

# Subarea Tree Routing (STR) in Multi-hop Wireless Ad hoc Networks

Guikai LIU, Chunli SHAN, Gang WEI, Hongjiang WANG

School of Electronic & Information Engineering, South China University of Technology, Guangzhou, China

E-mail: [gkliu@scut.edu.cn](mailto:gkliu@scut.edu.cn), [spg53@126.com](mailto:spg53@126.com)

Received February 28, 2009; revised March 15, 2009; accepted March 17, 2009

## Abstract

Subarea Tree Routing (STR), a new routing protocol for multi-hop wireless ad hoc networks, is proposed. The novelty of the STR protocol is to divide the whole network into many subareas constructed as a result of establishing subarea trees. Its main idea is to identify root nodes by manual configuration or auto-discovery process firstly, then the root nodes originate the process of establishing subarea trees, and finally each node either joins in a subarea tree or become an interconnect node. STR belongs to hierarchical routing protocol and does not attempt to consistently maintain routing information in every node. Furthermore, through the use of tree's intrinsic routing function, the STR protocol exhibits hybrid behavior of proactive and on-demand routing protocols. We prove the correctness of STR, and our simulation results show that the proposed scheme achieves lower route discovery delays, lower route discovery load and better performance of normalized routing load in large, mobile, ad hoc networks as compared with AODV.

**Keywords:** Wireless Ad Hoc Networks, Hierarchical Routing Protocol, Proactive Routing Protocol, On-Demand Routing Protocol, Subarea Tree Routing

## 1. Introduction

Multi-hop wireless ad hoc network, also called multi-hop wireless self-organizing network, does not rely on a fixed infrastructure and the network structure changes dynamically due to member mobility. Wireless ad hoc networks are very attractive for tactical communication in military and also expected to play an important role in many fields without the presence or use of a fixed infrastructure such as disaster search-and-rescue operations, data acquisition in remote areas, conference and convention centers etc. Each node in this network not only as a host but also as a router discovers and maintains routes to other nodes that may not be within direct wireless transmission range. To provide communications throughout the network, a sequence of neighbor nodes from a source to a destination form a multi-hop path and intermediate hosts relay packets in a store-and-forward mode.

The major challenges for multi-hop routing in wireless ad hoc networks are continuously changing network topology, low transmission power, and low available band-

width. In order to support multi-hop routing, much work has been done in this area and many protocols have been proposed. There are different standards to categorize these routing protocols: proactive routing versus on-demand routing, or flat routing versus hierarchical routing and so on.

In proactive protocols [1-6], routes between every two nodes are established in advance even if no data transmission is on demand. This is implemented by a node periodically updating its routing information and every node eventually has consistent and up-to-date global routing information for the entire network. This approach has the advantages of timely exchanging network information such as available bandwidth, delay, topology etc. and supporting real-time services. But it is not suitable for large-scale networks since many unused routes still need to be maintained and the periodic updating may incur overwhelming processing and communication overhead. The on-demand protocol (e.g. [7-10]) is more efficient because each node tries to reduce routing overhead by only sending routing packets when needed for data transmission and a route is released when the data transmission is finished. However, when link breakage is detected due to failure or node mobility, which often occurs in multi-hop wireless ad hoc networks, the delay and overhead of route reconstruction may be significant.

---

This work was supported in part by National High Technology Research and Development Program of China (863 Program) under contract 2007AA01Z210, and in part by the nature scientific fund of Guangdong province, China (No. 07006488).

As to flat routing and hierarchical routing, this is a classification according to the network structure underlying routing protocols. In flat routing protocol, there is a peer-to-peer relationship between every two nodes and each node participating in routing plays an equal role. Routing algorithm is easily realized. However, it is inferior scalability and limits network's scale in a certain extent. For example, AODV [7], DSR [8] and DSDV [1] are typical flat routing protocols. The hierarchical routing protocol divides the whole network into many logical areas and different routing strategies are used inside and outside the logical area. Compared with flat routing protocol, the hierarchical routing protocol possesses better scalability and is propitious to support large-scale networks. At present, the growing interest in wireless ad hoc network techniques has resulted in many hierarchical routing protocols such as ZRP [11], LANMAR [4], CGSR [12] and HSR [13] etc. In a word, each routing protocol has its own advantages and drawbacks as well. Research is continued on commendably satisfying the demands of multi-hop wireless ad hoc networks on the aspects of reliability, overhead control, efficiency and complexity.

