A Hexagonal-Tree TDMA-Based QoS Multicasting Protocol for Wireless Mobile Ad-Hoc Networks

Yuh-Shyan Chen and Han-Gu Tsai
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, Taiwan, R.O.C.

Abstract

In this paper, we propose a new TDMA-based QoS multicast routing protocol, namely the hexagonal-tree QoS multicast protocol, for a wireless mobile ad hoc network. Existing QoS routing solutions have addressed this problem by assuming a stronger multi-antenna model or a less stronger CDMA-over-TDMA channel model. This study attempts to build a new multicast tree structure, namely hexagonal tree, to serve as the QoS multicasting tree, where the MAC sub-layer is adopted the TDMA channel model. In this work, both of the hidden-terminal and exposed-terminal problems are taken into consideration to possibly exploit the time-slot re-use capability. The hexagonal-based scheme offers a higher success rate to construct the QoS multicast tree due to using the hexagonal-tree. The hexagonal-tree is a tree whose sub-path of the tree is the hexagonal-path. The hexagonal-path is a special two-path structure. This greatly improves the success rate by means of multi-path routing. Performance analysis results is finally discussed to demonstrate the efficient QoS multicasting achievement.

Keywords

hexagonal-tree, QoS, mobile ad hoc network (MANET), multi-path, time division multiple access, wireless network.

I. Introduction

A mobile ad-hoc networks (MANET) [9][20] consists of wireless hosts that communicate with each other, in the absence of a fixed wireless network infrastructure. Due to the factors, such as radio power limitations, power consumption, and channel utilization, a mobile host may not be able to communicate directly with other hosts via a single-hop communication. In a MANET, host mobility can cause unpredictable topology changeability, therefore the design of MANET QoS routing protocol is more complicated because it needs strong fault-tolerant capability. Extensive research efforts have been devoted to the design of MANET routing protocols [5][6][20]. These works all address the following issues: discovering a route from a source node to a destination, maintaining a route while it is being used, and delivering data packets to the destination host. However, these protocols, when searching for a route to the destination, always concerns with shortest-path routing and the availability of multitude routes in the MANET's dynamically changing environment. Connections with quality-of-service (QoS) requirements, such as multimedia with delay and bandwidth constrains, are less frequently addressed, especially for designing the QoS multicast protocol for MANETs.

Recently, some interest works intensively study the QoS issue in the MANET [2][3][4][7][8][10][12][13][14][15][16]. Initially, a quite ideal model is assumed that the bandwidth of a link can be determined independently of its neighboring links [4]. This strong assumption may be realized by a costly multi-antenna model such that a host can send/receive using different antennas independently and simultaneously. Under such a model, a ticket-based QoS routing protocol is proposed in [4]. Recently, a less stronger CDMA-over-TDMA

Yuh-Shyan Chen is with Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, R.O.C., Email: yschen@cs.ccu.edu.tw. This work was supported by NSC91-2213-E-305-001 and NSC91-2213-E-305-003.

channel model is assumed in [12][13] to develop QoS routing protocols 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. Observe that a code assignment protocol should be supported (this can be regarded as an independent problem, which can be found in [12][13]). Based on such a model, Lin calculates the end-to-end path bandwidth to develop DSDV-based QoS routing protocol [13] and on-demand QoS routing protocol [12] for MANETs. More recently, one simpler TDMA model [11] is further considered in [11] to develop an uni-path QoS routing protocol in the MANET. Their approach shows how to allocate time slots on each link of a path such that no two adjacent links share a common time slot.

This paper mainly presents a new TDMA-based QoS multicast routing protocol, namely the hexagonal-tree QoS multicast protocol, for a wireless mobile ad hoc network. Existing QoS routing solutions have addressed this problem by assuming a stronger multi-antenna model or a less stronger CDMA-over-TDMA channel model. This study attempts to build a new multicast tree structure, namely hexagonal tree, to serve as the QoS multicasting tree, where the MAC sub-layer is adopted the TDMA channel model. In this work, both of the hidden-terminal and exposed-terminal problems are taken into consideration to possibly exploit the time-slot re-use capability. The hexagonal-based scheme offers a higher success rate to construct the QoS multicast tree due to using the hexagonal-tree. The hexagonal-tree is a tree whose sub-path of the tree is the hexagonal-path. The hexagonal-path is a special two-path structure. This greatly improves the success rate by means of multi-path routing. Performance analysis results is finally discussed to demonstrate the efficient QoS multicasting achievement.

The rest of the paper is organized as follows. Section II presents basic idea and challenges. Our protocol is developed in Section III and experimental results are discussed in Section IV. Section V concludes this paper.

II. Basic Idea and Challenges

Existing TDMA-based routing protocols ignores the transmission activities of individual mobile hosts. Let's recall the hidden-terminal and exposed-terminal problems, which are well-known problems in the literature of radio-based communication. Consider the scenario in Fig. 1, where the bandwidth requirement is 2 time slots. Fig. 1(a) shows the hidden-terminal problem that if A sends to B on slots $1, 2$, then D will not allowed to send data to F on slots $1, 2$. On the contrary, Fig. 1(b) illustrates the exposed-terminal problem that if A sends to B on slots $1, 2$, then C allows to send data to E on slots $1, 2$. Observe that if QoS routing protocol only considers the hidden-terminal problem, then no three adjacent links are allowed to share the same free time-slots. This limitation can be improved if the exposed-terminal problem is also taken into consideration, such that it is possibly that no two adjacent links shares same time slots. This observation motivates our design to take the hidden- and exposed-terminal problems into consideration. The major contribution of this study is to design a QoS multi-path multicast routing under above considerations.

This work aims to develop a new multicast tree by exploiting the time-slot re-use capability. Before formally defining the our multicast-tree structure, the network model is assumed. The MAC sub-layer in our model is implemented by using TDMA channel model, where each frame is divided into a control and data phases, each data phase of a frame is split into k time slots. Some terms is defined. Let $(h_1, h_2, \cdots, h_k)$

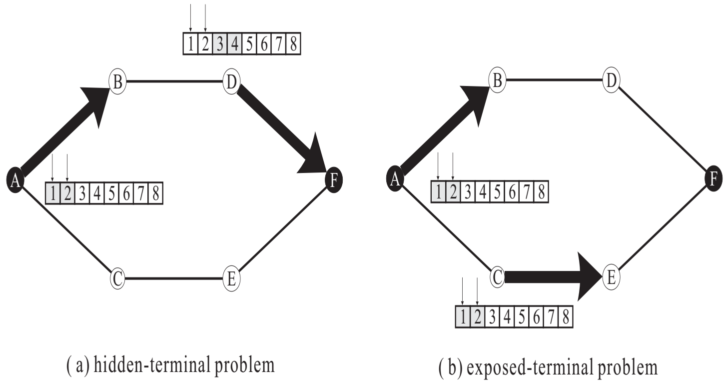

Fig. 1. Example of bandwidth calculation by the hidden-terminal and exposed-terminal problems

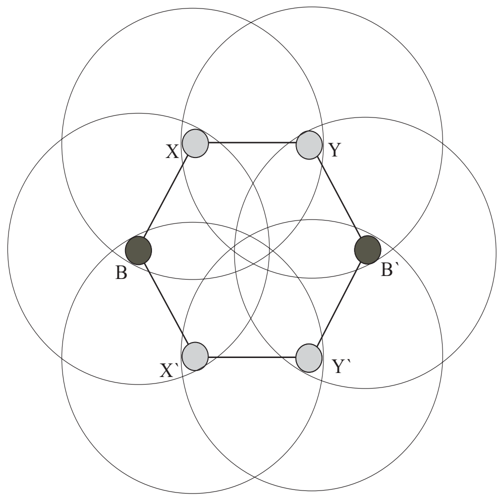

Fig. 2. Example of a hexgonal-block

represent as a path from node h_1 to node h_k. A node B is said as a *branch node* if there exist more than one disjoint paths from a three-hop neighboring node B', so that nodes B and B' are called as *a pair of branch nodes*. Some terms are defined as follows.

Definition 1 Hexagonal-Block: *Given a pair of two branch nodes B and B', there are two disjoint three-hop paths (B, X, Y, B') and (B, X', Y', B') between these two branch nodes, where X and X' are not 1-hop neighbors and Y and Y' are not 1-hop neighbor. Paths (B, X, Y, B') and (B, X', Y', B') forms a hexagonal-block, let $\begin{bmatrix} B \begin{matrix} X & Y \\ X' & Y' \end{matrix} B' \end{bmatrix}$ denote as a hexagonal-block between B and B'.*

For instance, a hexagonal-block $\begin{bmatrix} B \begin{matrix} X & Y \\ X' & Y' \end{matrix} B' \end{bmatrix}$ is shown in Fig. 2(a), there are two disjoint three-hop paths (B, X, Y, B') and (B, X', Y', B') between a pair of branch nodes B and B'. In general, nodes X and X' may be neighboring, and nodes Y and Y' may be neighboring. The hexagonal-block gives a less stronger condition that nodes X and X' are not 1-hop neighbors and nodes Y and Y' are not 1-hop neighbors, as illustrated in Fig. 2(a). This condition is important for the time-slot reservation scheme in order to exploit the time-slot re-use capability.

Using the hexagonal-block allows us to equally split the original data packet into two sub-packets, such that

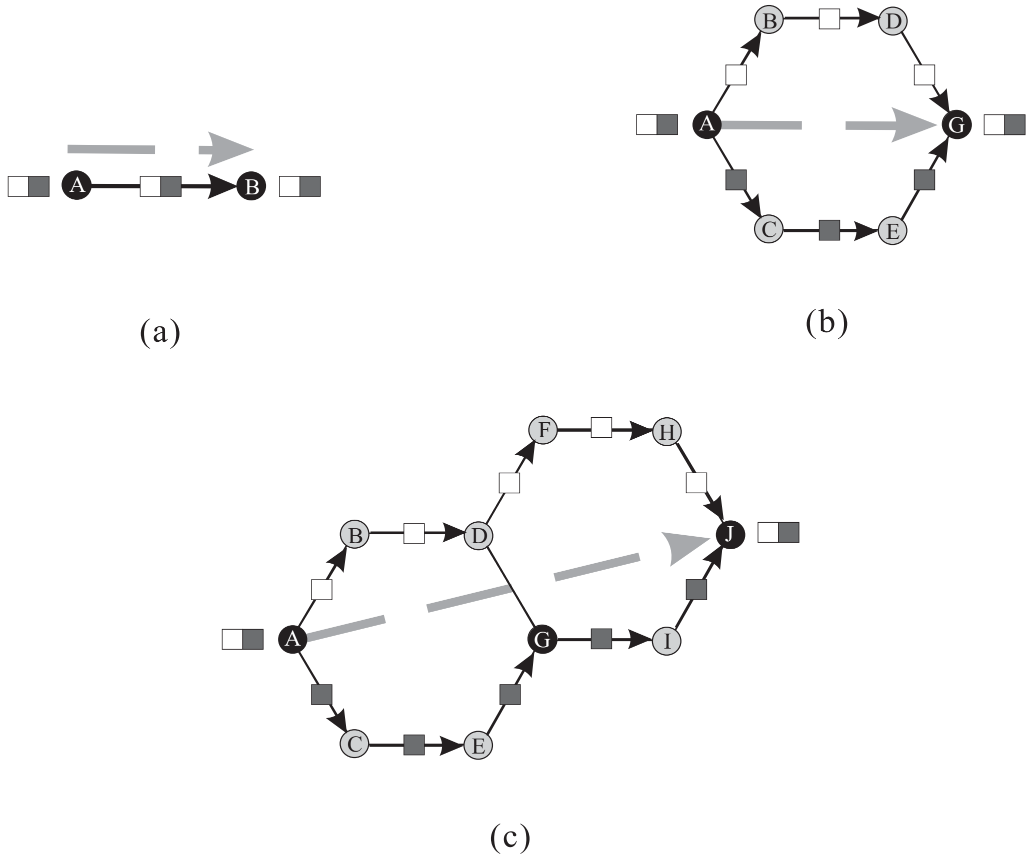

Fig. 3. Example of (a) an uni-path, (b) a hexagonal-block and (c) a hexagonal-twin

one sub-packet goes through (B, X, Y, B') and another sub-packet goes through (B, X', Y', B'). Continually, the hexagonal-twin is now formally defined as follows. This hexagonal-twin is a fundamental component to construct the *hexagonal-path* and *hexagonal-tree* structure.

