# SMPQ: Spiral-Multi-Path QoS Routing in a Wireless Mobile Ad-Hoc Network[*]

Yuh-Shyan Chen and Tong-Ying Juang and Yu-Ting Yu
Department of Statistics, National Taipei University
Taipei, 10433, Taiwan, R.O.C.
e-mail: yschen@mail.ntpu.edu.tw

## Abstract

In this paper, we present a SMPQ: Spiral-Multi-Path QoS routing protocol in the MANET, while the MAC sub-layer is adopted the CDMA-over-TDMA channel model. This work investigates the bandwidth reservation problem of an on-demand QoS routing in a wireless mobile ad-hoc network. The proposed approach increases the robustness of a route lies in identifying a robust path, namely *spiral-multi-path*, from source host to destination host, in a MANET to satisfy certain bandwidth requirement. This paper aims to strengthen the route-robustness and increase the success rate of the QoS route. Performance analysis results demonstrates that our SMPQ protocol outperforms other protocols.

## 1. Introduction

A wireless mobile ad-hoc networks (MANET) [9] consist of wireless mobile hosts that communicate with each other, in the absence of a fixed infrastructure. A multi-hop scenario occurs in a way that packets sent by the source host are retransmitted through several intermediate hosts before reaching the destination host. In a MANET, host mobility results in frequently unpredictable topology changes. Since MANET is characterized by its fast changing topology, extensive research efforts have been devoted to the design of routing protocols for MANETs [5, 6]. These protocols, when searching for a route to the destination, only concerns with shortest-path routing and the stability of routes in the MANET's dynamically changing environment. Connections with quality-of-service (QoS) requirements, such as multimedia with bandwidth constrains, are less frequently addressed.

Some works recently started to study the QoS issue in the MANET. [3, 4, 7, 8, 10, 11, 12, 13, 14, 15]. A ticket-based QoS routing protocol is initially proposed in [2] to find a route satisfying certain bandwidth and delay constraints. The basic idea uses tickets to limit the number of route-searching packets to avoid the blind flooding operation. A QoS routing protocol, proposed by Lin and Liu, [11, 12] calculates the end-to-end path bandwidth in MANET considering the CDMA-over-TDMA channel model. Recently, a multi-path QoS routing protocol is proposed by Liao *et al.* [10]. This QoS routing protocol presents a multi-path concept to satisfy the bandwidth constraints. A bandwidth requirement is split into sub-bandwidth requirements, while each one is responsible by one of the multiple paths. This scheme obtains better success rate of finding a QoS route, especially if the MANET is in a low-bandwidth environment. Unfortunately, Liao's scheme do not consider the radio interference problem. However, existing schemes are not handle the mobility-tolerant problem well. Effort will be made to develop a QoS route with high success rate of finding a QoS route and mobility-tolerant capability.

This paper calculates the end-to-end path bandwidth under CDMA-over-TDMA channel model, under the same model assumed in [12]. The multiple access scheme in MAC sub-layer of MANET in this paper is adopted the CDMA-over-TDMA channel model. In the model, the use of a time slot on a link is only dependent of the status of its one-hop neighboring links. Recently, Chen *et al.* [6] present a robust path scheme, namely spiral-path approach, to develop on-demand spiral-path routing and multicast routing protocols [5, 6], which are not the QoS routing protocols. One other contribution of this paper is a QoS-extension spiral-path routing protocol. Our QoS-extension routing proto-

---

[*] This work was supported by NSC90-2213-E-305-001.

col surely inherits the robust path capability from spiral-path scheme [5, 6].

This paper proposes a SMPQ: Spiral-Multi-Path QoS routing protocol in the MANET. We address the bandwidth reservation problem of an on-demand QoS routing in a MANET. The proposed approach increases the robustness of a route lies in identifying a robust path, namely *spiral-multi-path*, from source host to destination host, in a MANET to satisfy certain bandwidth requirement. Performance analysis results demonstrate that our SMPQ protocol outperforms other protocols. The most important contribution of this paper is to combine the spiral-path and multi-path to design a new robust QoS routing protocol.

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

## 2. Basic Idea and Challenge

The SMPQ (Spiral-Multi-Path QoS) routing protocol is a fully distributed scheme, which aims to dynamically identify the *spiral-multi-path*, from source host to destination host, in a MANET to satisfy the bandwidth requirement. Before formally defining the spiral-multi-path, the network model is assumed as follows. The MAC sub-layer in our model is implemented by using 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 which defined in [11, 12]. The CDMA (code division multiple access) is overlaid on top of the TDMA infrastructure. Multiple sessions can share a common TDMA slot via CDMA. To overcome the hidden-terminal problem, an orthogonal code used by a host should be different from that used by any of its two-hop neighbors. A code assignment protocol should be supported (this can be regarded as an independent problem, which can be found in [11, 12]). 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. Each data phase of a frame is assumed to be partitioned into $\kappa$ time slots. A free time-slot list is defined as $F\{\alpha_1, \alpha_2, \cdots, \alpha_\kappa\}$ or denoted as $\{\alpha_1, \alpha_2, \cdots, \alpha_\kappa\}$. Let $(N_1, N_2, \cdots, N_k)$ denoted as a path from node $N_1$, $\cdots$, to node $N_k$. For instance, $F\{1,3,4,5,6\}$ represents that free time slots $1,3,4,5$ and 6.

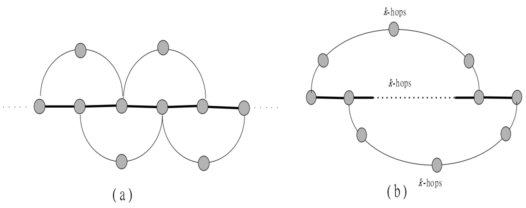


Figure 1: A spiral-path approach.

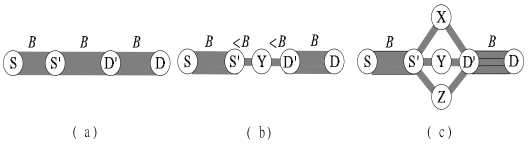


Figure 2: A multi-path approach.

Observe that our spiral-multi-path scheme is constructed by combining the spiral-path and multi-path schemes [6][10]. We initially review the spiral-path and multi-path routing protocols in [6][10]. First, a spiral-path routing is proposed by Chen *et al.* in [6]. The spiral-path is a robust routing structure, denoted as spiral-path $P_k$, as defined below.

**Definition 1** ***Spiral-path*** *[6]: Given a path P, if every node of path P connects to the next k-hop node in path P by some extra, disjoint links. Therefore, the path P and all extra links form a spiral-path $P_k$, where length of extra links is equal to k.*

For instance, a spiral-path $P_k$ is illustrated in Fig. 1(b). The route-robustness capability of the spiral-path is achieved by maintaining backup links. The more number of backup links is, the higher route robustness will be. If any link is failed, the failed link is replaced immediately by the backup link. If we consider the spiral-path $P_2$, then all extra links' length is equal to 2 as illustrated in Fig. 1(a). Observe that this paper considers the spiral-path $P_2$ using in our spiral-multi-path routing.

