

# Route research for Ad hoc network

**Zhengxi Wei**

School of Computer Science, Sichuan University of Science & Engineering, Zigong Sichuan 643000, PR China

**Email address:**

413789256@qq.com

**To cite this article:**

Zhengxi Wei. Route Research for Ad Hoc Network. *American Journal of Networks and Communications*. Vol. 3, No. 6, 2014, pp. 73-76.  
doi: 10.11648/j.ajnc.20140306.11

---

**Abstract:** Routing delivery is one of the key techniques in mobile ad hoc network, which involves many aspects such as topology update, routing exchange and so on. In this paper, we firstly apply routing tree to describe the complete topology of ad hoc network, and then store the information of routing tree into every node's routing table. Sparse routing tree, as a pruned tree, is used to exchange topology information, which reduces the header length of intranet packets. By multicasting packets in the intranet, the precious bandwidth over wireless link is saved greatly. We finally present a routing algorithm to select a most economical path among potential routing paths.

**Keywords:** Source Directed Route, Routing Tree, Topology Update, Ad Hoc Network

---

## 1. Introduction

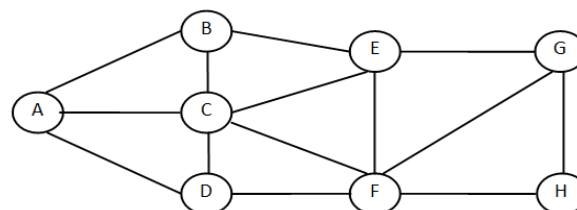
Mobile ad hoc networks play an important rule in military units or unmilitary fields. As well-known, traditional routing protocols are developed to support user communication in networks with a fixed infrastructure and reliable, high-capacity links. However, in mobile ad hoc network, the network infrastructure is dynamically changing, and the links are wireless with less capacity and more prone to errors. So how to exchange topology for each node is the primary one of a lot of crucial problems.

Many routing protocols [1] are designed to accurately determine network topology. This information is translated into routing tables, which are used to route information packets. If the routing table is inaccurate, packets will be misrouted, causing small delays at best, packet loss at worst. Ideally, a routing protocol should quickly identify changes in network topology to minimize the time during which information packets might be misrouted. Furthermore, the routing algorithm itself has impact on network performance since it must use network resources to determine the topology. It is so easy to construct an adaptive routing protocol which quickly and accurately determines the best routes, but leaves no bandwidth for user traffic. This case should be avoided as much as possible in routing algorithm.

In general, there are some effective routing algorithms available in ad hoc networks. Some based on active routing idea, e.g., DSDV [2] (Destination-Sequenced Distance Vector), WRP [3] (Wireless Routing Protocol), etc. The others

based on on-demand routing idea, such as DRS [4] (Dynamic Source Routing), AODV [5] (Ad hoc on Demand Distance Vector), and so on. Pointing to the characteristics of ad hoc network, these algorithms are to some extent available to frequent topology changes so as to improve the routing-search performance in the wireless self-organized network. The selection of algorithms depends on the actual network. However, there also exist many disadvantages including much waste of network sources (such as bandwidth, power supply, CPU resource), long time delay to wait for routing packet, and routing protocol always not being convergent state etc.. In order to save the precious bandwidth resource in wireless communication and reduce the transmission network traffic, this paper presents an algorithm based on routing tree, which is applied to source directed route in ad hoc network. It works well in some specific situations.

## 2. Problem Overview



**Figure 1.** Wireless links of network

Figure 1 shows the topology in ad hoc network at one time

[6]. Any node in this network can communicate with its adjacent one. If a node wants to communicate with nonadjacent nodes, this node has to rely on other nodes to relay packets. For example, there are no direct connective paths between wireless links: B—F, D—E and E—H. As a result, they have to get help from relay nodes to transmit information for each other.

Figure 1 may reflect the following actual situation. Each node labeled A through H could be considered to be a computer with an associated communication processor. The lines between two nodes indicate the connectivity in the whole network. Suppose the nodes know nothing about neighbor nodes that are more than 1 hop away, they need to exchange connectivity information.

Therefore our research problem is how to build up a more complete view of the network link's topology for every node and select a path from a source node to any other destination one.