Definition 2 Hexagonal-Twin: *Given two hexagonal-blocks* $Z=\begin{bmatrix} A & \begin{matrix} N_1 & \mathbf{N}_3 \\ N_2 & N_4 \end{matrix} & \mathbf{B} \end{bmatrix}$ *and* $Z'=\begin{bmatrix} \mathbf{N}_3 & \begin{matrix} P_1 & P_2 \\ \mathbf{B} & P_3 \end{matrix} & C \end{bmatrix}$, *let*

$$Z^{Z'} = \begin{bmatrix} & & & P_1 & P_2 & \\ & N_1 & \mathbf{N}_3 & & & C \\ A & & & \mathbf{B} & P_3 & \\ & N_2 & N_4 & & & \end{bmatrix}$$

denote as a hexagonal-twin if a link ($\mathbf{N}_3$, ***B***) *is shared by Z and Z'*.

For instance, a hexagonal-twin $Z^{Z'} = \begin{bmatrix} & & & F & H & \\ & B & D & & & J \\ A & & & G & I & \\ & C & E & & & \end{bmatrix}$ is given in Fig. 3(c), there are two disjoint paths (A, B, D, F, H, J) and (A, C, E, G, I, J) between nodes A and J. Similarly, using the hexagonal-twin, one sub-packet travels along (A, B, D, F, H, J) and another sub-packet travels along (A, C, E, G, I, J).

Definition 3 Hexagonal-Path: *A path is denoted as a hexagonal-path if each sub-path in the path may be an uni-path, a hexagonal-block, or a hexagonal-twin.*

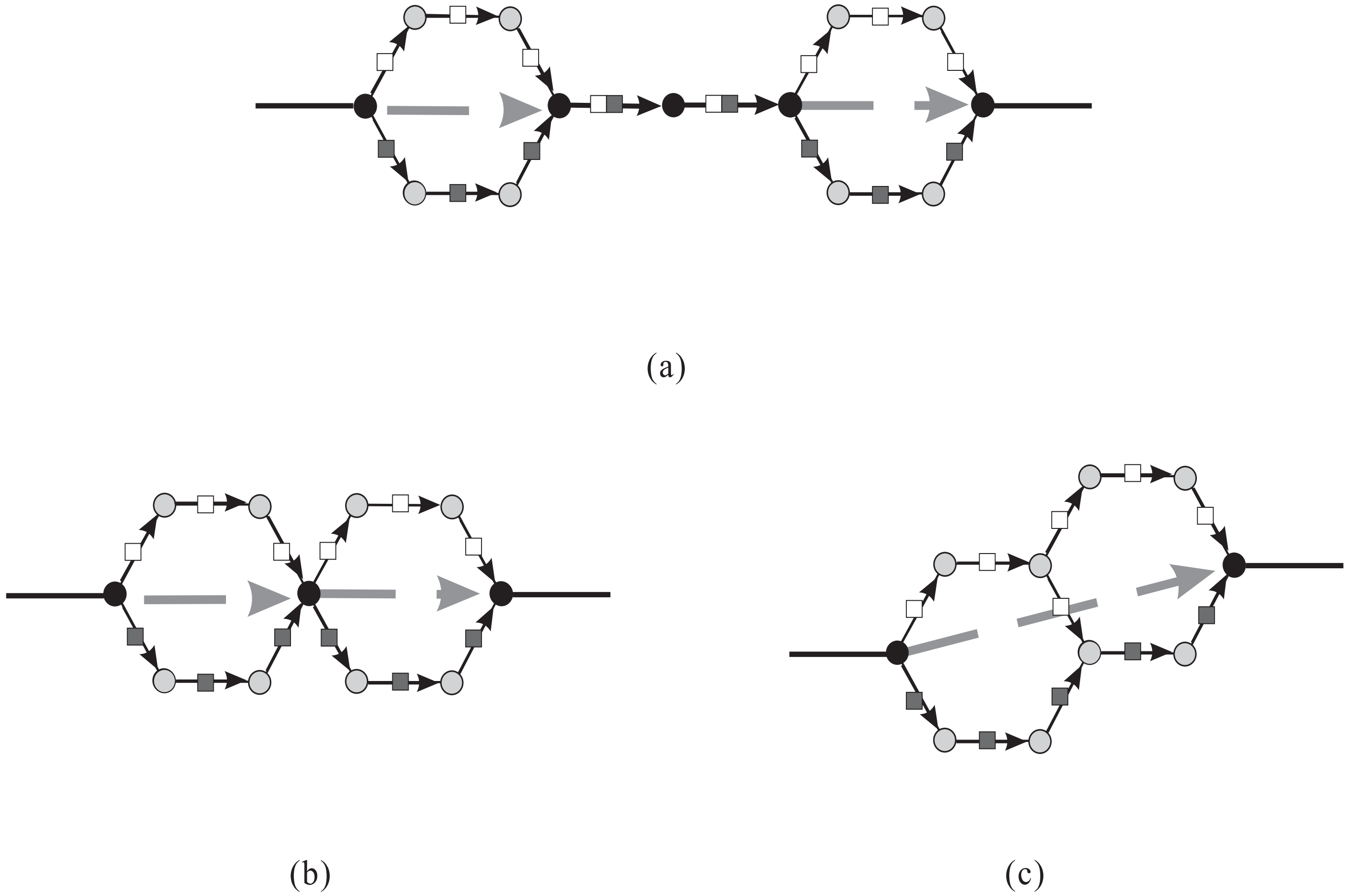

Fig. 4. Examples of hexagonal-paths

In the QoS route discovery, if an uni-path exists under satisfying a given QoS requirement, the uni-path will be identified. However, if the QoS uni-path is not existed, then the hexagonal-block and hexagonal-twin are used to offer a multi-path routing scheme, which aim to increase the success rate of identifying the QoS route. For instance as shown in Fig. 3(a), an uni-path with two time slots is constructed. If we cannot find an uni-path with two time slots, then a hexagonal-block with two time slots or a hexagonal-twin with two time slots are possibly identified, as shown in Fig. 3(b) and Fig. 3(c). Further, Fig. 4(a) shows a hexagonal-path with two non-adjacent hexagonal-blocks connecting by an uni-path. Fig. 4(b) illustrates a hexagonal-path constructed by two adjacent hexagonal-blocks. Fig. 4(c) displays a hexagonal-path constructed by a hexagonal-twin. Note that, all cases are instances of the hexagonal-path. We introduce the time-slot re-use capability of a hexagonal-block and hexagonal-twin as follows.

Lemma 1 *A time slot t can be used by a host X to send to another host Y without causing collision if the following conditions are all satisfied.*

1. Slot t is not yet scheduled to send or receive in neither X nor Y.

2. For any 1-hop neighbor Z of X, slot t is not scheduled to receive in Z. Slot t can be scheduled (or reused) to send in Z, but cannot send to any other 1-hop neighbors of X (conflict with condition 1).

3. For any 1-hop neighbor Z of Y, slot t is not scheduled to send in Z. Slot t can be scheduled (or reused) to

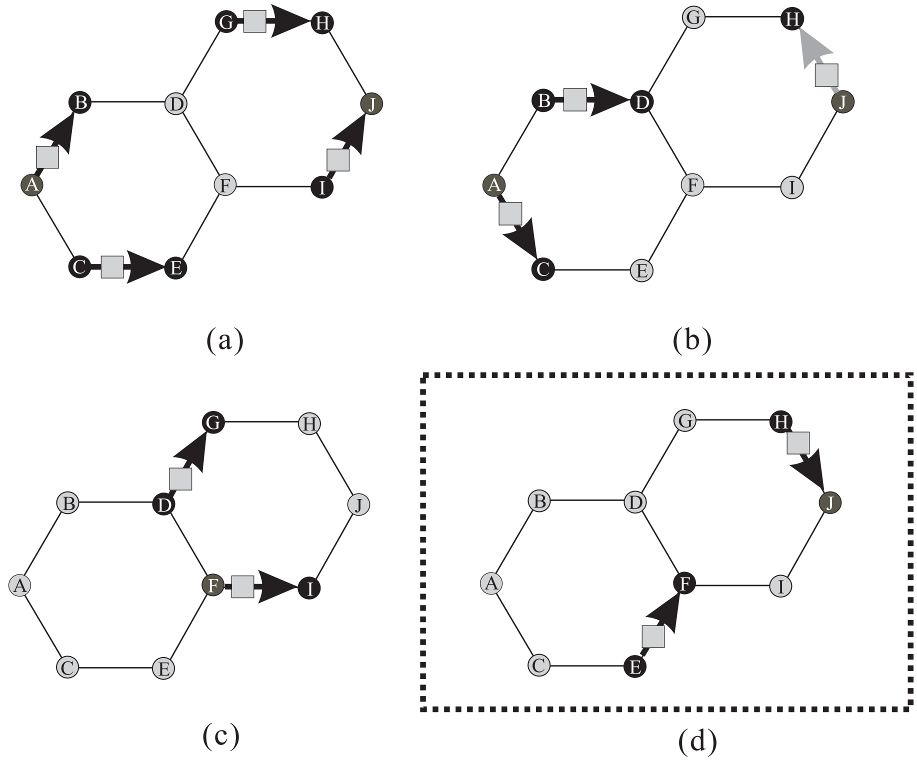

Fig. 5. Time-slot re-use capability of a hexagonal-twin

receive in Z, but cannot receive from any other 1-hop neighbors of Y (conflict with condition 1).

Given a hexagonal-twin $\begin{bmatrix} & & & G & H & \\ & B & D & & & J \\ A & & & F & I & \\ & C & E & & & \end{bmatrix}$ as illustrated Fig. 5, the time-slot re-use capability on all links of a hexagonal-twin are listed.

1. Time slot t scheduled in $\overrightarrow{AB}$ can be re-used in $\overrightarrow{CE}$, $\overrightarrow{GH}$, $\overrightarrow{IJ}$ (see Fig. 5(a)).
2. Time slot t scheduled in $\overrightarrow{BD}$ can be re-used in $\overrightarrow{AC}$, $\overrightarrow{JH}$ (see Fig. 5(b)).
3. Time slot t scheduled in $\overrightarrow{DG}$ can be re-used in $\overrightarrow{FI}$ (see Fig. 5(c)).
4. Time slot t scheduled in $\overrightarrow{EF}$ can be re-used in $\overrightarrow{HJ}$ (see Fig. 5(d)).

To prepare constructing a *hexagonal-tree* structure, a hexagonal-branch is defined based on the hexagonal-twin structure.

Definition 4 Hexagonal-Branch: *Given three hexagonal-blocks* $Z = \begin{bmatrix} A & \begin{matrix} N_1 & N_3 \\ N_2 & N_4 \end{matrix} & B \end{bmatrix}$, $Z' = \begin{bmatrix} N_3 & \begin{matrix} P_1 & P_2 \\ B & P_3 \end{matrix} & C \end{bmatrix}$ *and* $Z'' = \begin{bmatrix} N_4 & \begin{matrix} B & P_3 \\ P_4 & P_5 \end{matrix} & D \end{bmatrix}$, *let* $Z \begin{matrix} Z' \\ Z'' \end{matrix} =$

$$\begin{bmatrix} & & & P_1 & P_2 & \\ & N_1 & \mathbf{N}_3 & & & C \\ A & & & \mathbf{B} & \mathbf{P}_3 & \\ & N_2 & \mathbf{N}_4 & & & D \\ & & & P_4 & P_5 & \end{bmatrix}$$

denote as a hexagonal-branch which is constructed by Z, Z′, and Z″, where three links (N_3, B), (N_4, B) and (B, P_3) are shared by Z, Z′, and Z″.