Secondly, a multi-path scheme is proposed by Tseng *et al.* [10]. The multi-path scheme is to split the bandwidth requirement into sub-bandwidth requirements to improve the success rate of identifying the QoS route. Fig. 2(a) gives an example if the bandwidth requirement equals to $B$ units. Fig. 2(b) shows that there is no uni-path to satisfy the bandwidth requirement from node $S$ to node $D$. Fig. 2(c) illustrates that multi-path routing is used, while each of multi-paths is responsible for small bandwidth requirement.

Multi-path approach offers a higher success rate to find a satisfactory QoS route than those protocols which try to find a uni-path. Unfortunately, the multi-path routing is not providing the mobility-tolerant capability.

The spiral-path [6] is an uni-path routing scheme, but our proposed spiral-multi-path scheme is a multi-path routing scheme. Our scheme provides the mobility-tolerant capability. The definition of spiral-multi-path, or denoted as SMP, is very similar with the definition of spiral-path $P_k$. A branch node is formally given herein before defining the spiral-multi-path. A node is said as a branch node if there exist at least two disjoint paths from a same node, then such node is a branch nodes. For instance as shown in Fig. 3, node $G$ is a branch node due to there are four disjoint paths from node $A$. Observe that, in the spiral-path [6], only uni-path, between two adjacent branch nodes, acts as primary path for transmitting data. The spiral-multi-path is formally defined below.

**Definition 2** ***Spiral-multi-path:*** *A spiral-path is said as a spiral-multi-path if the spiral-path uses multiple paths as the primary path to transmit data between two adjacent branch nodes.*

The spiral-path is a special case of spiral-multi-path if primary path from $A$ to $G$ is a uni-path. However, the spiral-multi-path is seen as a spiral-path, except for using multi-path as primary path. As illustrated in Fig. 3, two disjoint paths $(A,E,F,G)$ and $(A,D,G)$ are used for transmitting data. The design challenge builds in the differences of time slot reservation for the uni-path and the multi-path between each pair of adjacent branch nodes. Denote $\overline{XY}$ as the link between node $X$ and node $Y$. Given two adjacent branch nodes $A$ and $E$ as shown in Fig. 4(a). Time slots reserve to $\overline{AD}$ and $\overline{AC}$ are $\{1,2\}$ and $\{1,4\}$, respectively. Slot 1 is shared by $\overline{AD}$ and $\overline{AC}$, since only one link is used in the uni-path routing. Similarly, time slots reserve to $\overline{DE}$ and $\overline{CE}$ are $\{3,6\}$ and $\{6,7\}$, while slot 6 is shared by $\overline{DE}$ and $\overline{CE}$. Observe that multi-path routing does not allow same time slots reserved to different links between a pair of adjacent branch nodes. Fig. 4(b) shows that reserved time slots for $\overline{AD}$ and $\overline{AC}$ are $\{1,2\}$ and $\{4\}$. Efforts will be made to reserve time slots for the spiral-multi-path routing.

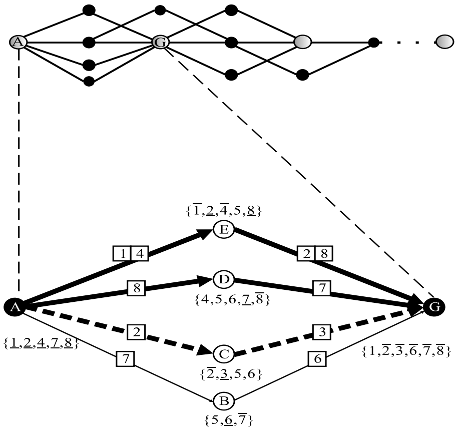


Figure 3: Example of a spiral-multi-path.

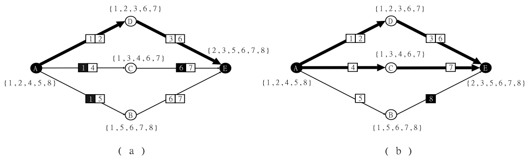


Figure 4: Time-slot reservation scheme.

## 3. Our Spiral-Multi-Path QoS Routing Protocol

### 3.1. Phase 1: Keeping Local Link-State Information

This phase aims to not only identify *branch nodes* and *branch supernodes*, but also keep local link-state information. The identification of *branch nodes* and *branch supernodes* allows us to possibly construct the spiral-multi-path. Moreover, local link-state information will be used in the route discovery phase to reserve path bandwidth for the spiral-multi-path.

Some notations are defined herein. 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 $\cap(\{\alpha_1,\alpha_2,\cdots,\alpha_{\kappa_1}\}$, $\{\beta_1, \beta_2, \cdots, \beta_{\kappa_2}\}) = (\gamma_1,\gamma_2,\cdots,r_{\kappa_3})$, where $(\gamma_1,\gamma_2,\cdots,r_{\kappa_3}) \in \{\alpha_1,\alpha_2,\cdots,\alpha_{\kappa_1}\}$, $(\gamma_1,\gamma_2,\cdots,r_{\kappa_3}) \in \{\beta_1,\beta_2,\cdots, \beta_{\kappa_2}\}$, and $\kappa_3 \leq \min\{\kappa_1, \kappa_2\}$. Let $(\gamma_1,\gamma_2,\cdots,r_{\kappa_3})$ represented as shared free time slots between nodes $A$ and $B$. This indicates that time slots of communicating between $A$ and $B$ should be selected from $(\gamma_1,\gamma_2,\cdots,r_{\kappa_3})$. For example as shown in Fig. 5(a), free time-slot list of $S$ and $A$ are $\{1,2,4,7,8\}$ and $\{1,3,4,5,6\}$, respectively, so $\cap(\{1,2,4,7,8\}, \{1,3,4,5,6\}) = (1,4)$ as illustrated in Fig. 5(b). Node $S$ communicates with node $A$ in time slots $(1,4)$.

Recalled the definition of branch node, Fig. 5(a) indicates that nodes $S$ and $C$ are branch nodes and two disjoint paths $(S,A,C)$ and $(S,B,C)$ exist between $S$ and $C$. In the following, we present how to identify the branch node in a MANET. The identification is accomplished by periodically flooding a Beacon packet within two hops. The beacon format is denoted as **Beacon**(*hopnumber*, *path_record*, $free\_slots_{path\_record[i]}$), where *hopnumber* denotes the packet lifetime by limited the number of hops, *path_record*, a path using an array to store it, records the history-path from the initiated-the-beacon node to the current node, and $free\_slots_{path\_record[i]}$ records the free time-slot list for each node $path\_record[i]$. We formally describe the identifying operation of branch node below.

