Journal of Universal Computer Science, vol. 14, no. 7 (2008), 1105-1117
submitted: 1/10/07, accepted: 21/1/08, appeared: 1/4/08 © J.UCS

A Knowledge Discovery Agent for a Topology Bit-map in Ad Hoc Mobile Networks

SungSoo Lee
(Department of Computer Engineering, Yeungnam University, Korea
lssung@chol.com)

HangKon Kim
(Department of Computer Information Communication, Catholic University of Daegu, Korea
hangkon@cu.ac.kr)

ChongGun Kim*
(Department of Computer Engineering, Yeungnam University, Korea
cgkim@yu.ac.kr)

Abstract: A central characteristic of ad hoc mobile networks is the frequent changes of their topology. This is the source of many problems that need to be solved. AODV is an on-demand routing protocol for decreasing maintenance overhead in ad hoc networks. However, some path breaks can cause significant overhead and transmission delays. If the maintenance overhead of the routing table can be reduced, table-driven routing methods could be an efficient substitution. In this paper, we propose a knowledge discovery agent for an effective routing method that uses simple bit-map topology information. The agent node gathers topology knowledge and creates topology bit-map information. All available paths from a source to a destination can easily be calculated using the bit-map. All the nodes in the network maintain the bit-map distributed from the agent, and use it for the source of routing. The performance and the correctness of the proposed agent method is verified by computer simulations.

Keywords: knowledge discovery agent, ad hoc networks, AODV, table-driven routing, bit-map table, reducing overhead
Categories: C.2.1, C.2.2, C.2.3

1 Introduction

The ad hoc network does not rely on existing infrastructure and uses nodes as hosts and routers to transmit packets. With its frequent changes of topology, the ad hoc network requires a special routing algorithm. Routing protocols used in ad hoc networks can be separated into two categories: the table-driven routing method [Perkins et al, 94] and the on-demand routing method [Lu et al, 03].

In the table-driven routing methods every node maintains information about the routing of every node in the network. Nodes perform the routing based on this information. The advantage of this method is that nodes can establish the path without discovering the route on demand. However, periodic exchange of information between nodes for routing information is inefficient with respect to transmission

* Corresponding author

bandwidth and the nodes' energy. It creates another type of overhead. Typical table-driven routing protocols are DSDV (Destination-Sequenced Distance Vector Routing Protocol) [Perkins et al, 94], OLSR (Optimized Link State Routing Protocol) [Clausen et al, 94], and TBRPF (Topology Broadcast based on Reverse-Path Forwarding Protocol) [Ogier et al, 03].

In the on-demand routing method, nodes start the path discovery process when data needs to be transmitted. The advantage of this method is that nodes do not periodically exchange information about the routing paths in the network, which significantly increases overhead. Protocols which use the on-demand routing method are AODV (Ad hoc On-demand Distance Vector) [Perkins et al, 03], DSR (Dynamic Source Routing) [Johnson et al, 03], R-AODV (Reverse AODV) [Kim et al, 06], and etc. [Nasipuri et al, 01].

Within an ad hoc network configured by multiple mobile nodes, an agent node that represents the network could control the wireless network. The agent can be broadly used to organize a tree type network [Vasudevan et al, 04]. Because of overhead in the table-driven routing method, caused by managing routing information, the on-demand routing method is mostly being studied. If this overhead could be decreased by some means, it would be practical to apply the table-driven routing method.

In this paper, we propose a knowledge discovery agent for simple and efficient link state routing, which uses bit-map information. All nodes in the network use identical bit-map information distributed by the agent. The bit map representing the network topology can be used for deciding routing paths from a source to a destination, at any instant. The agent collects network topology information and creates the bit-map information which represents the network topology. The agent node can be dynamically decided using node connectivity and battery life-time information for network stability.

2 The Knowledge Discovery Agent for Topology

In this section, the role and the maintenance method of the agent for topology are described.

2.1 Bit-map Creation and Delivery

The agent node gathers information and delivers the knowledge that represents the network topology, in the form of a bit-map table, to all nodes. Using the bit-map table, the overhead of maintaining routing tables can be decreased. Figure 1 shows the concept of a knowledge discovery agent. The main role of the agent is to gather and distribute network toplogy knowledge. Selecting an efficient agent is another important issue.

All nodes periodically broadcast a ‘Hello’ message in order to discover neighbouring nodes. The agent node broadcasts a query message to all neighbouring nodes to collect information about the network topology. The node transmitting the topology query message becomes the parent-node, and the receivers become the child-nodes.