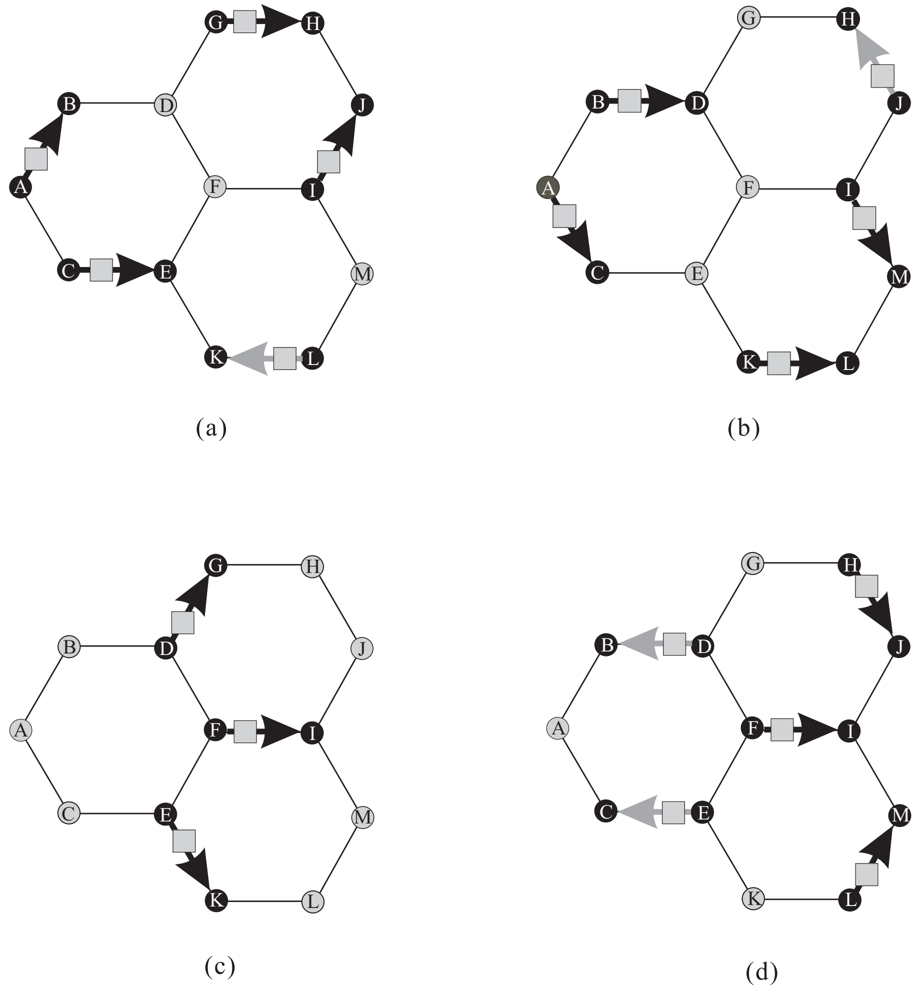

Fig. 6. Time-slot re-use capability of a hexagonal-branch

Given a hexagonal-branch $\begin{bmatrix} & & & G & H & \\ & B & D & & & J \\ A & & & F & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$ as illustrated Fig. 6, the time-slot re-use capability of all links of a hexagonal-branch are listed.

1. Time slot t scheduled in $\overrightarrow{AB}$ can be re-used in $\overrightarrow{CE}$, $\overrightarrow{GH}$, $\overrightarrow{IJ}$, $\overrightarrow{LK}$ (see Fig. 6(a), which is same as Fig. 5(a)).

2. Time slot t scheduled in $\overrightarrow{AC}$ can be re-used in $\overrightarrow{BD}$, $\overrightarrow{JH}$, $\overrightarrow{IM}$, $\overrightarrow{KL}$ (see Fig. 6(b), which is same as Fig. 5(b)).

3. Time slot t scheduled in $\overrightarrow{DG}$ can be re-used in $\overrightarrow{FI}$, $\overrightarrow{EK}$ (see Fig. 6(c), which is same as Fig. 5(c)).

4. Time slot t scheduled in $\overrightarrow{HJ}$ can be re-used in $\overrightarrow{FI}$, $\overrightarrow{LM}$, $\overrightarrow{DB}$, $\overrightarrow{EC}$ (see Fig. 6(d), which is different with Fig. 5(d)).

As shown in Fig. 7(d), a hexagonal-branch $Z^{Z'}_{Z''} = \begin{bmatrix} & & & G & H & \\ & B & D & & & J \\ A & & & F & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$ is given. Observe that $Z^{Z'}_{Z''}$ can be viewed as constructed by two hexagonal-twins $Z^{Z'} = \begin{bmatrix} & & & G & H & \\ & B & D & & & J \\ A & & & F & I & \\ & C & E & & & \end{bmatrix}$ and $Z_{Z''} = \begin{bmatrix} & B & D & & & \\ A & & & F & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$. Observe that, a original packet is split into two sub-packets. From host A, the two sub-packet can send to host J by paths (A, B, D, G, H, J) and (A, C, E, F, I, J). Similarly, the two sub-packet can send to host M by paths (A, B, D, F, I, M) and (A, C, E, K, L, M). This achieves the purpose of delivering the original message to two distinct hosts. This is the fundamental operation of the tree structure.

As shown in Fig. 7(d), a hexagonal-branch $Z^{Z'}_{Z''} = \begin{bmatrix} & & & F & H & \\ & B & D & & & J \\ A & & & G & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$ is given. The $Z^{Z'}_{Z''}$ can be viewed as two hexagonal-twins $Z^{Z'} = \begin{bmatrix} & & & F & H & \\ & B & D & & & J \\ A & & & G & I & \\ & C & E & & & \end{bmatrix}$ and $Z_{Z''} = \begin{bmatrix} & B & D & & & \\ A & & & G & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$. Observe that, a original packet is split into two sub-packets. From host A, the two sub-packet can send to host J by paths (A, B, D, F, H, J) and (A, C, E, G, I, J). Similarly, the two sub-packet can send to host M by paths (A, B, D, G, I, M) and (A, C, E, K, L, M). This achieves the purpose of delivering the original message to two distinct hosts. This is the fundamental operation of the tree structure.

The purpose of the tree branch is to send the same data packet into two distinct nodes under satisfying the given bandwidth requirement. Each node can continually build a sub-tree to cover all possible destination nodes. All possible cases of tree branch are illustrated in Fig. 7, and a tree branch is occurred between node A and nodes J, M. 7. Initially, Fig. 7(a) shows that node A sends packet to nodes J and M by using three uni-paths $\overrightarrow{AG}$, $\overrightarrow{GJ}$, and $\overrightarrow{GM}$ with two time-slots. If this case is failed, three other cases are then to construct for greatly improving the success rate of constructing a QoS multicast tree. First, Fig. 7(b) illustrates a hexagonal-block $\left[A \begin{smallmatrix} B & D \\ C & E \end{smallmatrix} G\right]$ and two uni-paths $\overrightarrow{GJ}$, and $\overrightarrow{GM}$ with two time-slots are used. Fig. 7(c) displays an uni-path $\overrightarrow{AG}$ and two hexagonal-blocks with two time-slots are used. Finally, Fig. 7(c) shows a hexagonal-branch with two time-slots is utilized. With the hexagonal-path and hexagonal-branch, the hexagonal-tree is formally defined as follows.

Definition 5 Hexagonal-Tree: *A tree is said as a hexagonal-tree if each path of the tree is the hexagonal-path.*

A possible hexagonal-tree, from source host S to destinations E and F, with uni-paths is illustrated in Fig.

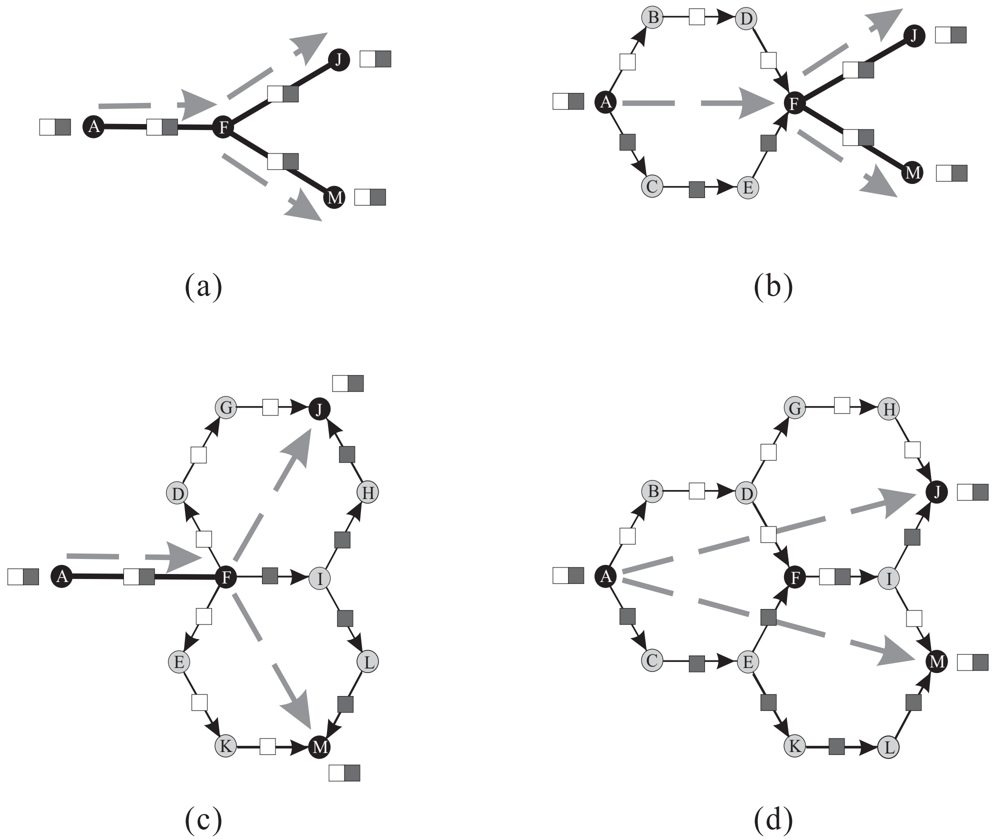

Fig. 7. Four cases of hexagonal-branchs

8, note that a hexagonal-branch appears between node B and nodes C, D. Observe that, the uni-path with satisfied QoS requirement will be identified with the higher priority during constructing the multicast tree. In the following, the construction of the hexagonal-tree will be presented.

III. The Hexagonal-Tree TDMA-Based QoS Multicast Routing Protocol

We first give an overview of our proposed protocol, a hexagonal-tree TDMA-Based QoS multicast protocol, in a MANET. The proposed protocol mainly constructs a hexagonal-tree to perform the on-demand QoS multicast routing operation. The designed protocol is achieved by the *hexagonal-branch/twin identification* and *hexagonal-tree construction* phases. The *hexagonal-branch/twin identification* phase identifies the hexagonal-branch/twin in a MANET. The *hexagonal-tree construction* phase constructs the hexagonal-tree by merging hexagonal-paths from a source to all destinations.

