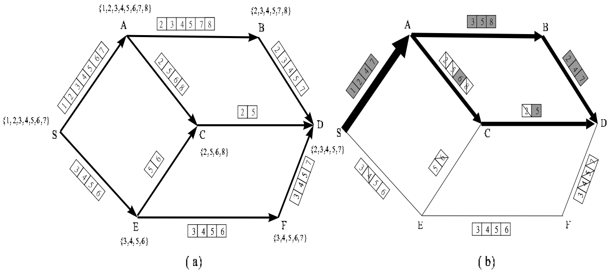

Figure 2:
Example of multi-path in CDMA-over-TDMA channel model.

2. Basic Idea and Challenges

Recently, Liao *et al.* presented a QoS multi-path routing approach [6]. The multi-path approach was developed which does not consider the radio interference problem. The challenge of our work was in attempting to design a QoS multi-path routing in a stronger model, which is the CDMA-over-TDMA channel model. We review Liao *et al.*'s multi-path approach [6]. The original bandwidth requirement is split into sub-bandwidth requirements, where each sub-path 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 $(h_1, h_2, \cdots, h_k)$ as a path from node $h_1, h_2, \cdots$, to node h_k. Given that a source node S initiates a QoS route request with a bandwidth requirement B, Fig. 1(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. 1(b) illustrates 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-state information in a MANET. However, multi-paths $(S', X, D'), (S', Y, D')$, and (S', Z, D') can be used for the QoS route request with bandwidth requirement B. We assume that multi-paths exist in the network. However, if there are no multi-paths, then the QoS fails. Fig. 1(c) shows that the bandwidth requirement B is divided into three sub-bandwidth requirements. 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.

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. The CDMA-over-TDMA channel model is assumed by following the same model as defined in [7, 8]. The CDMA (code division multiple access) is overlaid on top of the TDMA infrastructure. To overcome the hidden-terminal problem, an orthogonal code used by a host should differ from that used by any of its two-hop neighbors. A code assignment protocol should be supported. 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].

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.* [6] is obtained. The overview of our protocol is given now. 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 RREQ (Route REQuest) 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 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 $\{\alpha_1, \alpha_2, \cdots, \alpha_{\kappa-1}\}$ 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 $\{\alpha_1, \alpha_2, \cdots, \alpha_{\kappa_1}\}$ and $\{\beta_1, \beta_2, \cdots, \beta_{\kappa_2}\}$, $\kappa_1 \neq \kappa_2$, we define an intersection function $\cap(\{\alpha_1, \alpha_2, \cdots, \alpha_{\kappa_1}\}, \{\beta_1, \beta_2, \cdots, \beta_{\kappa_2}\}) = [\gamma_1, \gamma_2, \cdots, \gamma_{\kappa_3}]$, where $[\gamma_1, \gamma_2, \cdots, r_{\kappa_3}] \in \{\alpha_1, \alpha_2, \cdots, \alpha_{\kappa_1}\}$, $[\gamma_1, \gamma_2, \cdots, \gamma_{\kappa_3}] \in \{\beta_1, \beta_2, \cdots, \beta_{\kappa_2}\}$, and $\kappa_3 \leq \min\{\kappa_1, \kappa_2\}$. Let $[\gamma_1, \gamma_2, \cdots, \gamma_{\kappa_3}]$ 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 $[\gamma_1, \gamma_2, \cdots, \gamma_{\kappa_3}]$. For instance as illustrated in Fig. 2(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 illustrated in Fig. 2(b); paths (S, A, B, D) and (S, A, C, D) are estab-

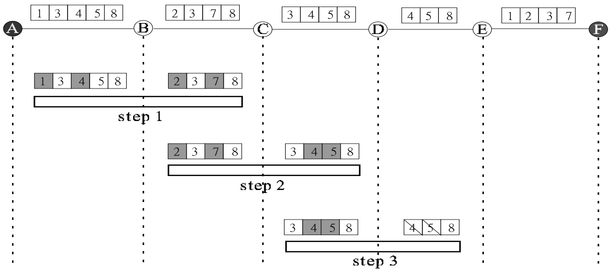

Figure 3: Example of hop-by-hop time slot reservation scheme.