In this paper, a novel hierarchical routing protocol that offers better scalability, low delay, low overhead and low normalized routing load in multi-hop wireless ad hoc networks is presented. The main feature of *Subarea Tree Routing (STR)* is to construct subarea trees according to the actual network environment and eventually the entire network will be divided into many logical subareas naturally. In addition, *STR* combines the advantages of proactive routing protocol and on-demand routing protocol, which provide the low route acquisition delay of proactive techniques and the low overhead of reactive methods. The rest of this paper is organized as follows: Next, a detailed description of *STR*, illustrating the key aspects of the protocol's operation, is given. Section 3 proves the correctness of *STR* and we analyze its performance in Section 4. Finally, Section 5 presents our conclusions.

## 2. Subarea Tree Routing

Our approach to routing in the multi-hop wireless ad hoc networks is based on the notion of *subarea tree*, which is originated by a selected root node. After a subarea tree is established, a logical subarea has already been formed. Namely a logical subarea consists of a subarea tree. Therefore, the whole network is composed of many subarea trees in the end. The *STR* protocol is hierarchical and includes the following components:

1. Identifying the root nodes and interconnecting them
2. Establishing subarea trees
3. Dynamic maintenance of subarea trees
4. Routing mechanism

### 2.1. Identifying the Root Nodes and Interconnecting Them

In an original network all nodes locate on equal status. Each node has a unique network identifier (ID or address) and its type is configured as "initial node". The root node is also a central node of a future subarea and we can identify the root nodes by static method or auto-discovery process. Static method is to adopt manual configuration to identify the root nodes and set the necessary parameters for them.

Auto-discovery process is more complex and involves the following steps: firstly given a regulation, namely the condition with which the root nodes must satisfy (such as ID Number, number of neighbor nodes, transaction capability, residual energy, stability or other measure values). Then select a node as source node arbitrarily from network (in general convenient for management or operation), and the source node broadcasts root auto-discovering request (*RADRO*) message which includes source node ID and the given regulation. Those nodes which satisfy the regulation will become root nodes and broadcast their own information such as node ID and node type. Every root node will records other root nodes' information what it has received.

It is interesting that there are two extreme states during the process of identifying root nodes. The first is that there is only one root node in the whole network. Namely the network is composed of one tree. The other is that every node is a root node. Of course, in general the second situation will not happen. However, as a matter of fact, what we want to identify is to find or approach the optimal balance point between these two states according to the real network environment.

Interconnecting root nodes means to build routing information between every two root nodes. The route discovery and route maintenance between root nodes can be realized by adopting existing flat routing protocol such as AODV [7], DSR [8] etc. The initial nodes which route passes through will become "interconnect node (IN)". Both root nodes and interconnect nodes establish routing tables to store the routing information.

### 2.2. Establishing Subarea Trees

At first, the definition of "tree node" is the nodes that join in subarea trees. The "depth" of a tree node is defined as the hop count between the tree node and its root node. Root node's depth is zero and the depth of initial nodes and interconnect nodes is null. The information of a tree node at least includes node ID, node type, depth, father node, child-node-list, the number of offspring nodes and routing-list, thereinto routing-list stores the routing information of offspring nodes.

The establishing process as follows:

Root nodes and tree nodes periodically broadcast Subarea Tree Establishing Message (*STEM*) which involves node ID, node type, node depth, number of offspring

nodes and other optional parameters (including number of neighbors, residual energy, transaction capability, stability etc.). Its format as Figure 1.

The initial node which receives *STEM* will set its own node type as “tree node”; set father node as the node ID included in the *STEM*; set own depth as the depth included in the *STEM* plus one. Then it returns Tree Node Updating Message (*TNUM*) to its father node. *TNUM* involves destination node ID, source node ID, number of offspring nodes and node routing-list. Its format is shown in Figure 2. After the father node receives *TNUM*, it will update child-node-list, the number of offspring nodes and routing-list; and continue to send *TNUM* to its father node till root node.