### 3. Routing Tree and Routing Table

#### 3.1. Routing Tree

Each node should store topology information as a routing tree graph. Figure 2 shows the routing tree for nodes A and C. They are prior to the exchange of any topology information. The routing trees for A and C contain only their nearest neighbors, the nodes which they can communicate directly. Similar graphs exist for all other nodes. This kind of data structure is called routing tree. The data structure of routing tree is defined as follows.

```
Type RouteTree=^tree
Tree=Record
[   child: RouteTree;
    info: elemtp;
    next: RouteTree;
    parent: RouteTree
];
t=RouteTree;
```

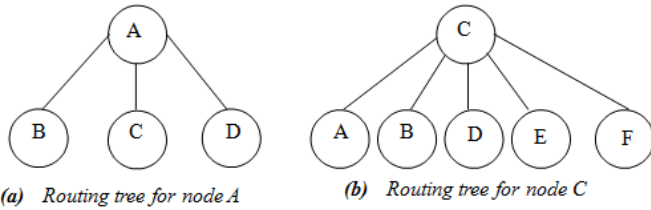


Figure 2. Routing tree for node A and C

As shown above, three pointer fields are created and they are used to connect the other adjacent nodes. However, the “info” field includes a good many sub-options, for example the label of node, the number of links (that indicates the distance between a node and its root node). We could also add any additional options to tree's info field if need. In this way routing tree' info is merged into a big field by a number of

subfields. The data structure of a routing tree node is shown in Figure 3.

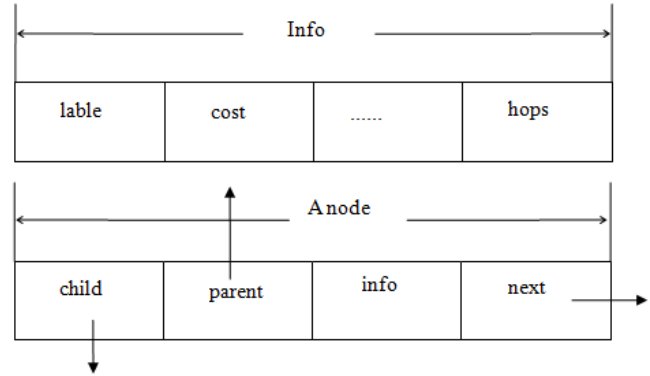


Figure 3. Data structure of routing tree

The corresponding relation between routing tree and its logic storage structure is created and shown in Figure 4. This design method builds up a map from a whole routing tree to its structure in storage.

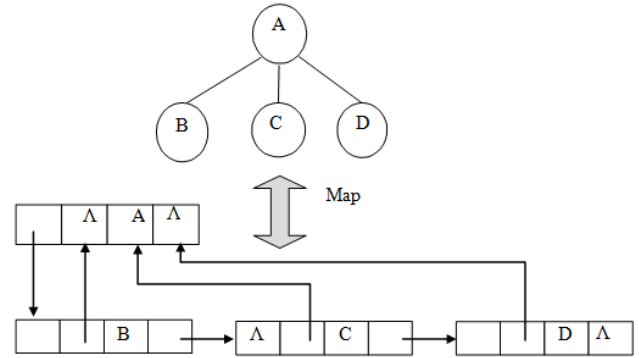


Figure 4. Routing tree and its storage structure

Each node should store topology information as a routing tree graph. Figure 5 shows the routing tree [7] for node A. Node A and its neighbor nodes are prior to the exchange of any topology information. The routing trees for A contain only their nearest neighbors, the nodes which they can communicate directly. However, the neighbor nodes also keep information of their own adjacent nodes. All topology graphs are merged into a kind of tree graph called routing tree.

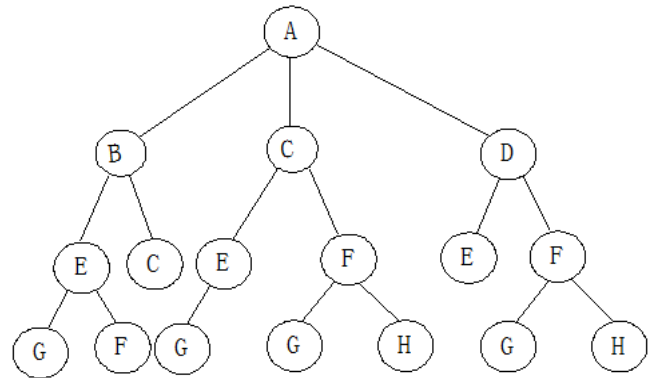


Figure 5. Routing tree for node A

### 3.2. Routing Table

The routing table for node A is shown in Table 1. It assumes some nodes are in quiet mode (namely, they must not have any subsequent branches in the routing tree), all nodes can participate in relay, and all links have a cost of 1. The actual link layer addresses for the nodes would be placed into the table in place of the symbols A, B, C, and so on.