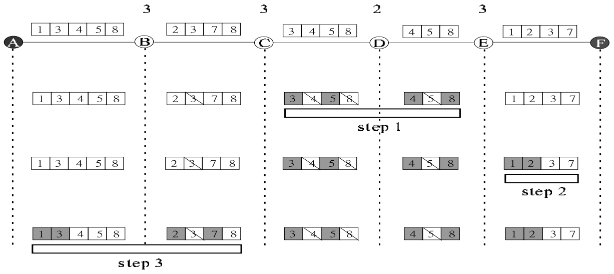

Figure 4:
Example of slot reservation by our QoS uni-path protocol.

lished if the path bandwidth requirement is four slots. We now discuss the basic idea of the uni-path routing scheme adopted in this paper. Under the CDMA-over-TDMA channel model, Lin [7, 8] 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. 3, a QoS route request, 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 requirement = two slots as shown in Fig. 4. 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 [7, 8] 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.

3. The On-Demand, Link-State, Multi-Path QoS Routing Protocol

3.1. Phase 1: On-Demand, Link-State Delivery and Collection

The source node initiates a QRREQ (QoS Route REQuest) 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. 6(a), 6(b), 6(c), 6(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. 2(a), is re-configured at destination node D.

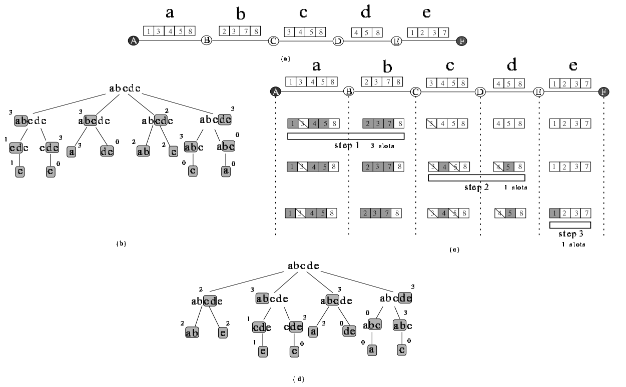

Figure 5: Example of T and T_{LCF} trees.

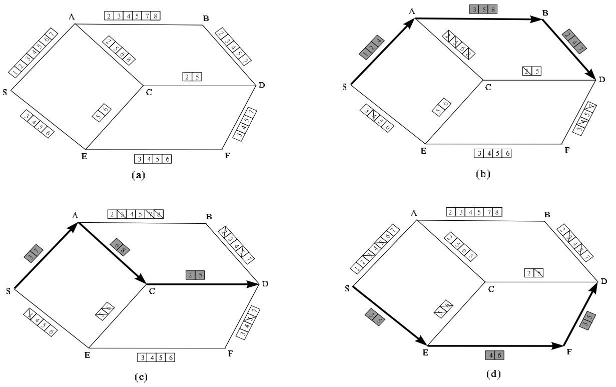

Figure 7: The original network and found multi-paths.

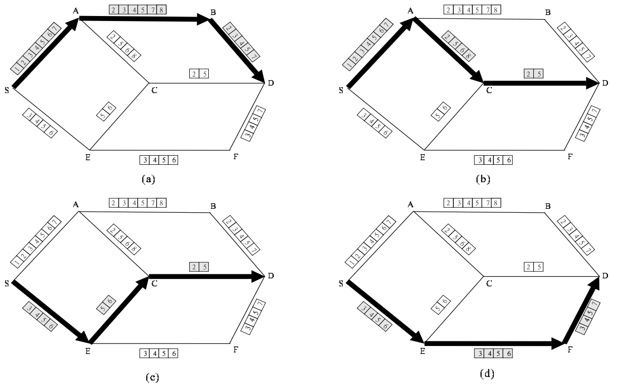

Figure 6:
On-demand, link-state delivery and collection operation.