If an initial node receives more than one *STEM*, it will select the optimal according to the selecting regulations and discard the others. Selecting regulations involve node depth, the number of offspring nodes or other optional parameters.

If an initial node cannot join in a subarea tree after a stated time  $T$  ( $T > 0$ ) since it just connects with interconnected nodes. Here one of the interconnected nodes adjacent with this initial node will be upgraded to become a root node and obtain other root nodes’ information from the nearest root node. It also notifies other root nodes that it has become a root node.

At last, all initial nodes will join in subarea trees to become tree nodes. The *STEM* received by root node or interconnect node will be discarded.

### 2.3. Dynamic Maintenance of Subarea Trees

Child node identifies whether its father node still exists or not through detecting the *STEM* sent periodically by its father node, and vice versa.

If father node inspects that the immediate relationship with one child node is already of nonexistence, it will delete this child node from child-node-list; update the number of offspring nodes and routing-list; then inform its father node by *TNUM*.

If child node inspects that the adjacent relationship with its father node is already failure, it will configure its node type as “initial node”; inform all its child nodes to configure their node type as “initial node” and clear relevant parameters by sending Tree Node Releasing Message (*TNRM*). The format of *TNRM* is shown in Figure 3. Nodes whose type is “initial node” restart to join in subarea trees.

MessageType	NodeID	NodeType	Depth	Num of Offspring	Optional
-------------	--------	----------	-------	------------------	----------

Figure 1. The format of STEM.

MessageType	DestID	SourceID	Num of Offspring	Routing-List
-------------	--------	----------	------------------	--------------

Figure 2. The format of TNUM.

MessageType	DestID	SourceID	TreeNodesReleased
-------------	--------	----------	-------------------

Figure 3. The format of TNRM.

### 2.4. Routing Mechanism

From the process of establishing subarea trees, we can see that every root node has acquired the routing information of other root nodes by interconnecting process, and father node has acquired the routing information of all its offspring nodes during the course of establishing a subarea tree. So root node knows the routing information of all the tree nodes in its subarea tree.

Meanwhile, after having finished the process of establishing subarea trees, a hierarchical network structure is formed. There are two tiers: the first tier (namely backbone) consists of root nodes and interconnect nodes; the second tier is each subarea tree. Moreover these two tiers have different routing strategies. The strategy in intra-subarea as well as among root nodes and interconnect nodes is proactive routing; whereas the strategy of inter-subarea is on-demand routing.

Based on the above-mentioned description, the routing mechanism proceeds as follows:

If the destination node of data packet is its own offspring node, the node will forward the packet to the destination directly. If the destination node of data packet is not its own offspring node, the node will forward the packet to its father node. That is to say, there are only two directions-up and down for a tree node to send data packets.

If data packet is sent by interconnected node and no routing information is found, the packet is sent to the nearest root node.

If root node can also not find the routing information, it will send Routing Inquiry Message (*RIM*) to the other root nodes. *RIM* includes message type, destination root node ID, source root node ID, and data packet’s destination node ID. Its format is shown in Figure 4. The root nodes or interconnect nodes which know data packet’s routing information will return Routing Reply Message (*RRM*) to the source root node. The format of *RRM* is same as *RIM*. Note that there is no demand for establishing a new route in this process and just use the established route.

After receiving *RRM*, the source root node forwards the data packet to the root node or interconnect node which knows the optimal routing. Root node or interconnect node will forward data packet to the destination node according to the known routing information after it receives the data packet.

### 2.5. Example

A simplified example is illustrated in Figure 5 and there are 22 network nodes. At first all the nodes are “equal” and each node is “initial node” with a unique ID. The line between two nodes denotes a wireless link and these two nodes can communicate directly.

MessageType	DestRootID	SourceRootID	DataDestID
-------------	------------	--------------	------------

Figure 4. RIM/RRM format.

Assume the condition of becoming a root node in this example is that the number of neighbors is more than or equal 4. Let node S broadcasts *RADRQ* message and nodes A, B, C will become root nodes. These three root nodes will broadcast their information so that they can know each other. Afterwards through existing routing protocol such as AODV or DSR, they will build routing information (routing metrics: hop count or burthen is least in this example):  $A \leftrightarrow B$ ;  $A \leftrightarrow D \leftrightarrow C$ ;  $B \leftrightarrow E \leftrightarrow F \leftrightarrow C$ . During this procedure, nodes D, E, F will become “interconnect node”, illustrated as Figure 6.