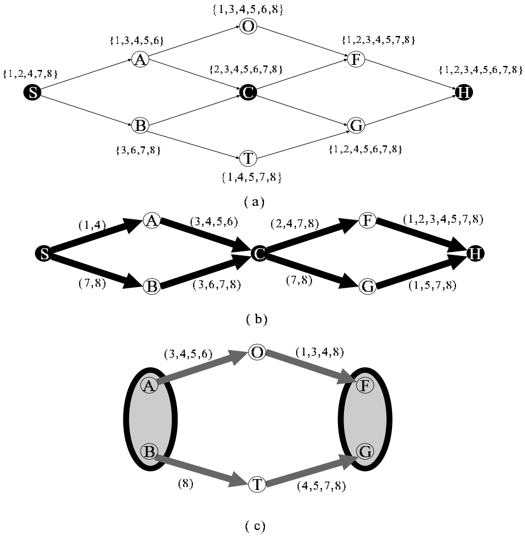


Figure 5: Identifying branch nodes/supernodes.

A1) Each node $N$ initiates and floods a **Beacon**($hopnumber--$, $path\_record[1]=[N]$, $free\_slots_N$), where $hopnumber = 3$ (or 4), records the free time-slot list into $free\_slots_N$, and re-forward the **Beacon** packet to all neighboring nodes. The flooding process is repeatedly done until $hopnumber = 0$, while *path_record* records the history-path from node $N$ to current node and $free\_slots_{path\_record[i]}$ records the free time-slot list in node $path\_record[i]$.

- For instance as shown in Fig. 5(a), if a **Beacon**$_1$ packet travels along path $(S,A,C)$, node $C$ eventually received the **Beacon**$_1$($path\_record = [S,A,C]$, $free\_slots_i$), where $free\_slots_S = \{1,2,4,7,8\}$, $free\_slots_A = \{1,3,4,5,6\}$, and $free\_slots_C = \{2,3,4,5\ ,6,7,8\}$.

A2) The shared time-slot list of a link bandwidth in path *path_record* is obtained by calculating the intersection function $\cap(free\_slots_{path\_record[i]}, free\_slots_{path\_record[]i+1})$, where $i \geq 1$.

- For instance as shown in Fig. 5(b), the shared time-slot list of link $\overline{SA}$ is $\cap(free\_slots_{path\_record[1]}, free\_slots_{path\_record[2]}) = \cap\ (free\_slots_{path\_record_S}, free\_slots_{path\_record_A}) = \cap(\{1,2,4,7,8\}, \{1,3,4,5,6\}) = (1,4)$.

A3) Let a node received two different **Beacon** packets, which denoted as **Beacon**$_1$ and **Beacon**$_2$. If $path\_record[1]$ of **Beacon**$_1$ is equal to $path\_record[1]$ of **Beacon**$_2$, then the node is a branch node.

- For instance as shown in Fig. 5(b), if node $C$ also received a second **Beacon**$_2$($path\_record = [S,B,C]$, $free\_slots_i$) from node $S$, where $free\_slots_S = \{1,2,4,7,8\}$, $free\_slots_B = \{3,6,7,8\}$, and $free\_slots_C = \{2,3,4,5\ ,6,7,8\}$. Node $C$ is a branch node since $path\_record[1]$ of **Beacon**$_1$ $= path\_record[1]$ of **Beacon**$_2$ $= S$ and $path\_record[3]$ of **Beacon**$_1$ $= path\_record[3]$ of **Beacon**$_2$ $= C$.

Given a pair of branch nods, a node is called as a gateway node if the node can communicate with both of the two branch nodes. In the following, we formally define the supernode and the branch supernode.

**Definition 3** ***Supernode:*** *All of the gateway nodes, between a pair of branch nodes, is logically said as a supernode. Let* $< \alpha_1, \alpha_2, \cdots, \alpha_{\kappa_1} >$ *denote as a supernode contains nodes* $\alpha_1, \alpha_2, \cdots, \alpha_{\kappa_1}$.

For instance in Fig. 5(b), nodes $A$ and $B$ are gateway nodes, and Fig. 5(c) illustrates that gateway nodes $A$ and $B$ forms a supernode $S =< A,B >$, and supernode $S' =< F,G >$.

**Definition 4** ***Branch supernode:*** *A supernode is said as a branch supernode if there at least exist two disjoint paths from a same supernode.*

For instance as shown in Fig. 5(c), supernode $S' =< F,G >$ is a branch supernode since there are two different paths from supernode $S = < A,B >$. Similarly, supernode $S$ is also a branch supernode. We now describe how to identify the branch supernodes and its corresponding link-state information as follows.

B1) If a branch node $E$ received **Beacon** packets from two neighboring branch nodes $E'$ and $E''$, two sets of nodes are formed. One supernode, denoted as $\alpha$, are gateway nodes between nodes $E$ and $E'$. Another supernode, denoted as $\beta$, consists of all gateway nodes between nodes $E$ and $E''$. Each node is a supernode must know all other node is the same supernode.

  - For instance as illustrated in Fig. 5(c), node $C$ are branch node, nodes $S$ and $H$ are neighboring branch nodes of node $C$. Two supernodes are $< A,B >$ and $< F,G >$, respectively.

B2) Each node $A$ in supernode periodically floods a **Beacon**(*hopnumber*$--$, *path_record*$[1] = [A]$, *free_slots*$_N$) packet to other neighboring supernode $\beta$, where $A \in$ supernode $\alpha$.

  - For instance as shown in Fig. 5(c), node $A, \in$ supernode $< A,B >$, sends a **Beacon** packet through node $O$ to node $F$ which $\in$ supernode $< F,G >$. Node $F$ received **Beacon**(*path_record* $= [A,O,F]$, *free_slots*$_i$) from node $S$, where $free\_slots_A = \{1,3,4,5,6\}$, $free\_slots_O = \{1,3,4,5,6,8\}$, and $free\_slots_F = \{1,2,3,4,5,7,8\}$.

B3) This step is same as A2 step.

  - For instance as shown in Fig. 5(c), free time-slot list of link $\overline{AO}$ is $\cap(free\_slots_{path\_record[1]}, free\_slots_{path\_record[2]}) = \cap(free\_slots_{path\_record_A}, free\_slots_{path\_record_O}) = \cap(\{1,3,4,5,6\}, \{1,3,4,5, 6, 8\}) = (3,4,5,6)$.

B4) If a supernode received at least two **Beacon** packets, which sent from a same supernode, then such a supernode is a branch supernode.