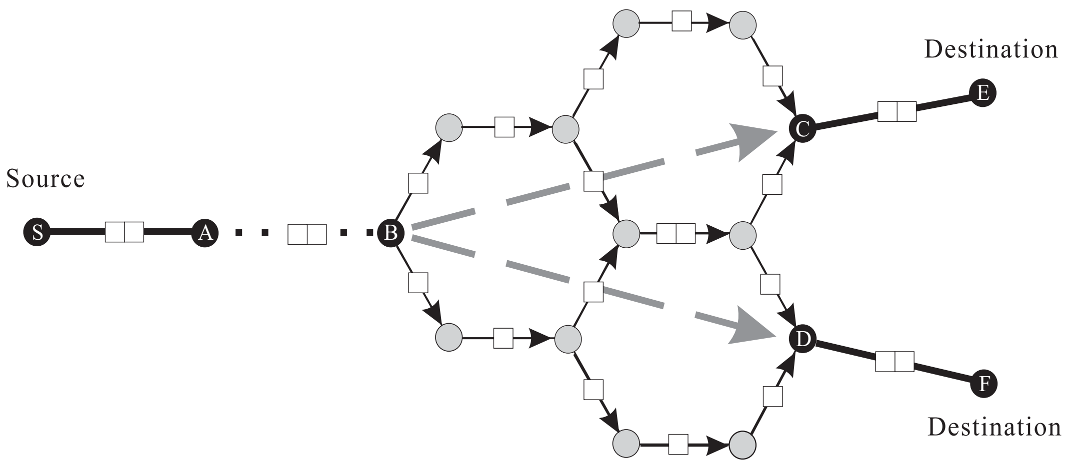

Fig. 8. Example of a hexagonal-tree

A. Phase 1: Hexagonal-Branch/Twin Identification

A.1 Hexagonal-Branch Identification

This phase searches for a hexagonal-branch under a given bandwidth requirement. Given a hexagonal-branch $\begin{bmatrix} & & & G & H & \\ & B & D & & & J \\ A & & & F & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$, which is constructed by three hexagonal-block $\left[A \begin{matrix} B & D \\ C & E \end{matrix} F\right]$, $\left[D \begin{matrix} G & H \\ F & I \end{matrix} J\right]$, and $\left[E \begin{matrix} F & I \\ K & L \end{matrix} M\right]$. Node F contains the following data structure if we attempt to successfully identify a hexagonal-branch. Observe that each node keeps local information of all one/two/three-hop neighboring nodes.

- $node_x$: Let $node_x \in \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$ and all one- or two-hop neighbors of $\{A, B, C, D, E, F, G, H, I, J, K, L, M\}$.
- A list of sending activities of $node_x$: Node $node_x$ records on which time slots $node_x$ will have sending activities. If $[y, l_1, \cdots, l_k]$ denotes $node_x$ send to node y on time slots $\{l_1, \cdots, l_k\}$. That is, a list of $[y, l_1, \cdots, l_k]$ is maintained.
- A list of receiving activities of $node_x$: Node $node_x$ records on which time slots $node_x$ will have receiving activities. If $[y, l_1, \cdots, l_k]$ denotes $node_x$ receive from node y on time slots $\{l_1, \cdots, l_k\}$, a list of $[y, l_1, \cdots, l_k]$ is maintained.

Observe that this section only discuss how to successfully identify a hexagonal-branch to exploit the time-slot re-use capability. Observe that, all records are collected into node F, which indicates that the time slot reservation of paths between A to J and A to M are determined by node F. This is the main overhead of our proposed protocol. From the simulation results in Section IV, the improved success rate of identifying QoS on-demand multicast tree can successfully cover this extra-overhead. Let Free_time_slot ($\overrightarrow{\alpha\beta}$) denote the time slots can be used to send from α to β, Send_time_slot ($\overrightarrow{\alpha\beta}$) denote the time slots have been used to send from α to β, and Receive_time_slot ($\overrightarrow{\alpha\beta}$) denote the time slots have been used to receive from α

to β. Observe that, Send_time_slot$(\overrightarrow{\alpha\beta})$ = Receive_time_slot$(\overrightarrow{\beta\alpha})$. For instance as illustrated in Fig. 9(a), Send_time_slot$(\overrightarrow{LK}) = \{1, 2\}$ and Receive_time_slot$(\overrightarrow{KL}) = \{1, 2\}$.

Given a hexagonal-branch $\begin{bmatrix} & & & G & H & \\ & B & D & & & J \\ A & & & F & I & \\ & C & E & & & M \\ & & & K & L & \end{bmatrix}$ as shown in Fig. 9(a), $\overrightarrow{AB}$ and $\overrightarrow{AC}$ cannot share same time slots, since our scheme is a multi-path (two-path). Based on Fig. 6(a), we do know that Free_time_slot $(\overrightarrow{AB})$ can be same with Free_time_slot $(\overrightarrow{CE})$, Free_time_slot $(\overrightarrow{GH})$, Free_time_slot $(\overrightarrow{IJ})$, Free_time_slot $(\overrightarrow{LK})$, but time slots scheduled in $\overrightarrow{AB}$ cannot be used in $\overrightarrow{FD}$. Similarly, from Fig. 6(b), Free_time_slot $(\overrightarrow{AC})$ can be same with Free_time_slot $(\overrightarrow{BD})$, Free_time_slot $(\overrightarrow{IM})$, Free_time_slot $(\overrightarrow{KL})$, Free_time_slot $(\overrightarrow{JH})$, but time slots scheduled in $\overrightarrow{AC}$ cannot be used in $\overrightarrow{FE}$, etc.

Let the bandwidth requirement be γ, we do the following reservation operations; (1) we try to allocate the $\frac{\gamma}{2}$ time slots to $\overrightarrow{AB}$, $\overrightarrow{CE}$, $\overrightarrow{GH}$, and $\overrightarrow{IJ}$ (see Fig. 6(a)), (2) we then try to allocate the $\frac{\gamma}{2}$ time slots to $\overrightarrow{AC}$, $\overrightarrow{BD}$, $\overrightarrow{IM}$, and $\overrightarrow{KL}$ (see Fig. 6(b)), (3) we then attempt to allocate $\frac{\gamma}{2}$ time slots to $\overrightarrow{HJ}$, $\overrightarrow{FI}$, and $\overrightarrow{LM}$ (see Fig. 6(c)), (4) we then attempt to allocate $\frac{\gamma}{2}$ time slots to $\overrightarrow{DG}$, $\overrightarrow{FI}$, and $\overrightarrow{EK}$ (see Fig. 6(d)), (5) we finally try to allocate $\frac{\gamma}{2}$ time slots to $\overrightarrow{DF}$ and $\overrightarrow{EF}$. If any one is failed, then the time slot reservation for a hexagonal-branch is failed. Assume that the bandwidth requirement is γ, the formal reservation procedure is given herein.

A1. Repeatedly reserve $\frac{\gamma}{2} - \alpha$ same time slots and α distinct time slots on $\overrightarrow{AB}$, $\overrightarrow{CE}$, $\overrightarrow{GH}$ and $\overrightarrow{IJ}$, for $\alpha = 0$ to $\frac{\gamma}{2}$, until the reservation is successful, and update the possible nodes' records of sending and receiving activities. Otherwise, if the reservation is failed, then the reservation for a hexagonal-branch is failed, and exit the procedure (see Fig. 6(a)).

A2. Repeatedly reserve $\frac{\gamma}{2} - \alpha$ same time slots and α distinct time slots on $\overrightarrow{AC}$, $\overrightarrow{BD}$, $\overrightarrow{IM}$, and $\overrightarrow{KL}$, for $\alpha = 0$ to $\frac{\gamma}{2}$, until the reservation is successful, and update the possible nodes' records of sending and receiving activities. Otherwise, if the reservation is failed, then the reservation for a hexagonal-branch is failed, and exit the procedure (see Fig. 6(b)).

A3. Repeatedly reserve $\frac{\gamma}{2} - \alpha$ same time slots and α distinct time slots on $\overrightarrow{DG}$, $\overrightarrow{FI}$, and $\overrightarrow{EK}$, for $\alpha = 0$ to $\frac{\gamma}{2}$, until the reservation is successful, and update the possible nodes' records of sending and receiving activities. Otherwise, if the reservation is failed, then the reservation for a hexagonal-branch is failed, and exit the procedure (see Fig. 6(c)).

A4. Repeatedly reserve $\frac{\gamma}{2} - \alpha$ same time slots and α distinct time slots on $\overrightarrow{HJ}$, $\overrightarrow{FI}$, and $\overrightarrow{LM}$, for $\alpha = 0$ to $\frac{\gamma}{2}$, until the reservation is successful, and update the possible nodes' records of sending and receiving activities. Otherwise, if the reservation is failed, then the reservation for a hexagonal-branch is failed, and exit the procedure (see Fig. 6(d)).

A**5.** Repeatedly reserve $\frac{\gamma}{2} - \alpha$ same time slots and α distinct time slots on $\overrightarrow{DF}$ and $\overrightarrow{EF}$, for $\alpha = 0$ to $\frac{\gamma}{2}$, until the reservation is successful, and update the possible nodes' records of sending and receiving activities. Otherwise, if the reservation is failed, then the reservation for a hexagonal-branch is failed, and exit the procedure.