Root nodes A, B, C start the establishing process of subarea trees by broadcasting *STEM*. The initial nodes receiving this message will join in subarea trees and become “tree node” simultaneously. Then tree nodes also send out *STEM* periodically. In this example, all the initial nodes except nodes H, I, R will join in subarea trees successfully through such a process. Under this circumstances, interconnect node E will be upgraded to become root node and it will gain other root nodes’ information from the nearest root node B as well as notify other root nodes of its own information. And then nodes E, H, I, R form a new subarea tree. At last the entire network is composed of four subarea trees illustrated as Figure 7 and two tiers of network structure is obvious.

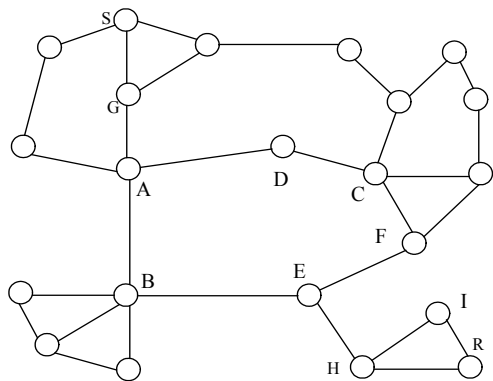


Figure 5. Original network.

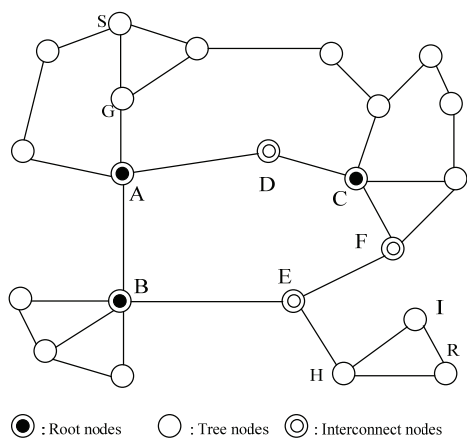


Figure 6. Root nodes and their routing.

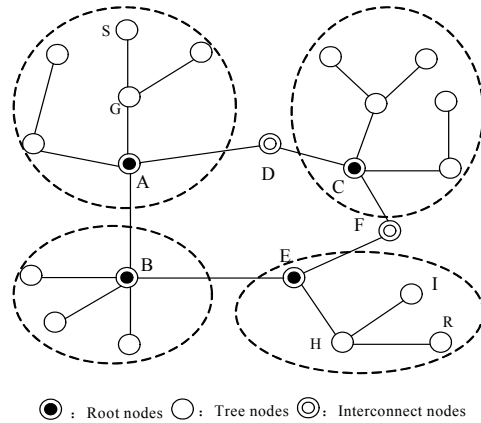


Figure 7. Subarea tree formed network.

Assume node S wants to send data packets to node R. Firstly S sends the data packets to its father node G. Node G finds node R is not its offspring node and sends the data packets to root node A. Root node A will lookup routing information for node R, but it cannot find the route. And then A sends *RIM* to root nodes B, C, E and node E will answer this message with *RRM* because it knows node R is its offspring node. Node A will send data packets to node E while it receives *RRM* from node E. Finally node R can receive the data packets through nodes E and H. The forwarding path is shown in Figure 8.

### 3. The Correctness of STR

According to the definition, initially every node in the network is an initial node. Well then can *STR* act on all these nodes and create a complete network structure after having finished the process of establishing subarea trees? Strictly speaking, this question is concerned with the effectiveness of *STR* and the correctness wants to be proved. The following theorem assures it.