The node which has received a topology query message delivers it to its child-nodes. If it has no child-node, i.e. it is a leaf node, then it sends a reply message to the parent-node which includes the following: node ID, battery life-time, and connection information. By repeating this sequence the agent node collects all connection information about nodes. After finishing the query process, the agent node creates topology knowledge using all this information, in the form of a bit-map table.

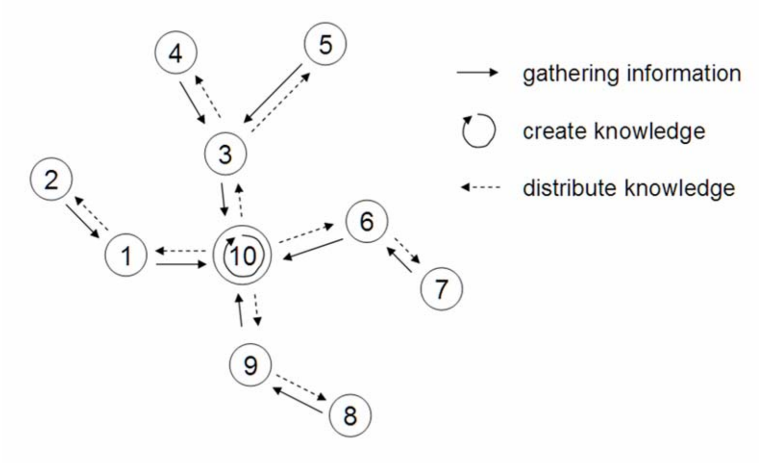

Figure 1: Knowledge Discovery Agent

In the network shown in Figure 2, we assume the agent node. Node 1 is the node that collects connection information about the network and creates the topology knowledge in the form of the bit-map table. Each node in Figure 2 has node ID, battery life-time, and neighbouring node information.

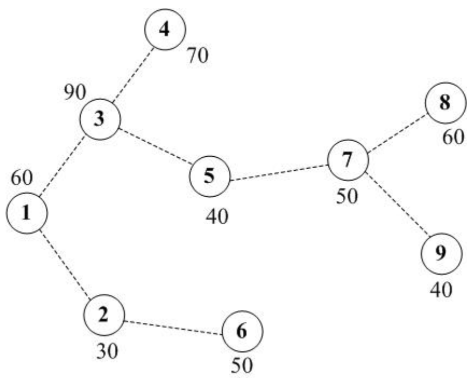

Figure 2: A network topology

The network topology represented in the form of a bit-map table is shown in Table 1, where “1” and “0” stand for “one-hop distance connection” and “non one-hop distance connection” respectively.

If a new node enters or leaves the network or there is any alteration of the network topology, all neighbouring nodes that have sensed the change send a topology reply message to the agent node. The agent node has to maintain the bit-map table and transmit the updated bit-map table information to all other nodes.

When the agent node is removed from the network, a new agent node has to be selected [Vasudevan et al, 04]. The agent node elects candidate substitute agent nodes, using all nodes connectivity and battery-life time information [Lee et al, 07]. The new agent node information can be broadcasted via a topology advertisement message to all nodes. By this means every node in the network is informed about the existence of the agent, and all connections in the network. Any arbitrary node can be the agent node when the network is initially created. In this initial case, the selection process for the proper agent node is the main requirement.

Node ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	1	1	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	0	0	1	1	0	0	0	0
4	0	0	1	0	0	0	0	0	0
5	0	0	1	0	0	0	1	0	0
6	0	1	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	1	1
8	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	1	0	0

Table 1: Bit-map table for network topology shown on Figure 2

2.2 Agent Selection

The agent selection algorithm is based on the leader election algorithm [Vasudevan et al, 04]. We propose a list of candidate agents, instead of just one agent to be maintained in every node. Each node has an agent list of five nodes (in descending order), where the first node is considered to be the active agent of the network. When the agent node is removed, the highest-valued–node from the listed candidates is selected as the substitute node [Lee et al, 07].

Network partitioning and merging can also be managed efficiently by our proposed method.

3 Bit-map Based Routing Protocol

All nodes in the network send a periodic ’Hello’ message in order to confirm the existence of neighbouring nodes [Chakeres et al, 02]. The format of this ‚Hello’ message is shown in Figure 3.

Node's IP Address
Node's Sequence Number
Life time

Figure 3: ‘Hello’ message format

After receiving a ‘Hello’ message from neighbouring nodes, a node constructs the neighbour information table. The topology information table consists of information about neighbouring nodes and the node's information.