  - For instance as shown in Fig. 5(c), two **Beacon** packets travelled along path $(A,O,F)$ and $(B,T,G)$, since nodes $A$ and $B \in$ supernode $< A,B >$ and nodes $F$ and $G \in$ supernode $< F,G >$, therefore both supernodes $< A,B >$ and $< F,G >$ are branch supernodes.

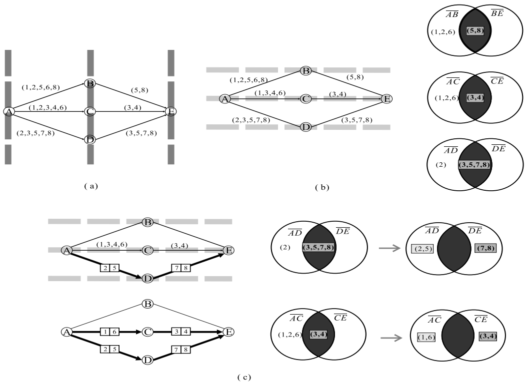


Figure 6:
A basic sub-path bandwidth reservation opertion.

### 3.2. Phase 2: Route-Discovery Phase

This phase discusses the discovery phase of our SMPQ routing protocol. The discovery phase builds by a sub-path bandwidth reservation operation. This operation is repeatedly applied on the each pair of branch nodes and then applied on each pair of branch supernodes. Our purpose is to pre-reserve time slots during constructing the spiral-multi-path. The time-slot allocation problem is a NP-complete problem [10]. This phase presents a heuristic algorithm to efficiently overcome the problem. A sub-path bandwidth reservation operation is given herein.

**A Sub-Path Bandwidth Reservation Operation** is performed between two adjacent branch nodes $B$ and $B'$. Assume that there are $\kappa$ disjoint paths between $B$ and $B'$, where each sub-path length is equal to 2. If the bandwidth requirement is $\gamma$, we reserve time slots on $\alpha$ disjoint paths, where $\alpha \leq \kappa$, such that the total path bandwidth is equal to $\gamma$. A priority value for each path is determined, so that we reserve time slots on these $\alpha$ disjoint paths, $\alpha \leq \kappa$, in order by the priority value.

C1) Each path calculates its own maximum sub-path bandwidth. The maximum sub-path bandwidth is obtained by calculating the maximum sub-path bandwidth without considering radio interference with other paths. A path with high priority value if this path has high maximum sub-path bandwidth.

  - For example as shown in Fig. 6(a), there are three paths $(A,B,E),(A,C,E)$, and $(A,D,E)$ between branch nodes $A$ and $E$. The maximum sub-path bandwidth of paths

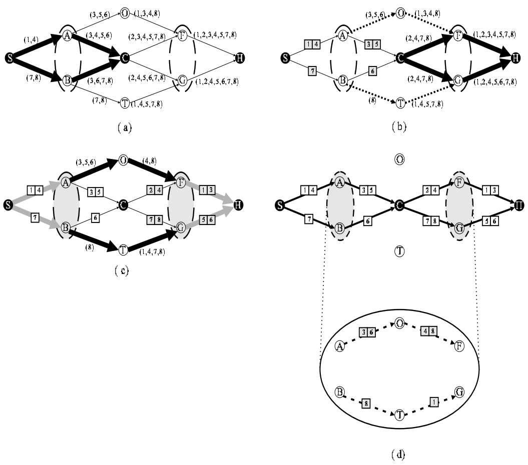


Figure 7: Route-discovery operation.

$(A,B,E),(A,C,E)$, and $(A,D,E)$ is 2, 2, and 2, three paths have same priority value.

C2) If there are $\chi \leq \kappa$ disjoint paths have the same maximum sub-path bandwidth. We still need to determine the priority value for $\chi \leq \kappa$ disjoint paths. We adopt a set-intersection operation on links' link bandwidth of $\chi$ disjoint paths to determine the priority value. If $(X,Y,Z)$ denotes a path of $\chi \leq \kappa$ disjoint paths, then $\cap(\overline{XY}, \overline{YZ})$ is calculated. Each set-intersection operation produces intersectant elements. The more number of intersectant elements is, the more free time-slots will have. A path with high priority value if this path has more number of intersectant elements.

- For instance as shown in Fig. 6(b), there are three disjoint paths $(A,B,E)$, $(A,C,E)$, and $(A,D,E)$. Let $\cap(\overline{AB}, \overline{BE})$ denote the set-intersection operation on links' bandwidth of links $\overline{AB}$ and $\overline{BE}$. Therefore, $\cap(\overline{AB}, \overline{BE}) = \cap((1,2,5,6,8), (5,8)) = (5,8)$, $\cap(\overline{AC}, \overline{CE}) = \cap((1,3,4,6), (3,4)) = (3,4)$, and $\cap(\overline{AD}, \overline{DE}) = \cap((2,3,5,7,8), (3,5,7,8)) = (3,5,7,8)$. The priority value of multi-path $(A,D,E)$ is larger than $(A,C,E)$ and $(A,B,E)$, since $|(3,5,7,8)| > |(3,4)|$ and $|(5,8)|$. Finally, a possible priority sequence is $(A,D,E),(A,C,E)$, $(A,B,E)$.

C3) After a priority sequence is determined by C1 and C2 steps, we apply the bandwidth reservation operation for paths between branch nodes. We repeatedly reserve time slots until the total path bandwidths are equal or larger than the original bandwidth requirement $\gamma$.

- For instance as shown in Fig. 6(c), if bandwidth requirement is 4 slots, we first reserve $(2,5)$ and $(7,8)$ to path $(A,D,E)$. We then reserve $(1,6)$ and $(3,4)$ to path $(A,C,E)$, such that the total path bandwidths are 4 slots.

We now present the route-discovery operation by repeatedly performing the basic time-slot reservation operation. One important notation is defined herein, let $[\overline{\alpha_1}, \underline{\beta_1}, \overline{\alpha_2}, \underline{\beta_2}, \cdots \overline{\alpha_\kappa}, \underline{\beta_\kappa}]$ denote as a feasible path with its path bandwidth from source to destination, where $\overline{\alpha_i}$ denotes a link bandwidth between two branch nodes and $\underline{\beta_i}$ represents as a link bandwidth between two branch supernodes. For instance, a path bandwidth $[\overline{3}, \underline{3}, \overline{4}]$ from node $S$ to node $H$ is given in Fig. 7. Destination eventually acquires many paths with corresponding path bandwidth, a final path with suitable path bandwidth will be determined based on the bandwidth requirement and mobility-tolerant capability, which will be described in the QoS route-reply phase. We now formally describe how to construct a possible spiral-multi-path and its path bandwidth as follow.

D1) Source node initiates and floods a QoS route-discovery packet and performs a basic time-slot reservation operation between source node to all possible branch nodes, where each path can records link bandwidth $[\overline{\alpha_1}]$ in its route-discovery packet.