**Theorem 1** We model the network as a graph  $G=(V, E)$ , where V is the set of vertices and E is the set of edges.

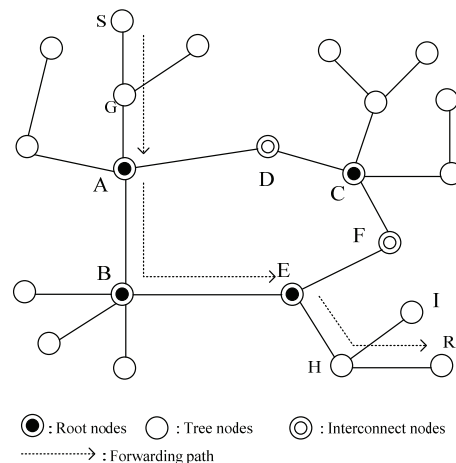


Figure 8. Data forwarding.

Assume  $G$  is connected, and then there are two situations for  $\forall v \in V$  through the process of establishing subarea trees:

- 1)  $v$  belongs to a subarea tree, namely  $\exists T=(V_T, E_T)$  makes  $v \in V_T$ , thereinto  $V_T \subseteq V, E_T \subseteq E$ ; or
- 2)  $v$  is an interconnect node (IN).

*Proof.*

If  $v$  satisfies the condition of selecting root node, then  $v$  becomes a root node; thereupon  $v$  belongs to the subarea tree whose root node is  $v$ .

If  $v$  is not a root node, but a node which interconnect routes of root nodes pass through, then  $v$  is an interconnect node (IN).

If  $v$  is neither a root node, nor a interconnect node, then there are four cases:

- (1)  $\exists$  a root node  $r$ , makes edge  $(r, v) \in E$ ; or
- (2)  $\exists$  a tree node  $t$ , makes edge  $(t, v) \in E$ ; or
- (3)  $\exists$  a interconnect node  $i$ , makes edge  $(i, v) \in E$ ; or
- (4) All the neighbors of  $v$  are initial nodes.

For cases (1) and (2),  $v$  can receive message *STEM* from  $r$  or  $t$  at least and join in a subarea tree because both  $r$  and  $t$  broadcast *STEM* periodically.

For case (3),  $i$  will be upgraded to become a root node when  $v$  cannot join in a subarea tree in a stated time, upon that  $v$  will receive *STEM* from  $i$  and join in a subarea tree.

For case (4), since  $G$  is connected, so  $\exists v_0 \in$  a subarea tree  $T$ , makes

Path  $(v_0, v) = (v_0, v_1, v_2, \dots, v_n, v_{n+1}=v)$  exists, thereinto  $v_j$  ( $j=1, 2, \dots, n, n+1$ ) are initial nodes. The following will prove that  $v$  can receive *STEM* and join in a subarea tree through mathematical induction:

Since  $v_0 \in T$ , so  $v_0$  sends *STEM*.

When  $j=1$ ,  $v_1$  is the neighbor of  $v_0$ ;  $v_1$  can receive *STEM* from  $v_0$  and become a tree node.

When  $j=2$ ,  $v_2$  is the neighbor of  $v_1$ ;  $v_2$  can receive *STEM* from  $v_1$  and become a tree node.

When  $j=n$ , assume  $v_n$  can receive *STEM* and become a tree node.

Thereupon, when  $j=n+1$ , since  $v_{n+1}$  is the neighbor of  $v_n$ , so  $v_{n+1}$  can receive *STEM* from  $v_n$  and join in a subarea tree. Namely  $\exists T=(V_T, E_T)$  makes  $v \in V_T$ .

## 4. Performance Analysis

### 4.1. Simulation Model

We used NS2 (Network Simulation 2) to evaluate the performance of *STR*. The simulation modeled a network in a  $2500m \times 2500m$  area with 50 mobile nodes. Radio transmission range is 250 meters. The mobility of each node is arranged from 2m/s to 8m/s, and the pause time of the mobile nodes is zero. Traffic sources are continuous bit rate (CBR) with the rate of 15kbit/s. The source-destination pairs are randomly selected over the network.

Four important performance metrics are evaluated: (1) Normalized routing load-the number of routing control packets transmitted per data delivered at the destination. Each hop-wise transmission of a routing control packet is counted as one transmission. (2) Route discovery delays-the delay between a route requests being issued and a reply with a valid route being received. (3) Route discovery load-the route discovery packets being used to find a valid route to the destination. (4) Packet delivery ratio-the ratio between the number of received data packets and those originated by the sources.