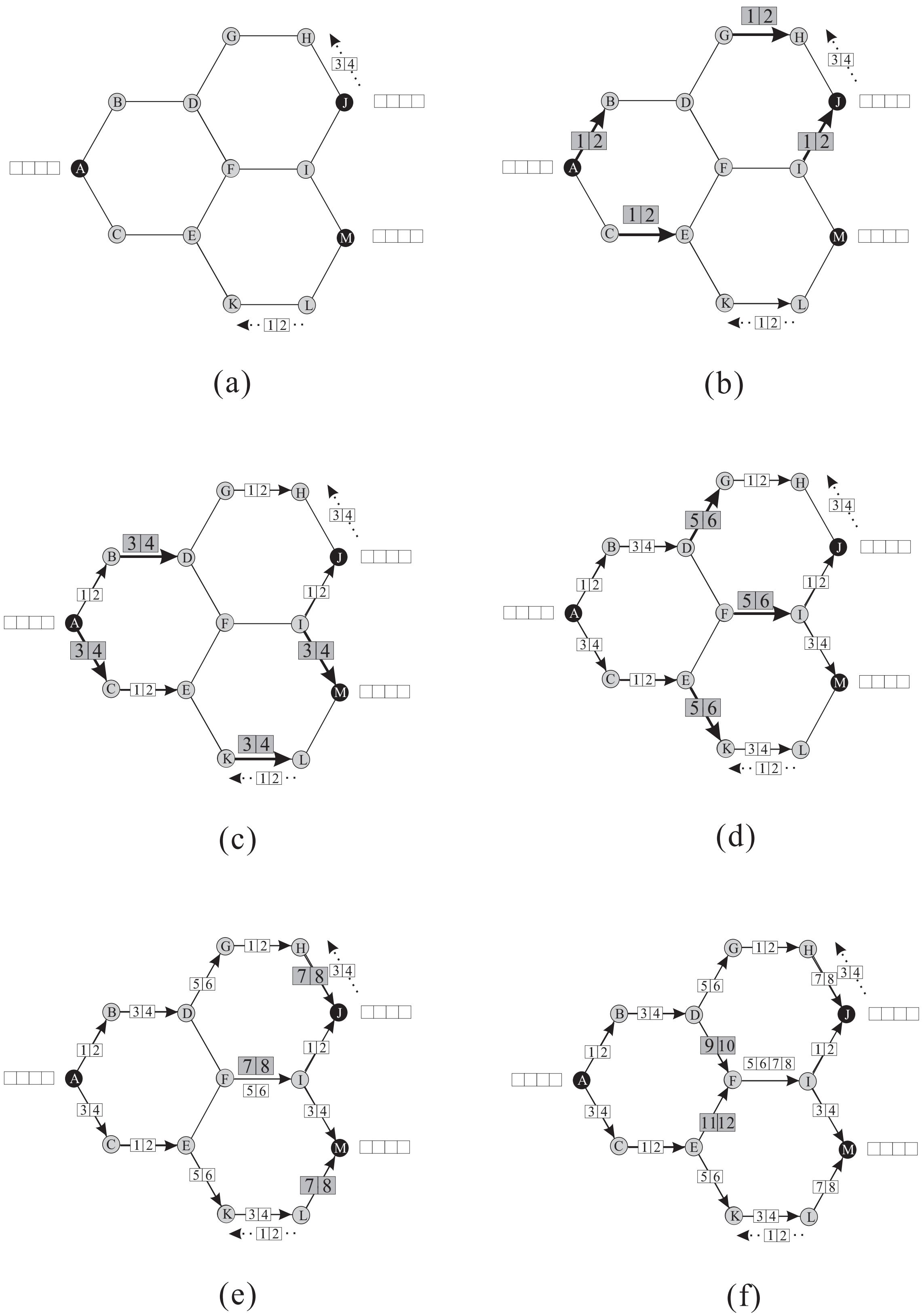

Fig. 9. Time-slot reservation for a hexagonal-branch

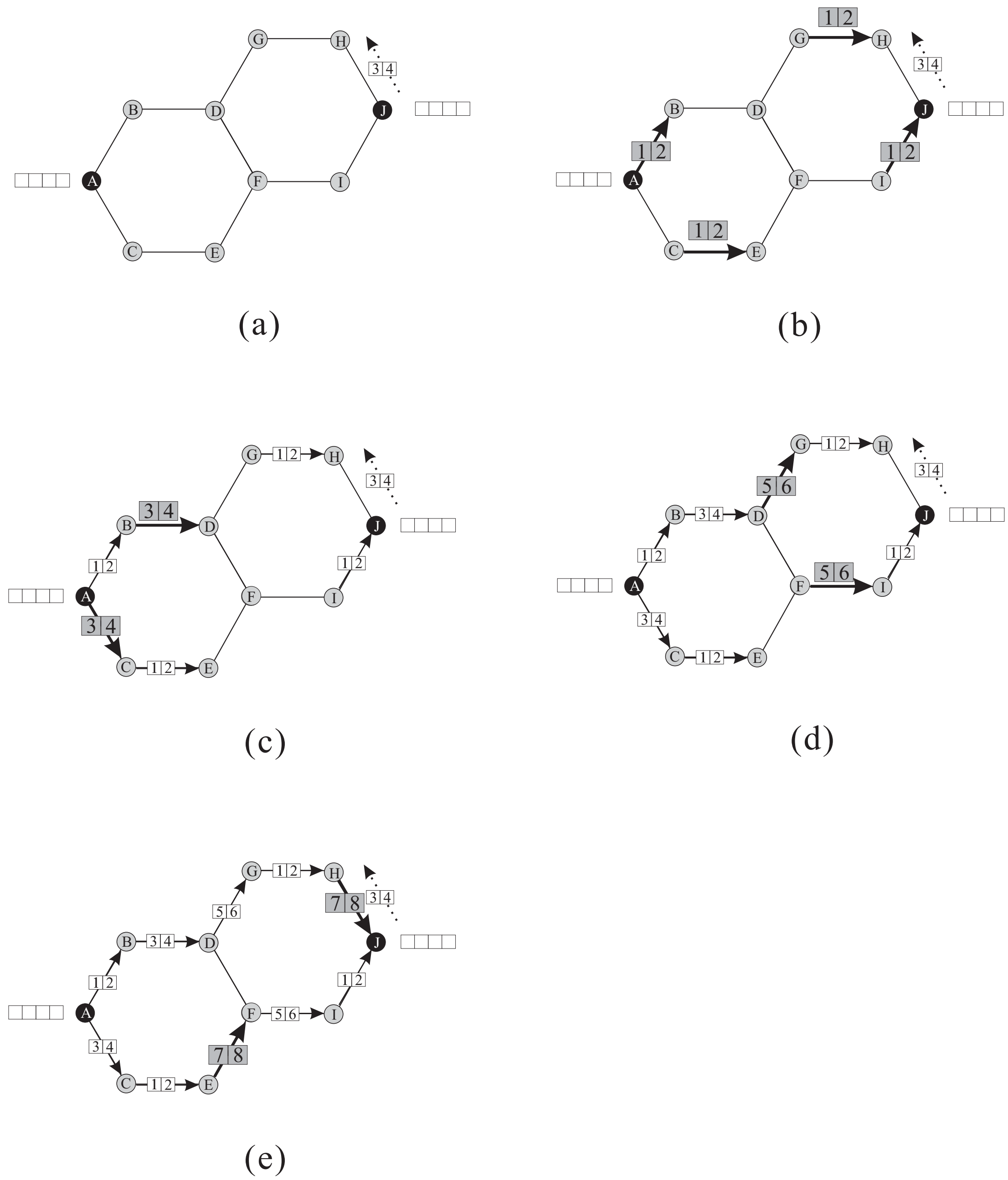

Fig. 10. The time slot reservation for a hexagonal-twin

For instance as shown in Fig. 9(a), let Send_time_slot($\overrightarrow{JH}$)=$\{3,4\}$ and Send_time_slot ($\overrightarrow{LK}$)=$\{1,2\}$ before constructing the hexagonal-branch. Let bandwidth requirement γ be 4 slots. Fig. 9(b) shows that slots $\{1,2\}$ are successfully allocate to $\overrightarrow{AB}$, $\overrightarrow{CE}$, $\overrightarrow{GH}$ and $\overrightarrow{IJ}$ for A1 step. Fig. 9(c) displays that slots $\{3,4\}$ are successfully allocate to $\overrightarrow{AC}$, $\overrightarrow{BD}$, $\overrightarrow{IM}$, and $\overrightarrow{KL}$ for A2 step. Fig. 9(d) displays that slots $\{5,6\}$ are successfully allocate to $\overrightarrow{DG}$, $\overrightarrow{FI}$, and $\overrightarrow{EK}$ for A3 step. Fig. 9(e) illustrates that slots $\{7,8\}$ are successfully allocate to $\overrightarrow{HJ}$, $\overrightarrow{FI}$, and $\overrightarrow{LM}$ for A4 step. Fig. 9(f) illustrates that slots $\{9,10\}$ and $\{11,12\}$ are respectively allocate to $\overrightarrow{DF}$ and $\overrightarrow{EF}$ for A5 step. Finally, a hexagonal-branch with four time slots is successfully constructed.

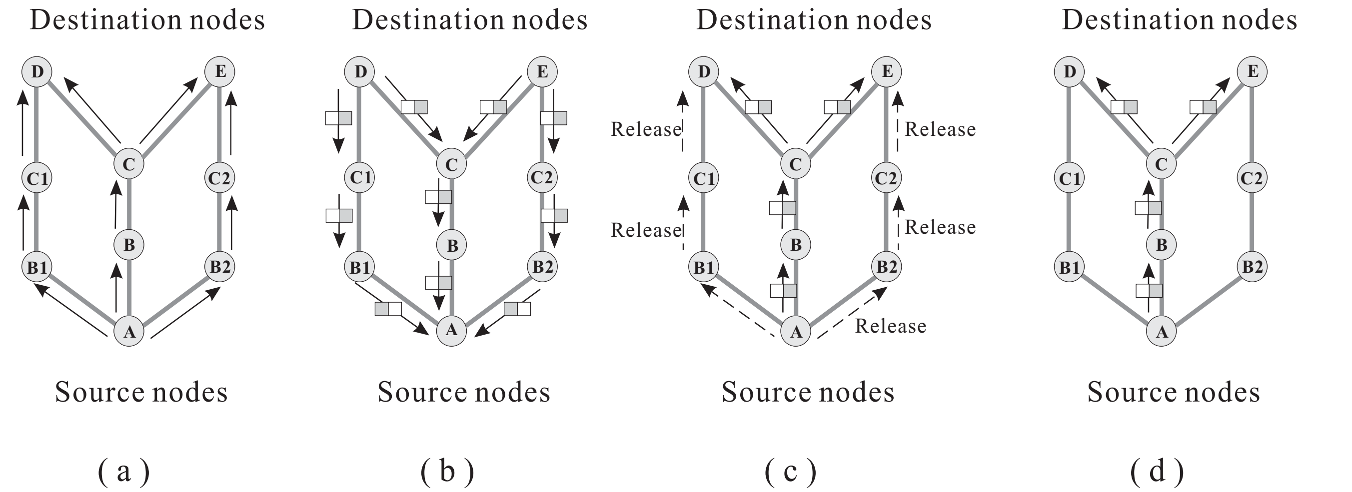

Fig. 11. Four merging steps of the hexagonal-tree

A.2 Hexagonal-Twin Construction

This phase searches for a hexagonal-twin if given bandwidth requirement is γ. A hexagonal-twin $Z^{Z'}$ or $Z_{Z''}$ is constructed from a hexagonal-branch $Z^{Z'}_{Z''}$ without the hexagonal-block Z' or Z''. The operation is similar with the identification of the hexagonal-branch. We omit the details herein. For instance as shown in Fig. 10(a), let Send_time_slot($\overrightarrow{JH}$)=$\{3, 4\}$ before constructing the hexagonal-twin. Let bandwidth requirement γ be 4 slots. Fig. 10(b) shows that slots $\{1, 2\}$ are successfully allocate to $\overrightarrow{AB}$, $\overrightarrow{CE}$, $\overrightarrow{GH}$ and $\overrightarrow{IJ}$ (same as A1 step, and see Fig. 5(a)). Fig. 10(c) displays that slots $\{3, 4\}$ are successfully allocate to $\overrightarrow{AC}$ and $\overrightarrow{BD}$ (same as A2 step, and see Fig. 5(b)). Fig. 10(d) displays that slots $\{5, 6\}$ are successfully allocate to $\overrightarrow{DG}$ and $\overrightarrow{FI}$ (same as A3 step, and see Fig. 5(c)). Fig. 10(e) illustrates that slots $\{7, 8\}$ are successfully allocate to $\overrightarrow{EF}$ and $\overrightarrow{HJ}$ (see Fig. 5(d)). Finally, a hexagonal-twin with four time slots is successfully constructed.

B. Phase 2: Hexagonal-Tree structure

The hexagonal-tree construction phase is divided into three operations.

1. *The hexagonal-path discovery*: Based on identified hexagonal-branches and hexagonal-twins, many hexagonal-paths from a source to a given set of destinations are constructed (see Fig. 11(a)).

2. *The hexagonal-path reply*: Destination receives the route-request packet from a source, and replies a route-reply packet to the source (see Fig. 11(b)).

3. *The hexagonal-tree construction*: Multiple hexagonal-paths are received from all destinations, the hexagonal-tree is finally established in the source (see Fig. 11(c)). Finally, the source sends data packet according to the determined the hexagonal-tree as shown in Fig. 11(d).