- For instance as shown in Fig. 7(a), $[\overline{3}]$ is obtained from node $S$ to node $C$ through multi-paths $(S,A,C)$ and $(S,B,C)$, where $(S,A,C)$ with two time-slots and $(S,B,C)$ with one time-slot.

D2) A branch node, which received route-discovery packet with record $[\overline{\alpha_1}]$, continue to perform a basic time-slot reservation operation between next possible neighboring branch node, and the path bandwidth becomes $[\overline{\alpha_1}, \overline{\alpha_2}]$. Notably, some free time-slots must be modified before performing the basic time-slot reservation operation. This is because that these time-slots have been allocated in D1 step.

- For instance as shown in Fig. 7(b), consider multi-paths $(C,F,H)$ and $(C,G,H)$, note that the link bandwidth of $\overline{CF}$ is changing from $(2,3,4,5,7,8)$ to $(2,4,7,8)$ since

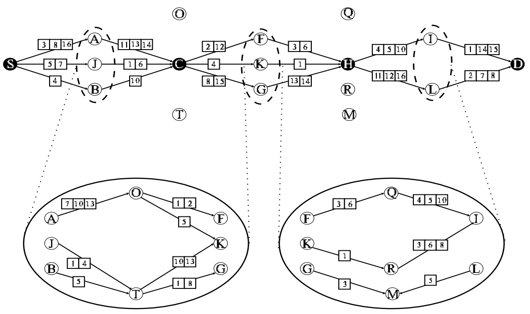


Figure 8: A spiral-multi-path with $[\overline{6},\underline{6},\overline{5},\underline{5},\overline{6}]$.

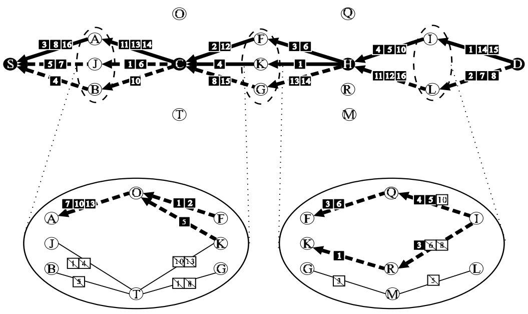


Figure 9: QoS route-reply operation.

$(3,5)$ is already allocated to link $\overline{AC}$, the link bandwidth of $\overline{CG}$ is changing from $(2,4,5,6,7,8)$ to $(2,4,7,8)$ since $(5)$ and $(6)$ are already allocated to multi-paths $(S,A,C)$ and $(S,B,C)$. Therefore, $[\overline{3},\overline{4}]$ is obtained.

D3) Two supernodes $S$ and $S'$ based on D1 and D2 steps are formed, a basic time-slot reservation operation is similarly to perform between these two supernodes. Surely, some free time-slots must be modified before performing the basic time-slot reservation operation. This is because that these time-slots have been allocated in C1 and C2 steps. Therefore, we let the path bandwidth becomes $[\overline{\alpha_1},\underline{\beta_1},\overline{\alpha_2}]$. It is noting that each node $e$ in supernode $S$ must possibly find multi-paths (or uni-path) to any node in supernode $S'$ such that the total sub-path bandwidths of the multi-paths are equal to the sub-bandwidth requirement of node $e$.

- For instance as shown in Fig. 7(c), a path bandwidth $[\overline{3},\underline{3},\overline{4}]$ is constructed. Other instance is given in Fig 3.5, node $A$ is in supernode $< A,J,B >$ and other supernode is $< F,K,G >$, sub-bandwidth requirement of node $A$ is 3 since time slots $(3,8,16)$ is reserved to link $\overline{SA}$. Therefore, node $A$ finds multi-paths $(A,O,F)$ and $(A,O,K)$, and the total sub-path bandwidth is equal to 3. Note that since links $\overline{FH}$ and $\overline{KH}'s$ link bandwidth are 2, so we also require both of links $\overline{OF}$ and $\overline{OK}'s$ link bandwidth are equal to 2. This is because that the final time-slot reservation must be exactly confirmed in the route-reply phase.

D4) Repeatedly execute the D2 and D3 steps until arriving to the destination, a path with path bandwidth $[\overline{\alpha_1}, \underline{\beta_1},\overline{\alpha_2}, \underline{\beta_2},\cdots\overline{\alpha_\kappa}, \underline{\beta_\kappa}]$ eventually is obtained in destination.

Intuitively, many paths, each one containing spiral-multi-paths, with their path bandwidth $[\overline{\alpha_1}, \underline{\beta_1},\overline{\alpha_2}, \underline{\beta_2},\cdots,\underline{\beta_{\kappa-1}}, \overline{\alpha_\kappa}]_i$, $i \geq 1$, can be obtained in the destination node. For other instance, a path, from $S$ to $D$, with its path bandwidth $[\overline{6},\underline{6},\overline{5},\underline{5},\overline{6}]$ is illustrated in Fig. 8. A simple selection criterion will be given in QoS route-reply phase to choose a suitable spiral-multi-path as our final one.

### 3.3. Phase 3: Route-Reply Phase

Assume that there are many paths, each one with its path bandwidth $[\overline{\alpha_1}, \underline{\beta_1},\overline{\alpha_2}, \underline{\beta_2},\cdots,\underline{\beta_{\kappa-1}}, \overline{\alpha_\kappa}]_i, i \geq 1$, is obtained in the destination node. Before replying packet to source, destination must to select a final spiral-multi-path and reserve the time-slots for the final one and release the time-slots for the unused spiral-multi-paths.

Consider that $[\overline{\alpha_1}, \underline{\beta_1},\overline{\alpha_2}, \underline{\beta_2},\cdots,\underline{\beta_{\kappa-1}},\overline{\alpha_\kappa}]_i, i \geq 1$, represented as a spiral-multi-path. Given the bandwidth requirement from source is $\gamma$, the selecting-path criteria of the final spiral-multi-path are given herein.