And the routing table would always be encoded into the intranet header for topology updates.

Table 1. Routing table for node A

Node Address	Node Predecessor	Hop	Cost	Not Relay	Quiet
B	A	1	1	0	0
C	A	1	1	0	0
D	A	1	1	0	0
E	B	2	1	0	0
C	B	2	1	0	0
E	C	2	1	0	0
F	C	2	1	0	0
E	D	2	1	0	0
F	D	2	1	0	0
G	E	3	1	0	1
F	E	3	1	0	1
H	F	3	1	0	1
G	F	3	1	0	1

## 4. Topology Update

### 4.1. Sparse Routing Trees

A full routing tree provides enough topology information, and full topology information of ad hoc network can be obtained by exchanging full routing trees. However, the amount of data in the routing tree would become very large, especially for fully connected networks. The number of links in a fully connected net with  $n$  nodes is  $n(n-1)/2$ . Although full routing trees should be stored by a mobile node, exchanging these routing trees may consume too much bandwidth. A smaller copy of the full routing tree (It is defined a sparse routing tree) should be prepared for transmission to neighbor nodes.

To reduce the number of branches in the routing tree, the repetitive paths of the nodes in the tree are pruned according to following rules:

- 1) Only the shortest paths from the root node to another node are retained
- 2) If there're redundant paths which are the same number of links from a root node to another node in the routing tree, at most 2 paths are retained.

After being pruned, the final routing tree is called sparse routing tree. For Node A, its sparse routing tree is shown in Figure 6. Nodes in the network gain more topology information by multicasting their sparse routing trees to their nearest neighbor nodes. This exchange of routing trees will percolate more complete topology information through the network.

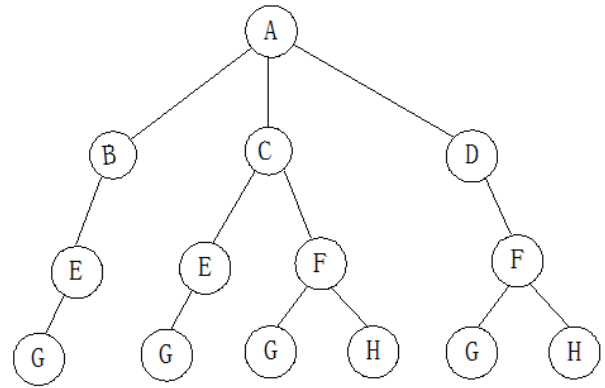


Figure 6. Sparse routing tree for node A

### 4.2. Update Routing Table

The pruning rules stated above have not been violated. They have been applied to the entries in the sparse routing table. The sparse routing graph is deduced from the table. Thus, quite a few redundant paths can be removed from the sparse routing table. Having been pruned, the topology table for node A is shown in Table 2.

Table 2. Sparse routing table for node A

Node Address	Node Predecessor	Hop	Cost	Not Relay	Quiet
B	A	1	1	0	0
C	A	1	1	0	0
D	A	1	1	0	0
E	B	2	1	0	0
F	C	2	1	0	0
E	C	2	1	0	0
F	D	2	1	0	0
G	E	3	1	0	1
H	F	3	1	0	1
G	F	3	1	0	1

Note that figure 2 shows more than 2 paths between nodes G and A; however, the sparse routing table, which is the information actually transmitted, shows only two entries for nodes G. Topology update packets are transmitted exclusively using a global multicast address.

## 5. Routing Algorithm

Source Directed Route provides a simple non-dynamic procedure for relaying a packet from an originator to one or more destinations. The source must calculate the path through the intranet network to reach each destination. These paths are based on the topology and routing table.

The specific source directed route for each destination must be encoded into the intranet header. If the routes for two or more destinations share common links along the paths, the two paths should be merged together.

We presents the routing algorithm based on routing tree is going to select paths from an originator to one or more destinations. For example, we will select multicasting routing paths for node A in Figure 6. In the example, node A wishes to deliver a packet to nodes D, E, F, G and H. In this case node A

again would select the most economical path to each destination, taking into consideration the impacts on network traffic and intranet header size. Assuming that all paths have the same quality and cost, any path may be selected by Node A. The routing algorithm has been designed with the following steps.