B.1 The Hexagonal-Path Discovery

After identifying possible hexagonal-branches and hexagonal-twins, we now describe how to construct the hexagonal-path. Let $[\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_i](b)$ denote as a hexagonal-path, where α_i can be an uni-path, or a

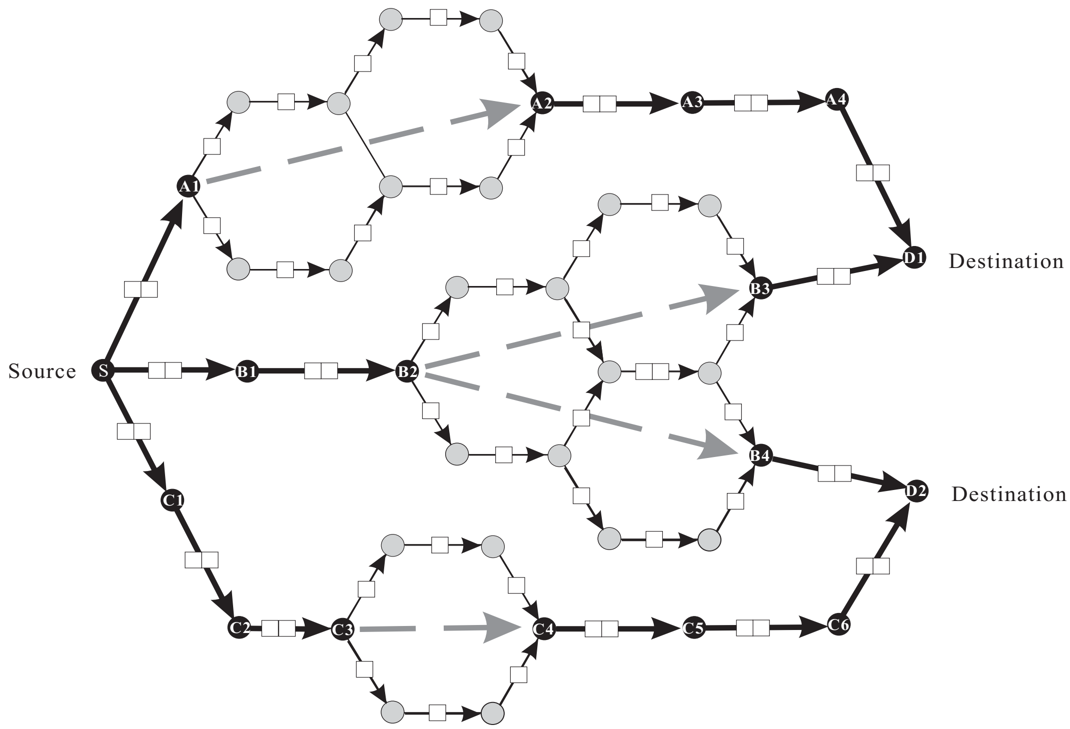

Fig. 12. The hexagonal-tree discovery

hexagonal-block, or a hexagonal-twin, and b represents as the hexagonal-path bandwidth. The hexagonal-path discovery operation is given.

$(C1)$ Source node initiates and floods a hexagonal-path REQUEST-packet into MANET. If there is an uni-path α_1 satisfying the required bandwidth, then a hexagonal-path $[\alpha_1]$ is constructed, where α_1 indicates an uni-path satisfying the bandwidth requirement.

$(C2)$ If no uni-path exists, we will further check if one or more hexagonal-branches exist. If a hexagonal-branch exists, then a hexagonal-path $[\alpha_1]$ is constructed, where α_1 is a hexagonal-branch.

$(C3)$ If both of steps $C1$ and step $C2$ are failed, we further check if a hexagonal-twin or a hexagonal-block exists or not. If a hexagonal-twin or a hexagonal-block exists, then a hexagonal-path $[\alpha_1]$ is constructed, where α_1 is a hexagonal-twin or a hexagonal-block.

$(C4)$ Repeatedly perform $C1$, $C2$ and $C3$ steps until a hexagonal-path arriving to one of destination nodes, then a hexagonal-path $[\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_k](b)$ is established.

For instance, many hexagonal-paths, from source S to destination nodes $D1$ and $D2$, are illustrated in Fig. 12.

B.2 The Hexagonal-Path Reply

Each destination node replies all possible hexagonal-path $[\alpha_i, \alpha_{i-1}, \alpha_{i-2}, \cdots, \alpha_1]$ to the source node. For instance, some hexagonal-paths are replied, from destination nodes $D1$ and $D2$ to source S, as illustrated in Fig. 13.

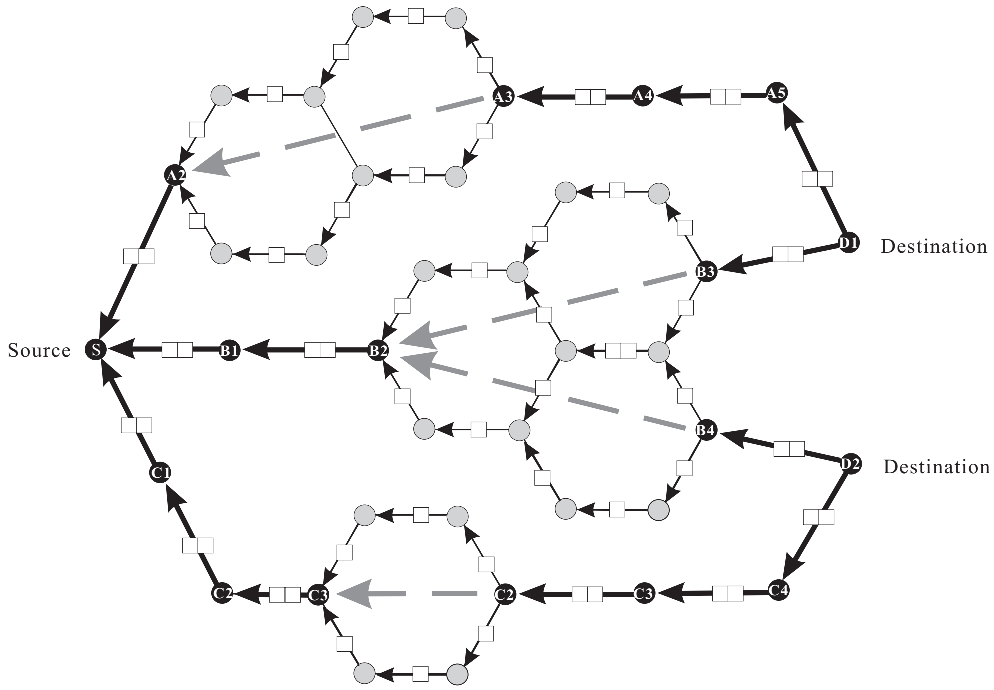

Fig. 13. The hexagonal-tree reply

B.3 The Hexagonal-Tree Construction

This study mainly constructs the hexagonal tree which is modified from the spiral-fat-tree on-demand multicast(SOM) protocol [5]. Observe that, the construction of the multicast tree can use the results of [3], [19], [18], [20], [17]. In this work, all spiral-path of the spiral-fat-tree [5] are replaced with the hexagonal-path for the purpose of providing the QoS-extension multicast protocol.

Given that two hexagonal-paths $[\alpha_1,\ \alpha_2,\ \alpha_p,\ \alpha_{p+1,} \cdots,\ \alpha_k](b)$, and $[\alpha_1,\ \alpha_2,\ \alpha_p,\ \alpha'_{p+1,} \cdots,\ \alpha'_k](b)$ have the same hexagonal sub-path $[\alpha_1,\ \alpha_2, \cdots,\ \alpha_p](b)$, we denote $\cap([\alpha_1, \alpha_2, \alpha_p, \alpha_{p+1,} \cdots, \alpha_k](b),\ [\alpha_1,\ \alpha_2,\ \alpha_p,\ \alpha'_{p+1,} \cdots,\ \alpha'_k](b))$ = $[\alpha_1,\ \alpha_2, \cdots,\ \alpha_p](b)$. Further, let $[\alpha_1,\ \alpha_2, \cdots,\ \alpha_p](b, s)$ denote as s hexagonal-paths with same $[\alpha_1,\ \alpha_2, \cdots,\ \alpha_p](b)$. We also denote $\mid [\alpha_1,\ \alpha_2, \cdots,\ \alpha_p](b) \mid$ = p to be the shared hexagonal-path length. Given each pair of s and s' replied hexagonal-paths, among many replied hexagonal-paths from destination nodes to the source node, we have the merging cirterion to construct the hexagonal-tree according to values of p, s, and b.

$(D1)$ If the s replied hexagonal-paths with the maximum values of p, s and b, then these s hexagonal-paths are merged together with the highest priority.

$(D2)$ If there are s and s' replied hexagonal-paths with the same values of p and s, then s hexagonal-paths with the greater value of b are merged together.

$(D3)$ If there are s and s' replied hexagonal-paths with the same values of b and s, then s hexagonal-paths with the greater value of p are merged together.

$(D4)$ If there are s and s' replied hexagonal-paths with the same values of b and p, then s hexagonal-paths with the greater value of s are merged together.

$(D5)$ If there are s and s' replied hexagonal-paths with the same values of p, then s hexagonal-paths with

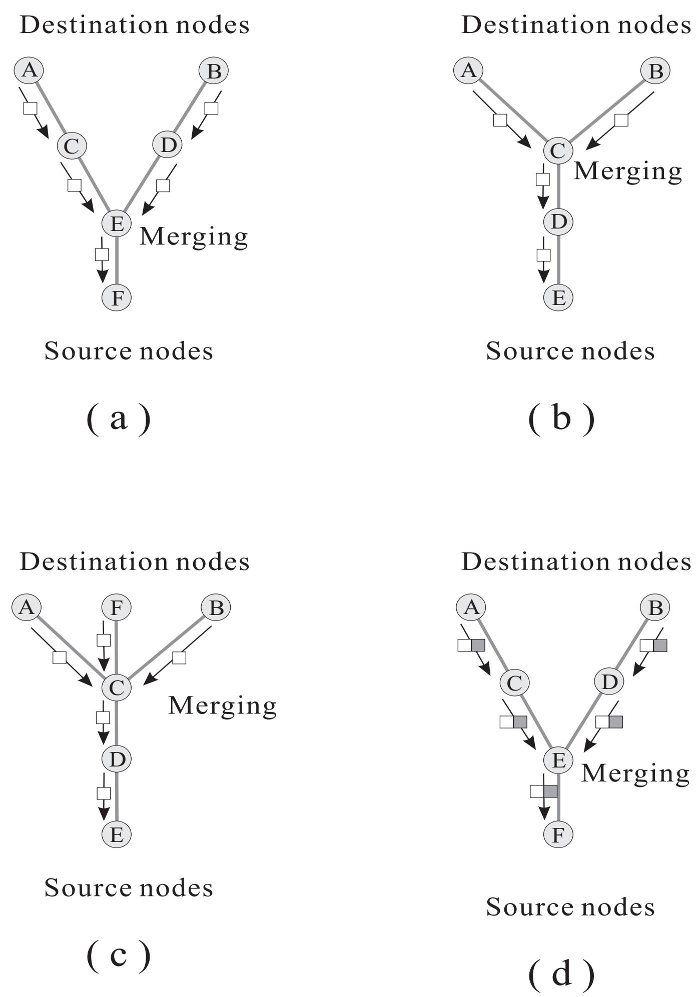

Fig. 14. The merging examples

the greater value of b are merged together.

$(D6)$ If there are s and s' replied hexagonal-paths with the same values of s, then s hexagonal-paths with the greater value of b are merged together.

$(D7)$ If there are s and s' replied hexagonal-paths with the same values of b, then s hexagonal-paths with the greater value of s are merged together.

$(D8)$ If there are s and s' replied hexagonal-paths with the different values of p, b and s, then s hexagonal-paths with the greater value of b are merged together.

For example, Fig. 14(a) and Fig. 14(b) are instance of the D3 case, Fig. 14(b) and Fig. 14(c) are instance of D7 case, Fig. 14(c) and Fig. 14(d) are instance of D2 case, and Fig. 14(b) and Fig. 14(d) are instance of D6 case.

IV. Experimental Results

A simulator is developed by C++ to evaluate the performance of our approach. The simulation parameters in the simulator are given below.

- The *number of mobile hosts* is ranging from 20 to 50.

- The *number of time slots* of the data frame is assumed to be 16 slots.
- The *bandwidth requirement* are 2, 4, 6 and 8 time slots.
- The *mobility* is ranging from 10 to 40 Km/hr.
- The *message length* is ranging from 1 to 4 Mbit/s.
- The *radio transmission range* is 200 m.
- The *transmission rate* is 5Mbits/sec.

To examine the effectiveness of our approach, we compare our proposal with two well-known MANET multicast routing protocols, which are AODV [3] and ODMRP [17]. Observe that, AODV and ODMRP do not offer the QoS capability. In addition, Tseng [11] designed a TDMA-based QoS uni-path routing protocol. To make a fair comparison, we provide the QoS-extension AODV and ODMRP protocols by integrating with Tseng's TDMA-based QoS uni-path routing result [11]. In this simulation, these two integrated results are denoted as AODV(+Tseng) and ODMRP(+Tseng), respectively.