E1) **Mobility-tolerant capability for gateway node:** A spiral-multi-path satisfies condition,

$$[\overline{\alpha_j}]_i \geq \gamma, \text{for all } 1 \leq j \leq \kappa.$$

E2) **Mobility-tolerant capability for branch node:** A spiral-multi-path satisfies condition,

$$[\underline{\beta_j}]_i \geq \gamma, \text{for all } 1 \leq j \leq \kappa - 1.$$

Our spiral-multi-path has to at least satisfy the E1 condition such that a QoS route is found which possibly contains the on-line recovery capability for the gateway node. There is the on-line recovery capability for any branch node if the spiral-multi-path only satisfies the E1 condition. Notably, all backup paths in $[\underline{\beta_j}]_i$ are used to recover the failed node, which is a branch node. It is possible to find a path without such on-line recovery capability for branch node. For instance, given $\gamma = 3$, a path with $[\overline{6}, \underline{0}, \overline{5}, \underline{0}, \overline{6}]$ satisfies the E1 condition, but a path with $[\overline{6}, \underline{6}, \overline{2}, \underline{5}, \overline{6}]$ does not satisfy the E1 condition. Further, if we hope our spiral-multi-path with on-line recovery capability for gateway/branch node, the spiral-multi-path must both satisfies E1 and E2 conditions. For instance, given $\gamma = 3$, a path with $[\overline{6}, \underline{6}, \overline{5}, \underline{5}, \overline{6}]$ satisfies the E1 and E2 conditions. Given a spiral-multi-path with its path bandwidth $[\overline{\alpha_1}, \underline{\beta_1}, \overline{\alpha_2}, \underline{\beta_2}, \cdots, \underline{\beta_{\kappa-1}}, \overline{\alpha_\kappa}]_i, i \geq 1$, an average extra bandwidth or $EB_{av}$ of the spiral-multi-path is calculated by

$$EB_{av} = \overline{EB}_{av} + \underline{EB}_{av} = \frac{\sum_{i=1}^{\kappa} |\overline{\alpha_i} - \gamma|}{\kappa} + \frac{\sum_{i=1}^{\kappa-1} |\underline{\beta_i} - \gamma|}{\kappa - 1}$$

, where $\overline{EB}_{av}$ denotes the average extra bandwidth for every pair of branch nodes, and $\underline{EB}_{av}$ represents the average extra bandwidth for every pair of branch supernodes. Recall above instance, a path with $[\overline{6}, \underline{6}, \overline{6}, \underline{5}, \overline{6}]$ satisfies both the E1 and E2 conditions, and its $EB_{av} = \overline{EB}_{av} + \underline{EB}_{av} = \frac{|6-3|+|5-3|+|6-3|}{3} + \frac{|6-3|+|5-3|}{2} = 5.16$. Notice that the $EB_{av}$ is used to make a selecting-path decision if there are many spiral-multi-paths which satisfy both E1 and E2 conditions. We observe that the value of $EB_{av}$ is larger, the high QoS route stability will be. To maintain the QoS route stability, a spiral-multi-path with high $EB_{av}$ is elected as our final spiral-multi-path. Specially, a spiral-multi-path is without any mobility-tolerant capability if $\overline{EB}_{av} = \underline{EB}_{av} = 0$. We will further discuss the impact of performance under different ratio of $\overline{EB}_{av}$ and $\underline{EB}_{av}$ in our simulation.

A simple route reply phase is to reply a route-reply packet from destination to source, to reserve all time-slots in selected spiral-multi-path with path bandwidth $[\overline{\alpha_1}, \underline{\beta_1}, \overline{\alpha_2}, \underline{\beta_2}, \cdots, \underline{\beta_{\kappa-1}}, \overline{\alpha_\kappa}]_j$,where $1 \leq j \leq i$. Additionally, destination also replies packets to all other spiral-multi-paths to release their bandwidth resource. For each spiral-multi-path with path bandwidth $[\overline{\alpha_1}, \underline{\beta_1}, \overline{\alpha_2}, \underline{\beta_2}, \cdots, \underline{\beta_{\kappa-1}}, \overline{\alpha_\kappa}]_j$,where $1 \leq j \leq i$, just only need to reply a reply packet to reserve the path bandwidth $[\overline{\gamma}, \underline{\gamma}, \overline{\gamma}, \cdots, \underline{\gamma}, \overline{\gamma}]$ if the bandwidth requirement is $\gamma$. However, the extra link bandwidth $|\overline{\alpha}_x - \gamma|$ or $|\underline{\beta_x} - \gamma|$ is reserved to provide the mobility-tolerant capability. After a final spiral-multi-path with path bandwidth $[\overline{\alpha_1}, \underline{\beta_1}, \overline{\alpha_2}, \underline{\beta_2}, \cdots \overline{\alpha_{\kappa-1}}, \underline{\beta_{\kappa-1}}, \overline{\alpha_\kappa}]$ is determined. The reply packet is replied according to following rules.

F1) Destination initiates a reply packet to a pre-branch node, if there is a uni-path which its pre-reserved time-slot number is at least larger than $\gamma$, then let the uni-path as the primary path by reserving $\gamma$ time slots. All other possible multi-paths are backup paths. If there is no any uni-path which pre-reserved time-slot number is smaller than $\gamma$, then multi-paths are used as the primary paths. The more number of pre-reserved time-slot path will be reserved firstly, which aims to use less number of multi-paths. Observe that a path with the less number of pre-reserved time slots will be chosen as last one of our multi-primary-paths. This is because that we can keep other path with more time-slots to be the backup path.

- For instance as shown in Fig. 9 if a path bandwidth $[\overline{6}, \underline{6}, \overline{5}, \underline{5}, \overline{6}]$ is determined, destination node $D$ sends a reply packet to a pre-branch node $H$, there is a unit-path $(D, I, H)$ with 3 time-slots. Other path $(D, L, H)$ with 3 time-slots is a backup path.

F2) The branch node continually re-forward the reply packet to the next pre-branch node, and then performs the similar operation described in F1 step.

- For instance as shown in Fig. 9, branch node $H$ replies the reply packet to branch node $C$, there is no any path with 3 time-slots. Therefore, multi-paths $(H, F, C)$ and $(H, K, C)$ are used in responsible for the 3 time-slots. However, path $(H, G, C)$ with 2 time slots is also reserved as a backup path. Observe that we do choose the path $(H, K, C)$ as one of primary paths, and let $(H, G, C)$ be a backup path. This can achieve the good mobility-tolerant ability than if we use path $(H, G, C)$ as primary path.

F3) Two supernodes $S$ and $S'$ based on F1 and F2 steps are formed. Every node in primary paths of supernode $S$ must try to connect to nodes in primary paths of supernode $S'$ under the bandwidth requirement $\gamma$. Note that all multi-paths in D3 step are belong to backup paths.

- For instance as shown in Fig. 9, node $I \in < I, L >$ is in primary path must connect

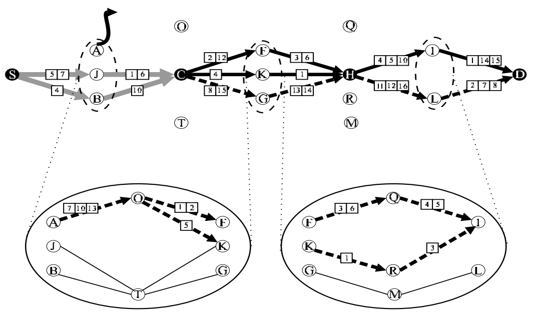


Figure 10: Tolerating failed gateway node.