3.2. Phase 2: Uni-Path Discovery

Observe that a network topology with link-state information is re-constructed in the destination node. 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 [7] is one of them. For example, there is a path $(A,B,C,\cdots,F)$ between source node A and destination node F. Let $a,b,c,\cdots,e$ denote free time slots of links $\overrightarrow{AB}$, $\overrightarrow{BC}$,$\overrightarrow{CD}$,$\cdots$, and $\overrightarrow{EF}$, respectively, as illustrated in Fig. 5(a). To follow the hop-by-hop reserving sequence, our reservation scheme follows the order of $\overrightarrow{AB}$, $\overrightarrow{BC}$,$\overrightarrow{CD}$,$\cdots$, and $\overrightarrow{EF}$. The example shown in Fig. 5(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. 5(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 time-slot 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. A time-slot reservation tree T is firstly constructed. 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 $\overrightarrow{AB}$, $\overrightarrow{BC}$,$\overrightarrow{CD}$,$\cdots$, and $\overrightarrow{EF}$. Observe that trees T and T_{LCF} are constructed to represent all possible conditions of time-slot reservation. Given a path $(A,B,C,D,E,\cdots Y,Z)$, let $abcd\cdots yz$ denote free time slots of links $\overrightarrow{AB}$, $\overrightarrow{BC}$,$\overrightarrow{CD}$,$\cdots$, and $\overrightarrow{YZ}$. 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 $(A,B,C,D,E,\cdots Y,\ Z)$, 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 illustrated in Fig. 5(b). As shown in Fig. 5(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 illustrated in Fig. 5(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 seacrh 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 illustrated in Fig. 5(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 illustrated in Fig. 5(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 $(A, B, C, D, E, \cdots Y, Z)$, and let $abcd \cdots yz$ denote free time slots of links $\overrightarrow{AB}$, $\overrightarrow{BC}$, $\overrightarrow{CD}$, $\cdots$, and $\overrightarrow{YZ}$. Assume that x and x' are free time lists of links $\overrightarrow{XX'}$ and $\overrightarrow{X'X''}$, and let $ab \cdots \underline{xx'} \cdots yz$ denote reserved time slots of links $\overrightarrow{XX'}$ and $\overrightarrow{X'X''}$ 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. 4.

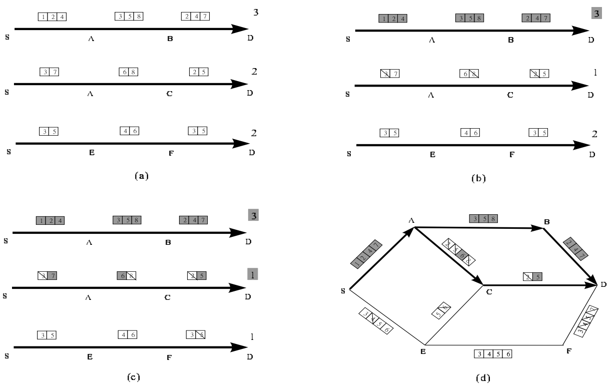

Figure 8: Example of multi-path discovery.

3.3. Phase 3: Multi-Path Discovery and Reply

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 given below.

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.

An example of searching multi-path is shown in Fig. 7 and Fig. 8.

4. Experimental Results

To examine the effectiveness of our approach, we compare our proposal with an existing on-demand QoS routing scheme, denoted as LIN, presented in [7]. Our scheme is denoted MP1 if using the constructed T_{LCF} tree and denoted MP2 if not using the constructed T_{LCF} tree. The simulation parameters are given below.

- The *number of mobile hosts* ranges from 20 to 40.

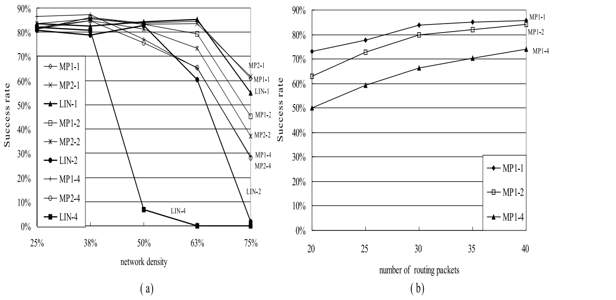

Figure 9:
Success rate vs. network density and number of routing packets.

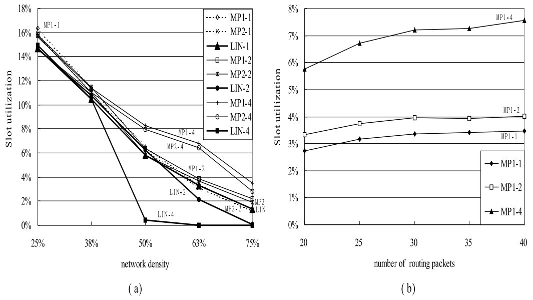

Figure 10:
Slot utilization vs. network density and number of routing packets.