### 4.2. Simulation Result

In our experiments, we compare the performance of *STR* with that of AODV. Figure 9 reports the normalized routing load. For small number of pairs (say 10), we can see that the normalized routing load increases with the mobility of each node in both two protocols. And the performance of AODV is little better than *STR*. *STR*'s poor performance can be attributed to route control messages used to maintain the connection between child-father pairs. However, with the increase of source-destination pairs (say 30); the normalized load of AODV is higher than *STR*, especially when the mobility speed increases to a high level. This is because AODV has a much higher O/H than *STR*.

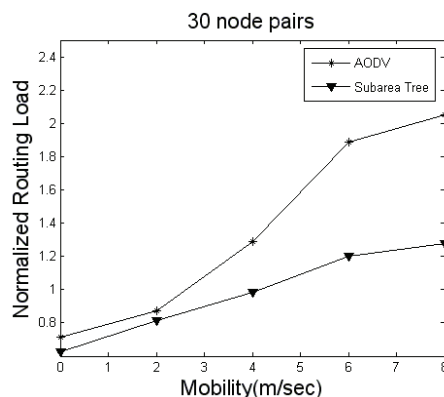
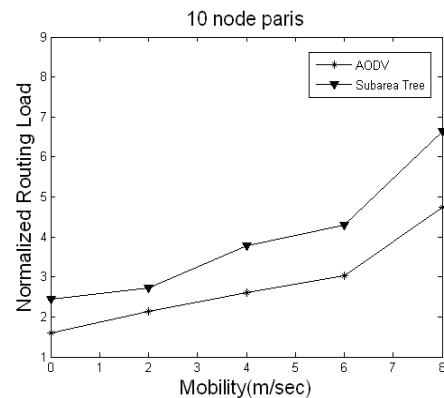


Figure 9. Normalized routing load.

In Figure 10, the route discovery delays are reported. Here we assume that all source-destination pairs are in different subarea and the node speed is fixed at 2m/s. When the hop count of the source-destination is small, the performance of these two protocols is quite similar. The route discovery delays increase with hops between source and destination. However, we can see from the figure that AODV has a higher increase rate than *STR*. Because, in *STR* protocol, route discovery process only need to find a route within root nodes. On the other hand, if the source-destination pairs are in the same subarea, there will be no route discovery delays by using *STR* protocol.

Experiment in Figure 11 shows the effect of distance between source and destination on route discovery load. Clearly in both protocols the route discovery load is increased with distance. At the beginning, the route discovery load of *STR* is almost zero, since when source-destination pair is close to each other, they are usually in the same subarea, no route packet need to be used. The poor performance of AODV is because that it uses an expanding ring search technique to disseminate the RREQs, while in *STR* only the root nodes and interconnect nodes need to handle the Routing Inquiry Message (*RIM*).

Figure 12 shows the packet delivery ratio of *STR* under various offered load with different speeds. Packet delivery ratio declines both with speeds and offered load. We note that at low speeds, packet delivery ratio is sensitive with offered load. Packet delivery ratio declines while

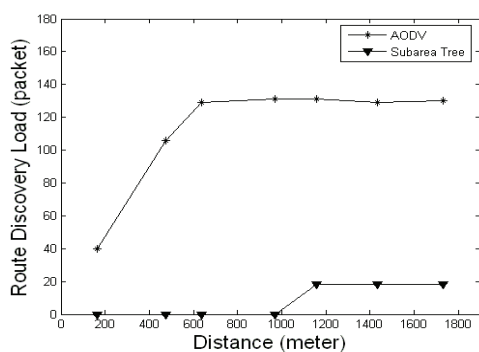


Figure 10. Route discovery delay.

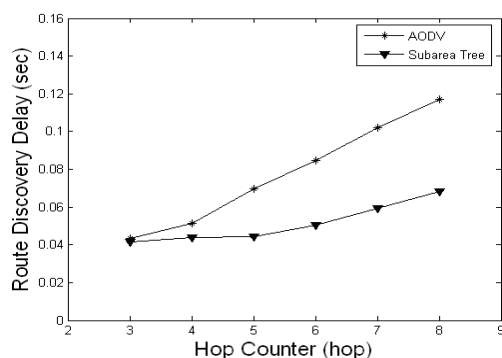


Figure 11. Route discovery load.