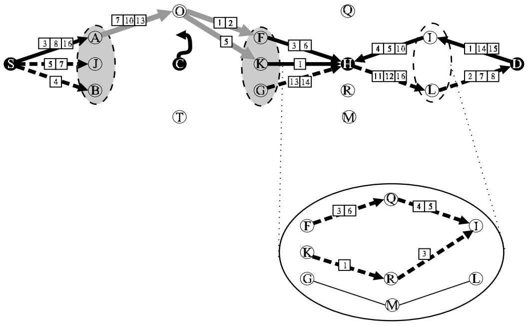


Figure 11: Tolerating failed branch node.

to node $F$ and $K$, which are in primary paths of supernode $< F,K,G >$. Multi-paths $(I,Q,F)$ and $(I,R,K)$ with 2 and 1 time slots are possibly reserved as the backup paths.

F4) Repeatedly execute the F2 and F3 steps until arriving to the source, a path with path bandwidth $[\overline{\alpha_1},\ \underline{\beta_1}, \overline{\alpha_2},\ \underline{\beta_2}, \cdots \overline{\alpha_\kappa},\ \underline{\beta_\kappa}]$ eventually is constructed.

Eventually, a spiral-multi-path is constructed and reserved the path bandwidth $[\overline{\alpha_1},\ \underline{\beta_1}, \overline{\alpha_2},\ \underline{\beta_2}, \cdots \overline{\alpha_\kappa},\ \underline{\beta_\kappa}]$ under the bandwidth requirement $\gamma$,. A fully example is given in Fig. 9 to reserve a spiral-multi-path with path bandwidth $[\overline{6},\underline{6},\overline{5},\underline{5},\overline{6}]$.

### 3.4. Phase 4: Route-Maintenance Phase

Our main contribution of proposed protocol is to provide the on-line route-recovery capability. A node is said to be a failed node if this node is moving out of the original transmission radius or is really failed. We shall show how to achieve the on-line recovery capability such that QoS route can be continually performed. The on-line recovery capability of our proposed protocol is achieved by a multi-path and spiral-path replacement strategy. The proposed replacement strategy is given according to the different roles of a failed node.

G1) If the failed node is a gateway node, all backup multi-paths between branch nodes are used to replace with the failed path. Here we use an ACK. packet between branch nodes to make sure that the data message has been successfully received by the downstream branch node. That is, if an upstream branch node waits for a period of time and still does not receive an ACK. from the downstream branch node, then it detects the failed path. And it will initiate the backup multi-paths to replace the failed path.

- For instance as shown in Fig. 10, if node $A$ is a failed node and the path $(S,A,C)$, which sub-path bandwidth = 3, is broken, then the backup paths $(S,J,C)$ and $(S,B,C)$ with total sub-path bandwidth = 3, are used to replace with the failed path.

G2) If the failed node is a branch node, all backup spiral-multi-paths between branch supernodes are used to replace with the failed paths.

- For instance, as shown in Fig. 11, two supernodes are $< A,J,B >$ and $< F,K,G >$. If branch node $C$ is a failed node and the multi-paths $(A,C,F)$ and $(A,C,K)$, whose path bandwidth = 3, are broken, the backup spiral-multi-paths $(A,O,F)$ and $(A,O,K)$, whose path bandwidth = 3, are used immediately to replace with the failure paths and to keep the bandwidth requirement.

## 4. Experimental Results

To illustrate the efficiency of our proposed scheme and compare to Lin scheme [11], a simulator is developed. The simulator is simulated in a 1000×1000 $m^2$ area. The radio transmission range is 400 m. Data rate is 4 Mbit/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. Source and destination are selected randomly. Once a QoS request is successful, a time slots is reserved for all the subsequent packets. The reservation is released when either the data transmission process is finished or the link is broken. The total simulation time is $10^6$ ms. The simulation parameters in the simulator are given below.

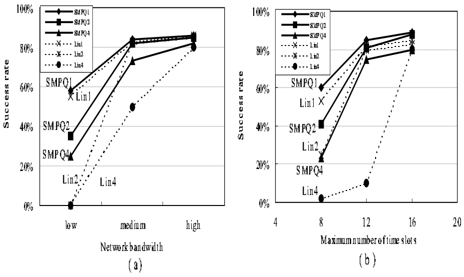


Figure 12: Performance of success rate.

- The *number of mobile hosts* is ranging from 20 to 40.
- The *maximum number of time slots* is ranging from 8, 12 to 16.
- Three *different bandwidth requirement* are 1, 2 and 4 slots, which denoted as Lin-*x* for Lin scheme [11], and SMPQ-*x* of our proposed scheme, where $x = 1, 2,$ and 4.
- The *network bandwidth* are denoted as low, medium, and high, where assuming 25%, 50%, 75% of free time slots as the low, medium, and high network bandwidths.

The mean values of inter-arrival time for SMPQ-*x* and Lin-*x* schemes, where *x*=1, 2, and 4, are 100ms, 50ms, and 25ms. A packet is dropped if the packet stays in a node exceeds the maximal queuing delay time, which is setting to four frame lengths (328 ms). Additionally, performance metrics of our simulation are defined 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 number of packets used for constructing and maintaining the QoS route from source to destination.

### 4.1. Performance of $Success\ Rate$ $(SR)$

The observed results of our SMPQ and Lin schemes are shown in Fig. 12 to reflect the performance of success rate vs. network bandwidth and number of time slots. Two types of effects are discussed.

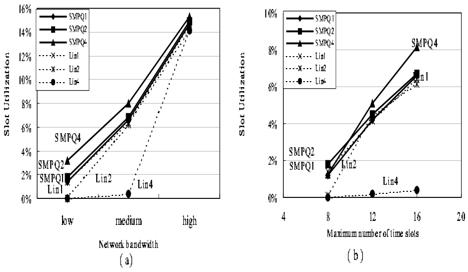


Figure 13: Performance of slot utilization.

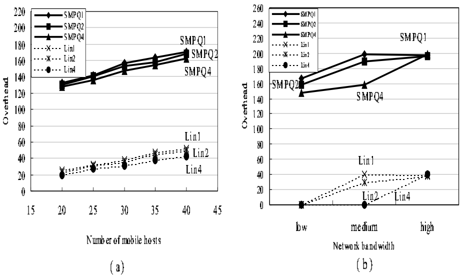
1a) *Effects of Network Bandwidth*: Each value in Fig. 12(a) is obtained by assuming the number of time slots is 16 and the number of mobile hosts is 30. From Fig. 12(a), we observe that in the low network bandwidth, our SMPQ obtains higher *SR* than the Lin's. For example, under the low network bandwidth, *SR* values of SMPQ-1, SMPQ-2, SMPQ-4 are 58%, 35%, and 25%, and *SR* values of Lin-1, Lin-2 and Lin-4 are 56%, 0%, and 0%. This indicates that our spiral-multi-path scheme is more easily finding a QoS route than Lin's scheme under a MANET with low network bandwidth. Basically, the higher network bandwidth can acquire the more higher value of *SR*. Observe that our scheme and Lin one acquire same values of *SR* in a MANET with high network bandwidth.