- The *number of time slots* in the data phase of a frame is assumed to be 16 slots.
- Three *different bandwidth requirement* 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 *bandwidth density* is set from 25% to 75% of the network density.
- The *mobility* ranges from 2 to 10 ft/sec.
- Each simulated result is obtained by average values through 5000 runs.

The performance metrics of our simulation are given below.

- $Success\_Rate$ (SR): the number of successful QoS route requests divided by the total number of QoS route requests 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.

4.1. Performance of *Success Rate (SR)*

1A) Effect of network density: Each value in Fig. 9(a) is obtained by assuming that the number of mobile hosts is 30 and the number of routing packets is 30. Fig. 9(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. 9(a) illustrates 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 9(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. 9(b) illustrates that a high success rate is obtained when using a greater number of $QRREQ$ packets. Fig. 9(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 (SU)*

2A) Effect of network density: The simulation assumption is the same for case 1A. Fig. 10(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. 10(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. 10(b), that the higher success rate obtains better slot utilization.

4.3. Performance of *Overhead (OH)*

3A) Effect of number of mobile hosts: Each value in 11(a) is obtained by assuming that the number of mobile hosts is 30 and the network density is 38%. Fig. 11(a) shows the performance of overhead under various numbers of mobile hosts. Observe that our MP protocol has the same

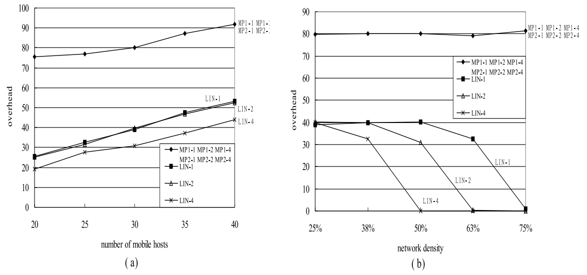

Figure 11:
Routing overhead vs. network size and network density.

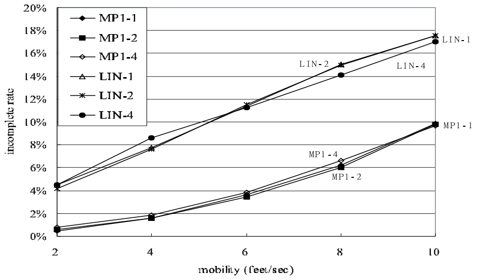

Figure 12: Incomplete rate vs. mobility.

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. 11(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.

3B) Effect of network density: The simulation assumption is the same for case 1A. Fig. 11(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 (IR)*

4A) Effect of Mobility: Each value in Fig. 12 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. 12 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.

5. Conclusions

This paper presents an on-demand, link-state, multi-path QoS routing protocol in a MANET. Our proposed protocol reactively collect link-state information from source to destination, and aims to dynamically construct a partial network topology which is only sketched from source to destination. A multi-path QoS routing protocol is developed by searching multi-path routes at the destination under the CDMA-over-TDMA channel model. Our approach greatly improves the success rate by means of searching the QoS multi-path routing. Performance analysis results demonstrate that our proposed protocol outperforms existing QoS routing protocol.

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 and K. Nahrstedt. Distributed Quality-of-Service Routing in Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, Aug. 1999.

[3] S. Chen and K. Nahrstedt. Distributed Quality-of-Service Routing in Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, Aug. 1999.

[4] Y.-K. Ho and R.-S. Liu. On-Demand QoS-based Routing Protocol for Ad Hoc Mobile Wireless Networks. In *Proc. of the Fifth IEEE Symposium on Computers and Communications (ISCC)*, pages 560–565, 2000.

[5] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5*, pages 81–153, 1996.

[6] W.-H. Liao, Y.-C. Tseng, J.-P. Sheu, and S.-L. Wang. A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network. In *Proc. of IEEE ICN'01: International Conference on Networking, Part II*, pages 158–167, July, 2001.

[7] C.-R. Lin. On-Demand QoS Routing in Multihop Mobile Networks. In *Proc. of IEEE INFOCOM 2001*, pages 1735–1744, April 2001.

[8] C.-R. Lin and J.-S. Liu. QoS Routing in Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1426–1438, Aug. 1999.

[9] S. Chen Y and K.-C. Lai. MESH: Multi-Eye Spiral-Hopping Protocol in a Wireless Ad Hoc Network. *IEICE Transactions on Communications*, E84-B(8):2237–2248, Aug. 2001.