Traverse the routing tree and save all potential paths from node A to node Di (Di represents any destination node).

Count the number n, which denotes how many subsequent destination nodes the father node j in a potential routing path includes, and save n into an array Di[j].

Select a node leading to Di by the following principle.

a. Scan nodes of the second layer in routing tree, assign  $k := \max\{Di[B], Di[C], \dots, Di[D]\}$ , and figures out the number maxValueAmount, which denotes how many elements in  $\{Di[B], Di[C], \dots, Di[D]\}$  their values equal to k, namely,  $\text{maxValueAmount} := \text{count}(Di[j]=k)$ .

b. if  $\text{maxValueAmount}=1$ , the selected node is the one whose  $Di[j]=k$ .

c. if  $\text{maxValueAmount} \geq 2$ , these nodes whose subsequent branches include destination nodes, are prior to be selected. For example, node F is both a destination and a relay while node C is a relay node only. Although  $DF[C] = DF[D] = 4$ , the final path A—D—F is selected.

d. if potential paths all include destination nodes, similar searching process will be applied to nodes of the third layer or even the higher layer in routing tree according to the above rules until it comes to the highest layer. If maxValueAmount is always greater than or equal to 2, searching process will stop and determine any one of remnant potential paths.

Table 3 lists the potential and selected paths from node A to each of the intended destinations. All items are figured out according to above routing algorithm. What's more, timing topology update in the wireless network ensures the validity of the routing table.

The selected paths are the best links on which node A transmits messages to the other network nodes in a time. Multicast information on these links may make the channel reuse rate the maximum. As a result, it is going to decrease the transmission delay to a minimum, reduce the transmission network traffic, and save a lot of valuable bandwidth resources over wireless networks.

**Table 3.** Final routing path for node A

Destination Node Di	Potential Paths	Selected Path
D	A—D	A—D
E	A—B—E A—C—E	A—C—E
F	A—C—F A—D—F	A—D—F
G	A—B—E—G A—C—E—G A—C—F—G A—D—F—G	A—D—F—G
H	A—C—F—H A—D—F—H	A—D—F—H

## 6. Conclusion

In this paper, we have discussed source directed route in ad

hoc network, which are based on network topology and routing table. Compared with the other methods, it reduces the length of packets by using sparse routing tree and saves the precious bandwidth over wireless link by multicasting packet in intranet.

In particular, our routing method has the following advantages. Firstly, the adaptive topology update makes the establishment of the network, network maintenance, network recovery and other tasks fully automatically by mobile nodes without manual configuration. Secondly, sparse routing tree, sparse routing table as well as their simple data structure are adopted so that the additional costs that topology updates give rise to are small and take up less bandwidth. Finally, the source routing mode allows multiple paths to choose along the nodes, enhances the robustness of the network, and avoids the route 'closed loop' occurring.

We have applied the research result to a project about military tactical Internet successfully. After a lot of repeated actual tests, the practice proves this kind of route works well in ad hoc network.

## Acknowledgements

The research was supported by Artificial Intelligence Key Laboratory of Sichuan Province (No. 2013RYY04) and the Sichuan Provincial Education Department's Key Project (No.14ZA0210).

Our work was also supported by university Key Laboratory of Sichuan Province (No. 2013WYY09) and Fund Project of Sichuan Provincial Academician (Experts) Workstation (No.2014YSGZZ02).

## References

- [1] Royer Em, Chai\_Keong T. A review of current routing protocols for ad hoc mobile wireless network[J]. IEEE Personal Communications, 1999, 6(2): 46-55.
- [2] Perkins C, Bhagwat P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers[Z]. ACM SIGCOMM'94, London, 1994. 234-244.
- [3] Murthy S, Garcia-Lunes-Aceves J. An efficient routing protocol for wireless networks[J]. ACM Balzer Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communications Networks, 1996, 1(2): 183-197.
- [4] Johnson DB, Maltz DA.. Dynamic Source Routing in Ad Hoc Wireless Networks. New York: Kluwer Academic Publishers, 1996. 153~181.
- [5] Perkins C, Royer E. Ad-Hoc on Demand Distance Vector Routing[Z]. The 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, 1999. 90-100.
- [6] MIL-STD-188-220B[S]. Department of Defense, Washington, 1998.
- [7] YAN WEI-min, WU Wei-min. Data Structure[M], Second Edition, Beijing: Tsinghua University Publishing Company, 2000.