1b) *Effects of Maximum Number of Time Slots*: Every value in Fig. 12(b) is obtained by assuming the number of mobile hosts is 30 and the network bandwidth is medium. Fig. 12(b) shows that our scheme has higher *SR* than the Lin one under various maximum number of time slots. It is observed that *SR* values of SMPQ-*x* is always larger than *SR* value of Lin-*x*, where *x*=$\{1,2,4\}$. For instance, if maximum number of time slots is 12, then *SR* values of SMPQ-1, SMPQ-2, SMPQ-4 and Lin-1, Lin-2, Lin-4 are $84\%, 81\%, 72\%$ and $80\%, 79\%, 15\%$.

### 4.2. Performance of $Slot\ Utilization$ $(SU)$

The observed results of performance of slot utilization vs. network bandwidth and number of time slots are given in Fig. 13. Two types of effects are given below.

2a) *Effects of Network Bandwidth*: The simulation assumption is the same as the 1a)



(a) (b)

Figure 14: Performance of overhead.

case. A higher *SU* implies that a better scheme will be. Fig. 13(a) shows that our scheme has better *SU* than the Lin's in the various network bandwidth under different bandwidth requirement. In a MANET with low network bandwidth, when the bandwidth requirement increases, Lin's *SU* values decrease and our *SU* values increase, since our scheme can acquire better *SR* values in the low network bandwidth. For instance, if in a low network bandwidth, then *SU* values of SMPQ-1, SMPQ-2, SMPQ-4 and Lin-1, Lin-2, Lin-4 are $6.8\%, 7\%, 8\%$ and $6.4\%, 6.2\%, 0.6\%$.

2b) *Effects of Maximum Number of Time Slots*: The simulation assumption is the same as the 1b) case. Fig. 13(b) shows that our scheme obtains higher slot utilization than the Lin's under various maximum number of time slots. Observe that the *SU* is very low for Lin-4 as shown in Fig. 13(b), because that it's not easy for the Lin's scheme to find a QoS route if bandwidth requirement is 4. For instance, if maximum number of time slots is 12, then *SU* values of SMPQ-1, SMPQ-2, SMPQ-4 and Lin-1, Lin-2, Lin-4 are $4.1\%, 4.3\%, 4.7\%$ and $4.23\%, 4.2\%, 0.12\%$.

### 4.3. Performance of $OverHead$ $(OH)$

The observed results of the performance of overhead vs. number of mobile hosts and network bandwidth are illustrated in Fig. 14. Two kinds of effects are illustrated.

3a) *Effects of Number of Mobile Hosts*: Each value in Fig .14(a) is obtained by assuming the number of time slots is 16 and the network bandwidth is medium. Since our scheme must to collect time slot information for two- and three-hop neighbors, extra control packets are needed. Therefore, our scheme has more packets than Lin scheme. Although a lower *OH* implies that a better scheme will be, our scheme has high *OH* value than Lin's one. Fig. 14(a) shows that our scheme's overhead is about four times than Lin scheme. However, this is cost of our scheme. For instance, if number of mobile host is 30, then *OH* values of SMPQ-1, SMPQ-2, SMPQ-4 and Lin-1, Lin-2, Lin-4 are $162, 158, 145$ and $40, 38, 32$.

3b) *Effects of Network Bandwidth*: The simulation assumption is the same as the 1a) case. Fig. 14(b) illustrates that the *OH* value of our protocol is also dependent on the conditions of $Network\ Bandwidth$. Fig. 14(b) shows that our scheme has higher *OH* than the Lin's scheme in various network bandwidth. Observe that the *OH* values will be same if in the high network bandwidth as illustrated in Fig. 14(b). For instance, if both schemes are in high network bandwidth, then *OH* values of SMPQ-$x$ and Lin-$x$ are about 200 and 40, where $x = 1, 2, 4$.

## 5. Conclusions

In this paper, we present an efficient on-demand QoS routing protocol, namely SMPQ: Spiral-Multi-Path QoS routing protocol, in a wireless mobile ad-hoc network. The MAC sub-layer in our model is adopted the CDMA-over-TDMA scheme. This protocol aims to dynamically identify the *spiral-multi-path*, from source host to destination host to satisfy certain bandwidth requirement. Performance analysis results demonstrates that our SMPQ protocol outperforms other protocols.

## References

[1] IEEE Std 802.11-1997. 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] Shigang Chen and Klara Nahrstedt. Distributed Quality-of-Service Routing in Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, August 1999.

[4] Shigang Chen, Klara Nahrstedt, and Yuval Shavitt. A QoS-Aware Multicast Routing Protocol. In *Proc. of IEEE INFOCOM*, pages 1594–1603, 2000.

[5] Yuh-Shyan Chen, Tzung-Shi Chen, and Ching-Jang Huang. SOM: Spiral-Fat-Tree-Based On-Demand Multicast Protocol in a Wireless Ad-Hoc Network. In *Proc. of IEEE ICOIN-15: The 15th International Conferenceon Information Networking*, pages 17–24, January 2001.

[6] Yuh-Shyan Chen and Kuan-Chang Lai. MESH: Multi-Eye Spiral-Hopping Protocol in a Wireless Ad Hoc Network. *IEICE Transactions on Communications*, E84-B(8), aug. 2001.

[7] Ying-Kwei Ho and Ru-Scheng 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.

[8] Yih-Chun Hu and David B.Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. In *Proc. of MOBICOM*, pages 231–242, 2000.

[9] David B. Johnson and David 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.

[10] Wen-Hwa Liao, Yu-Chee Tseng, Jang-Ping Sheu, and Shu-Ling 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.

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

[12] Chunhung Richard Lin and Jain-Shing Liu. QoS Routing in Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1426–1438, Auguest 1999.

[13] H. C. Lin and P. C. Fung. Finding Available Bandwidth in Multihop Mobile Wireless Networks. In *Proc. of the IEEE Vehicular Technology Conference*, pages 15–18, May, 2000.

[14] Anup Kumar Talukdar and B. R. Badrinath. Rate Adaptation Schemes in Networks with Mobile Hosts. In *Proc. of MOBICOM*, pages 169–180, 1998.

[15] Yu-Chee Tseng, Jang-Ping Sheu, and Chin-Yu Lin. A New Mutli-Channel MAC Protocol with On-Demand Channel Assignment for Mobile Ad Hoc Networks. *Int'l Symp. on Parallel Architectures, Algorithms and Networks (I-SPAN)*, pages 232–237, 2000.