The simulator is simulated in a 1000$\times$1000 m^2 area. 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. 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). 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.
- $Overhead$ (OH): The number of packets of transmitted packets, including the control and data packets.
- $Latency(LT)$: The interval from the time the multicast was initiated to the time the last host finishing its multicasting.

It is worth mentioning that an efficient QoS routing protocol is achieved by with high success rate SR, and low overhead OH and low LT. In the following, we illustrate the performance of SR, OH and LT from several prospects.

A. Performance of $Success\ Rate$ (SR)

The observed results of the performance of success rate vs. number of mobile hosts and network bandwidth are illustrated in Fig. 15. The low success rate of our approach will be obtained if the bandwidth requirement is large. Four kinds of effects are illustrated.

1a) Effects of number of hosts: Each value in Fig. 15(a) is obtained by assuming the bandwidth requirement (slot number) is 2, the number of destinations is 2 and the mobility is 10 Km/hr. From Fig. 15(a), the greater number of hosts is, the higher SR value will be obtained. For instance, if number of hosts is 20, then SR values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 12%, 14%, and 17%, respectively. If number of hosts is 50, then SR values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 62%, 69%, and 81%, respectively. This indicates that our protocol acquires the better success rate than other two schemes under various number of hosts.

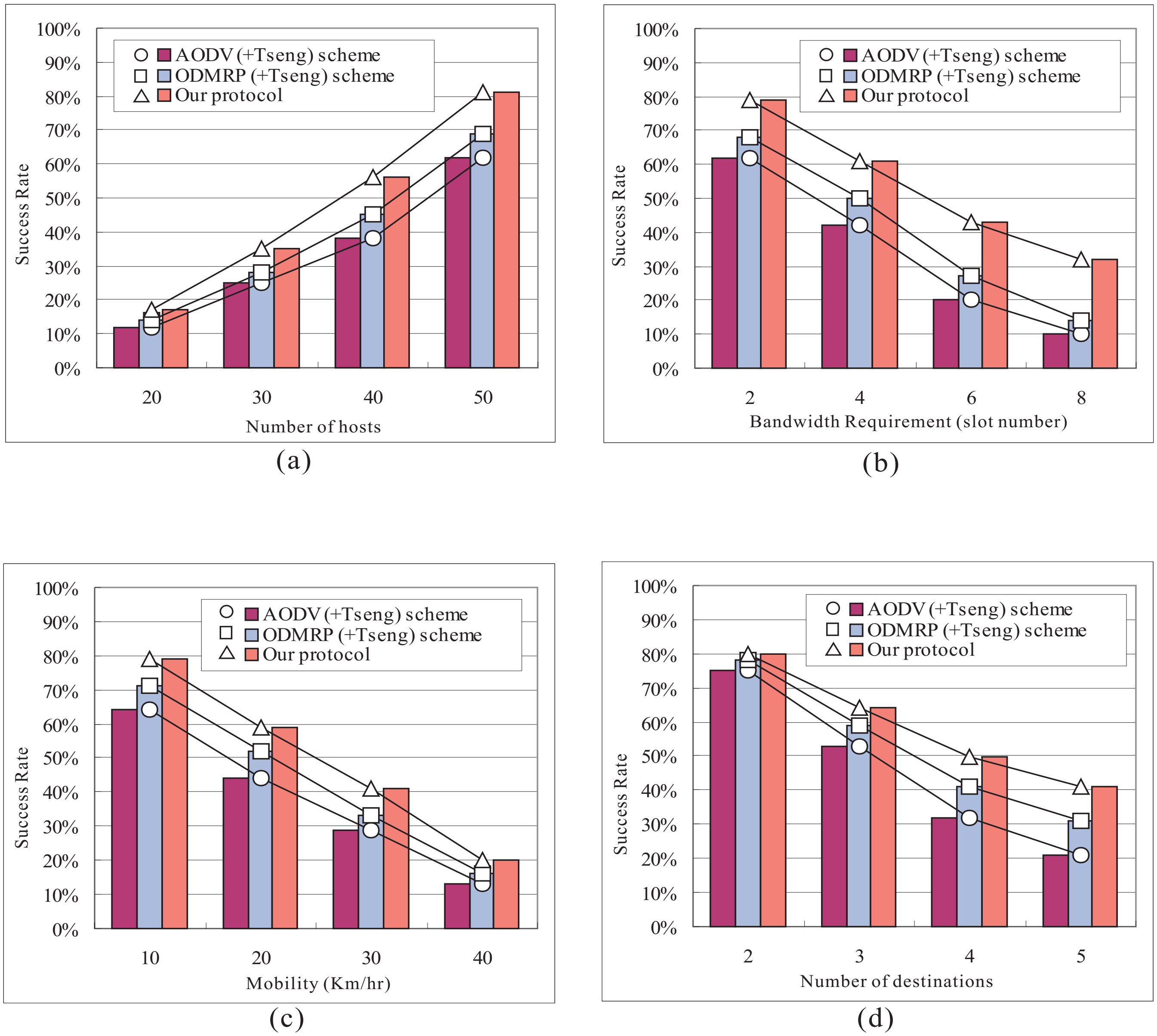

Fig. 15. Performance of success rate vs. effect of (a) number of hosts, (b) bandwidth requirement, (c) mobility, and (d) number of destinations

1b) Effects of bandwidth requirement: Each value in Fig. 15(b) is obtained by assuming the number of hosts is 50, the number of destinations is 2 and the mobility is 10 Km/hr. Fig. 15(b) illustrates that the higher bandwidth requirement is, the lower SR value will be obtained. For instance, if bandwidth requirement is two, then SR values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 62%, 68%, and 79%, respectively. If bandwidth requirement is eight, then SR values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 10%, 14%, and 32%, respectively. This indicates that our protocol acquires the better success rate than other two schemes even in various bandwidth requirement.

1c) Effects of mobility: Each value in Fig. 15(c) is obtained by assuming the number of hosts is 50, the number of destinations is 2 and bandwidth requirement is 2. Fig. 15(c) indicates that the higher mobility is, the lower SR value will be. For instance, if the mobility is 10 Km/hr, then SR values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 64%, 71%, and 79%, respectively. If the mobility is 40 Km/hr, then SR values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 13%, 16%, and 20%, respectively. This indicates that our protocol acquires the better success rate than other two schemes under various mobility.

1d) Effects of number of destinations: Each value in Fig. 15(d) is obtained by assuming the number of hosts is 50, the mobility is 10 Km/hr and bandwidth requirement is 2. Fig. 15(d) indicates that the greater number of destinations is, the lower SR value will be. For instance, if the number of destination is two, then SR values of AODV(+Tseng), ODMRP(+Tseng), and our protocol,are 75%, 78%, and 80%, respectively. If the number of destination is five, then SR values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 21%, 32%, and 41%, respectively. This indicates that our protocol acquires the better success rate than other two schemes under various number of destinations.

B. *Performance of* $OverHead$ (OH)

The observed results of the performance of overhead vs. number of mobile hosts and network bandwidth are illustrated in Fig. 16. The low overhead of our approach will be obtained if the bandwidth requirement is large. Four kinds of effects are illustrated.

2a) Effects of number of hosts: Each value in Fig. 16(a) is obtained by assuming the bandwidth requirement is 2, the number of destinations is 2, message length is 1M and the mobility is 10 Km/hr. From Fig. 16(a), the greater number of hosts is, the higher OH value will be obtained. For instance, if number of hosts is 20, then OH values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 243, 312, and 411, respectively. If number of hosts is 50, then OH values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 1534, 1836, and 2410, respectively. This indicates that our protocol acquires the higher overhead than other two schemes.

2b) Effects of bandwidth requirement: Each value in Fig. 16(b) is obtained by assuming the number of hosts is 20, the number of destinations is 2, message length is 1M and the mobility is 10 Km/hr. Fig. 16(b) shows that the higher bandwidth requirement is, the stable OH value will be obtained. For instance, if bandwidth requirement is 2, then OH values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 286, 336, and 409, respectively. If bandwidth requirement is eight, then OH values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 289, 337, and 410, respectively. This indicates that our protocol acquires the higher overhead than other two schemes under various bandwidth requirement.

2c) Effects of mobility: Each value in Fig. 16(c) is obtained by assuming the number of hosts is 20, the number of destinations is 2, message length is 1M and bandwidth requirement is 2. Fig. 16(c) indicates that the higher mobility is, the higher OH value will be. For instance, if the mobility is 10 Km/hr, then OH values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 243, 276, and 386, respectively. If the mobility is 40 Km/hr, then OH values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 856, 965, and 1121, respectively. This indicates that our protocol acquires the higher overhead than other two schemes under various mobility.

2d) Effects of number of destinations: Each value in Fig. 16(d) is obtained by assuming the number of hosts is 20, the mobility is 10 Km/hr, message length is 1M and bandwidth requirement is 2. Fig. 16(d) indicates

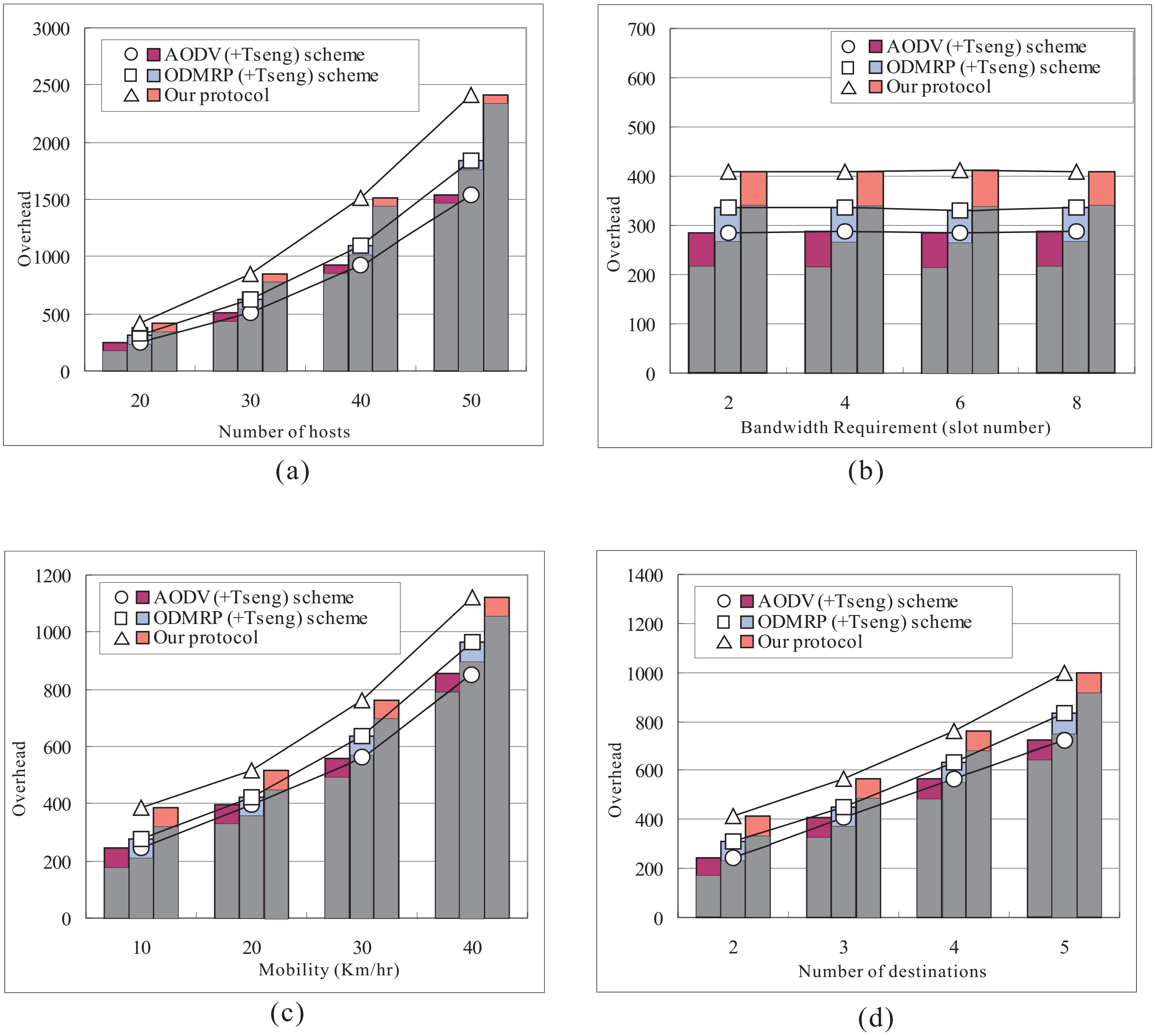

Fig. 16. Performance of overhead vs. effect of (a) number of hosts, (b) bandwidth requirement, (c) mobility, and (d) number of destinations

that the greater number of destinations is, the higher OH value will be. For instance, if the number of destination is two, then OH values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 235, 314 and 413, respectively. If the number of destination is five, then OH values of AODV(+Tseng), ODMRP(+Tseng) and our protocol are 724, 863 and 998. This indicates that our protocol acquires the higher overhead than other two schemes under various number of destinations.