Node ID	Neighboring node	Battery	Number of neighboring nodes	Agent (Yes/No)
3	1,4(0),5(0)	90	3	

Figure 4: Neighbour information table of node 3

Figure 4 shows the information table collected by an intermediate node,3, based on Figure 2. In the network shown in Figure 2, node 1 is the agent and node 3 can have the following information in its table:

a. Receives topology query from node 1.
b. Assigns node 1 as the parent-node, and nodes 4 and 5 as child-nodes.
c. Broadcasts a topology query message to child-nodes, i.e. to nodes 4 and 5.

To create the network topology, the agent node broadcasts a topology query message to all neighbouring nodes. Figure 5 shows the format of the query message. By this means it constructs a self-centred topology tree. The node which has received the topology query message assigns the node which has sent the message to be the parent-node, and all other neighbouring nodes are assigned to be child-nodes.

After receiving a topology query message, a node sends a topology reply message to the parent-node, after gathering sub-tree information. The message contains the node’s ID, battery life-time and connectivity information.

<table>
<tr><td>Type</td><td>T_query ID</td><td>Reserved</td></tr>
<tr><td colspan="3">Agent IP address</td></tr>
<tr><td colspan="3">Agent sequence Number</td></tr>
</table>

Figure 5: Topology query message format

After receiving all topology reply messages from its child-nodes, a parent node records all information in the topology information table and transmits the information to its parent-node. Figure 6 shows the format of a topology reply message.

The parent-node sends a topology reply message to its parent-node only after it receives topology reply messages from all its child-nodes. Using this method, the agent node can accumulate topology information for all nodes in the network.

<table>
<tr><td>Type</td><td>T_reply ID</td><td colspan="2">Reserved</td></tr>
<tr><td colspan="4">Node's IP address</td></tr>
<tr><td>Battery</td><td colspan="2">Connectivity</td><td>Agent</td></tr>
<tr><td colspan="4">Neighboring Node's IP address</td></tr>
<tr><td>Battery</td><td colspan="2">Connectivity</td><td>Agent</td></tr>
<tr><td colspan="4">.
.
.</td></tr>
<tr><td colspan="4">Agent IP address</td></tr>
</table>

Figure 6: Topology reply message format

Nodes 4, 6, 8 and 9 in Figure 2 have no child-nodes, thus, they will promptly send topology reply messages to their parent-nodes. The topology information table of node 5, after the node receives a topology reply message, is shown on Figure 7.

Node ID	Neighboring Node	Battery	Connection information	Agent
5	3(1), 7(0)	40	2	
7	5(1), 8(0), 9(0)	50	3	
8	7(1)	60	1	
9	7(1)	40	1	

Figure 7: Topology information table of node 5

After receiving topology information, the agent node constructs a bit-map table. Based on the bit-map table, the agent node can be changed according to the following rules:

a. The intermediate node is the preferred agent node.
b. The agent node should have the longest battery life-time.
c. The agent node should have as many neighboring nodes as possible.

From the application of the above rules, node 3 is the best agent node. Nodes 1, 7, 5 and 2 are candidates. In this case, the initial agent node delivers the collected bit-map table, which represents the network topology, to the new agent node.

The agent node sends the bit-map information to all nodes in the network, using a topology advertisement message. Figure 8 shows the topology advertisement message format.

Nodes that have received the topology advertisement message check whether there are any differences between this and previous topology information. Then, the node updates the topology table. The agent node periodically broadcasts the topology advertisement message to all other nodes, which need recent information about the network topology and the agent node.

Type	T_adv ID	Reserved
Agent node ID		
1st candidate-agent node ID		
2nd candidate-agent node ID		
3rd candidate-agent node ID		
4th candidate-agent node ID		
5th candidate-agent node ID		
Bit-map table		

Figure 8: Topology advertisement message format

When a node needs to transmit data to another node, it checks a routing path to the destination node by using the bit-map table. It can quickly establish the connection. The bit-map table represents a graph which shows the network topology. The pseudo algorithm for the path search process is illustrated in Figure 9.

```
step 1:     Perform "AND" operation with source node
     and destination node.
   1.A.     If there is a path to the destination
       node, then go to step 3
   1.B.     If there is no path, then go to step 2
step 2:     Find neighboring node.
   2.A.     Select an unchecked neighboring node,
       perform "AND" operation with destination.
          2.A.a.  Find a path to destination, then go
              to step 3.
          2.A.b.  If there is no path, go to step 2-A.
   2.B.     When all neighbors are checked, then
       select a neighbor and perform step 2
       recursively.
step 3:     Use searched nodes for the path, and
     finish.
```

Figure 9: Algorithm of path discovery using "AND" operation

Initially, the source node performs an “AND” operation with itself and the destination node. If the source node finds a connection between itself and the destination node, then the connection is established immediately.