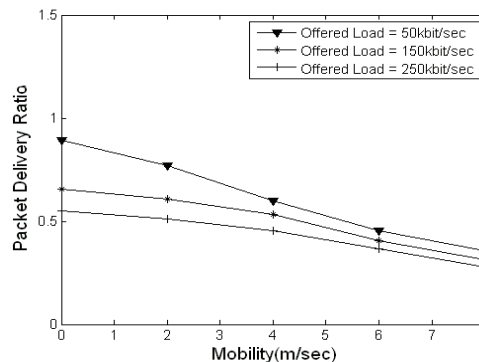


Figure 12. Packet delivery ratio of *STR*.

offered load is increasing. This is because most data packets are sent to their father node before they go to the destination in *STR*. And it will probably lead to traffic congestion in some father nodes and cause packet loss. As mobility increase, the packet delivery ratio is less sensitive with offered load. Since most packet loss ascribe to the fiercely change of network topology.

## 5. Conclusions

*Subarea Tree Routing (STR)*, a novel routing protocol for multi-hop wireless ad hoc networks based on the idea of establishing subarea trees and dividing the whole network into many logical subareas, has been proposed. This protocol constructs a hierarchical network structure with two tiers in which different routing strategy is adopted, and routing mechanism combines the advantages of proactive routing and on-demand routing. Indeed, since the extents of proactive routing are restricted in intra-subarea as well as among root nodes and interconnect nodes, *STR* does not incur heavy overhead due to maintaining routing information, especially in large, mobile, ad hoc environment. Furthermore, since on-demand routing is operated only between root nodes whose number is very small in general, the delay due to route searching is also low. Theorem 1 assures the correctness of *STR* and our ns-2-based simulation has confirmed the advantages of *STR* and demonstrated a significant routing efficiency and scalability improvement.

## 6. References

- [1] C. E. Perkins, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," Proceedings of ACM SIGCOMM, pp. 234-244, 1994.
- [2] G. Pei, M. Gerla, and T. W. Chen, "Fisheye state routing in mobile ad hoc networks," in Proceedings of the 2000 ICDCS Workshops, Taipei, Taiwan, pp. D71-D78, April 2000.
- [3] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," Proceedings of IEEE International Conference on Network Protocols (ICNP), pp. 273-282, 1999.
- [4] G. Y. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark routing for large scale wireless ad hoc networks with



- group mobility,” Proceedings of IEEE/ ACM Workshop on Mobile Ad hoc networking & Computing, MobiHOC'00, Boston, MA, USA, MA, USA, pp. 11–18, 2000.
- [5] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR),” IETF Internet RFC3626, October 2003.
- [6] R. Ogier, F. Templin, and M. Lewis, “Topology broadcast based on reverse-path forwarding (TBRPF),” IETF Internet RFC3684, February 2004.
- [7] C. E. Perkins, E. M. Belding-Royer, and I. Chakeres, “Ad hoc on demand distance vector (AODV) routing,” IETF Internet RFC3561, July 2003.
- [8] D. Johnson, Y. Hu, and D. Maltz, “The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4,” IETF Internet RFC4728, February 2007.
- [9] V. Park and S. Corson, “Temporally-ordered routing algorithm (TORA) Version 1 functional specification,” IETF Internet draft (draft-ietf-manet-tora-spec-04.txt), July 2001.
- [10] C. K. Toh, “Associativity-based routing for ad hoc mobile networks,” WLPersonal Communications Journal, Special Issue on Mobile Networking and Computing Systems, Kluwer, Vol. 4, No. 2, pp. 103–139, March 1997.
- [11] Z. J. Haas and M. R. Pearman, “The performance of query control schemes for the zone routing protocol,” IEEE/ACM Transactions on Networking, Vol. 9, No. 4, pp. 427–438, August 2001.
- [12] C. C. Chiang and M. Gerla, “Routing and multicast in multihop, mobile wireless networks,” Proceedings of IEEE ICUPC '97, San Diego, CA, pp. 546–551, October 1997.
- [13] G. Pei, M. Gerla, and X. Y. Hong, *et al.*, “A wireless hierarchical routing protocol with group mobility,” Proceedings of IEEE WCNC'99, New Orleans, LA, pp. 1538–1542, September 1999.