C. Performance of $Latency$ (LT)

The observed results of the performance of latency vs. number of mobile hosts and network bandwidth are illustrated in Fig. 17. The high overhead of our approach will be obtained if the bandwidth requirement is large. Four kinds of effects are illustrated.

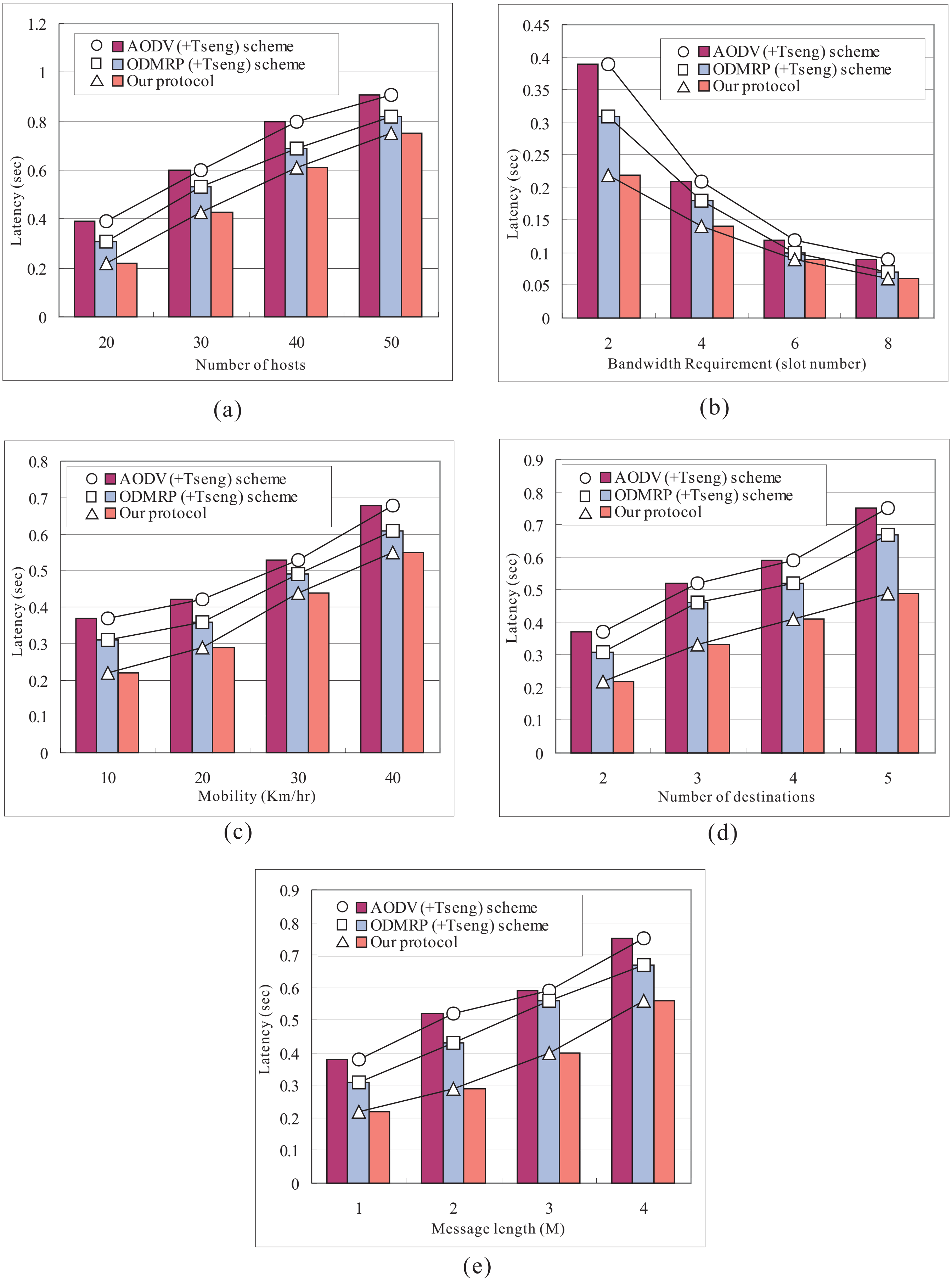

Fig. 17. Performance of latency vs. effect of (a) number of hosts, (b) bandwidth requirement, (c) mobility, (d) number of destinations, and (e) message length

3a) Effects of number of hosts: Each value in Fig. 17(a) is obtained by assuming the bandwidth requirement is 2, the number of destinations is 2, message length is 1M and the mobility is 10 Km/hr. From Fig. 17(a), the greater number of hosts is, the higher LT value will be obtained. For instance, if number of hosts is 20, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.39, 0.31, and 0.22, respectively. If number of hosts is 50, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.91, 0.82, and 0.75, respectively. This indicates that our protocol acquires the lower latency than other two schemes under various number of hosts.

3b) Effects of bandwidth requirement: Each value in Fig. 17(b) is obtained by assuming the number of hosts is 20, the number of destinations is 2, message length is 1M and the mobility is 10 Km/hr. Fig. 17(b) illustrates that the higher bandwidth requirement is, the lower LT value will be obtained. For instance, if bandwidth requirement is two, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.39, 0.31, and 0.22, respectively. If bandwidth requirement is eight, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.09, 0.07, and 0.06, respectively. This indicates that our protocol acquires the lower latency than other two schemes even in various bandwidth requirement.

3c) Effects of mobility: Each value in Fig. 17(c) is obtained by assuming the number of hosts is 20, the number of destinations is 2, message length is 1M and bandwidth requirement is 2. Fig. 17(c) indicates that the higher mobility is, the higher LT value will be. For instance, if the mobility is 10 Km/hr, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.37, 0.31, and 0.22, respectively. If the mobility is 40 Km/hr, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.68, 0.61, and 0.55, respectively. This indicates that our protocol acquires the lower latency than other two schemes under the various mobility.

3d) Effects of number of destinations: Each value in Fig. 17(d) is obtained by assuming the number of hosts is 20, mobility is 10 Km/hr, message length is 1M and bandwidth requirement is 2. Fig. 17(d) indicates that the greater number of destinations is, the low LT value will be. For instance, if the number of destination is two, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.37, 0.3, and 0.23, respectively. If the number of destination is five, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.75, 0.67, and 0.49, respectively. This indicates that our protocol acquires the lower latency than other two schemes under the various number of destinations.

3e) Effects of message length: Each value in Fig. 17(e) is obtained by assuming the number of hosts is 20, mobility is 10 Km/hr, the number of destinations is 2 and bandwidth requirement is 2. Fig. 17(e) indicates that the greater number of destinations is, the lower LT value will be. For instance, if the number of destination is two, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.37, 0.22, and 0.15, respectively. If the number of destination is five, then LT values of AODV(+Tseng), ODMRP(+Tseng), and our protocol are 0.75, 0.67, and 0.56, respectively. This indicates that our protocol acquires the lower latency than other two schemes under various message lengths.

Apparently, it is desirable to have a high SR and low LT. The higher the SR is, the lower the LT will be. It is always beneficial to adopt our protocol as demonstrated by the simulation results.

V. Conclusions

This paper presents a new TDMA-based QoS multicast routing protocol for a wireless mobile ad hoc network. This study builds a new multicast tree structure, namely hexagonal tree, to serve as the QoS multicasting tree, where the MAC sub-layer is adopted the TDMA channel model. Both of the hidden-terminal and exposed-terminal problems are taken into consideration to possibly exploit the time-slot re-use capability during constructing the hexagonal-tree. In this paper, the proposed hexagonal-based scheme indeed offers a higher success rate to construct the QoS multicast tree. This greatly improves the success rate by means of two-path routing. Performance analysis results finally demonstrate our efficient QoS multicasting achievement. In addition, the future work will attempt to develop a multi-source-extension QoS multicast protocol in MANETs.

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, August 1999.

[3] E.-M. Royer and C.-E. Perkins, ”Ad-hoc On-Demand Distance Vector Routing”, *Second IEEE Workshop on Mobile Computer System and Applicantion*, pp. 1488-1505, Aug. 1999.

[4] S. Chen, K. Nahrstedt and Y. Shavitt, ”A QoS-Aware Multicast Routing Protocol”. *In Proc. of IEEE INFOCOM*, pages 1594-1603, 2000.

[5] Y.-S. Chen and T.-S. Chen and C.-J. Huang, ”SOM: Spiral-Fat-Tree-Based On-Demand Multicast Protocol in a Wireless Ad-Hoc Network”, *Proc. of IEEE ICOIN-15: The 15th International Conferenceon Information Networking*, pages 17-24, January 2001.

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

[7] 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.

[8] Y.-C. Hu and D. Johnson, ”Caching Strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks”. In *Proc. of MOBICOM*, pages 231-242, 2000.

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

[10] W.-H. Liao and Y.-C. Tseng and J.-P. Sheu and S.-L. Wang, ”A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network”, *Proc. of IEEE ICN’01: International Conference on Networking*, July, 2001.

[11] W.-H. Liao and Y.-C. Tseng and K.-P. Shih, ”A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad Hoc Network”, *IEEE International Conference on Communications.*, pages 3186-3190, 2002.

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

[13] 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,Auguest 1999.

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

[15] A.-K. Talukdar and B. R. Badrinath. ”Rate Adaptation Schemes in Networks with Mobile Hoses”. In *Proc. of MOBICOM*, page 169-180,1998.

[16] Y.-C. Tseng, J.-P. Sheu, and C.-Y. Lin, ”A New Multi-Channel MAC Protocol with On-Demand Chennel Assignment for Mobile Ad Hoc Networks”. *Int’l Symp. on Parallel Architectures, Algorithms and Networks(I-SPAN)*, page 232-237, 2000.

[17] S.-J. Lee and W. Su and M. Gerla, ”On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks”, *IETF manet(draft-ietf-manet-odmrp-02.txt)*, 2000.

[18] S. Murthy and J. J. Garcia-Luna-Aceves, ”An efficient Routing Protocol for Wireless Networks”, *ACM Monile Networks*, pp. 183-197, Oct. 1996.

[19] S. E. Deering and D.R. Cheriton, ” Multicast Routing in Datagram Internetworks and Extend LANs”, *ACM Transactions on Computer Systems*, Vol. 8, No. 2, pp. 85-110, May 1990.

[20] C.-K. Toh, ”Associativity-based Routing For Ad-hoc Mobile Networks”, *Wireless Pers. Commun*. Rev, Vol. 2, No. 4, pp. 1-36, Mar. 1996.