If no connection is found between the source and the destination node, the source node searches for neighbouring nodes. The neighbouring nodes become intermediate nodes which act as a relay to the destination. Then, the “AND” operation is repeated with the neighbouring nodes and the destination node. If no connection is found between the neighbouring nodes and the destination node, the neighbouring node searches for the neighbours of the neighbouring nodes. Then, the “AND” operation is performed with the destination node. This process is continued until a connection to the destination is found. The process is very similar to the BFS (Breadth First Search) method in tree networks.

The method explained above is used for finding a path from the source to the destination node. A multipath can also be established between the source and the destination nodes. The other paths could be assigned to be reference paths.

4 Verification of Route Discovery

To verify the correctness of the proposed agent method, an example ad hoc network is shown in Figure 10. The created bit-map is shown in Table 2. The agent can be decided by using the method described in section 2.2.

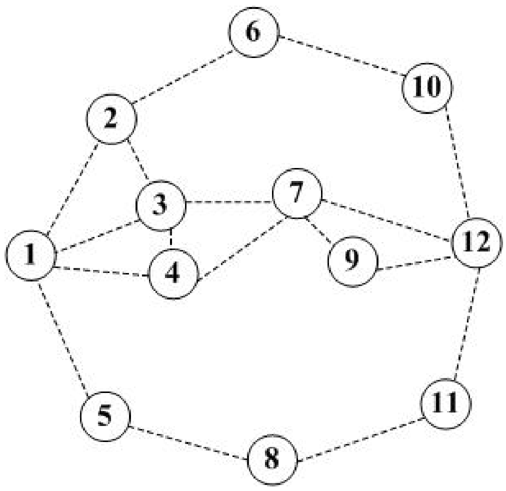

Figure 10: An experimental network topology

Node ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0	1	1	1	1	0	0	0	0	0	0	0
2	1	0	1	0	0	1	0	0	0	0	0	0
3	1	1	0	1	0	0	1	0	0	0	0	0
4	1	0	1	0	0	0	1	0	0	0	0	0
5	1	0	0	0	0	0	0	1	0	0	0	0
6	0	1	0	0	0	0	0	0	0	1	0	0
7	0	0	1	1	0	0	0	0	1	0	0	1
8	0	0	0	0	1	0	0	0	0	0	1	0
9	0	0	0	0	0	0	1	0	0	0	0	1
10	0	0	0	0	0	1	0	0	0	0	0	1
11	0	0	0	0	0	0	0	1	0	0	0	1
12	0	0	0	0	0	0	1	0	1	1	1	0

Table 2: Bit-map table for network shown in Figure 10

Node 1 is set as the source node. To find all destinations, a bit-map operation is performed using the algorithm of Figure 9. The results of the operation are shown in Figure 11. Paths are established for all destinations and there are multiple paths with a different number of hops for the same destination.

```
NODE 1
Input SRC & DST: 1 2   1 -> 2

Input SRC & DST: 1 3   1 -> 3

Input SRC & DST: 1 4   1 -> 4

Input SRC & DST: 1 5   1 -> 5

Input SRC & DST: 1 6   1 -> 2 -> 6

Input SRC & DST: 1 7   1 -> 3 -> 7,    1 -> 4 -> 7

Input SRC & DST: 1 8   1 -> 5 -> 8

Input SRC & DST: 1 9   1 -> 3 -> 7 -> 9,       1 -> 4 -> 7 -> 9

Input SRC & DST: 1 10  1 -> 2 -> 6 -> 10

Input SRC & DST: 1 11  1 -> 5 -> 8 -> 11

Input SRC & DST: 1 12  1 -> 3 -> 7 -> 12
```

Figure 11: Path search procedure

To transmit data, a path with a minimum number of hops will be chosen. In practice, when there are multiple paths with same number of hops, one of them will be chosen for the primary path and the others will be reserved as substitutions. We have experimented on various topologies. The probability of finding the paths to the destinations is 100%.

5 Performance Comparison

The effectiveness of the proposed method is compared with AODV and DSDV.

5.1 Simulation Environment

Our simulations are implemented in a Network Simulator (NS-2) [NS, 2004] from Lawrence Berkeley National Laboratory (LBNL). The simulation parameters are as follows:

- Number of nodes: 40,60, 80;
- Testing area: 1000m x 1000m;
- Mobile node speed: 0m/s;
- Mobility model: random way point model (when the node reaches its destination, it pauses for several seconds, e.g., 1s, then randomly chooses another destination point within the field, with a randomly selected constant velocity);
- Traffic load: UDP, CBR traffic generator;
- Radio transmission range: 250 m; and
- MAC layer: IEEE 802.11.

Each simulation is run for 500 seconds. Each case has 10 different topologies.

5.2 Simulation Results

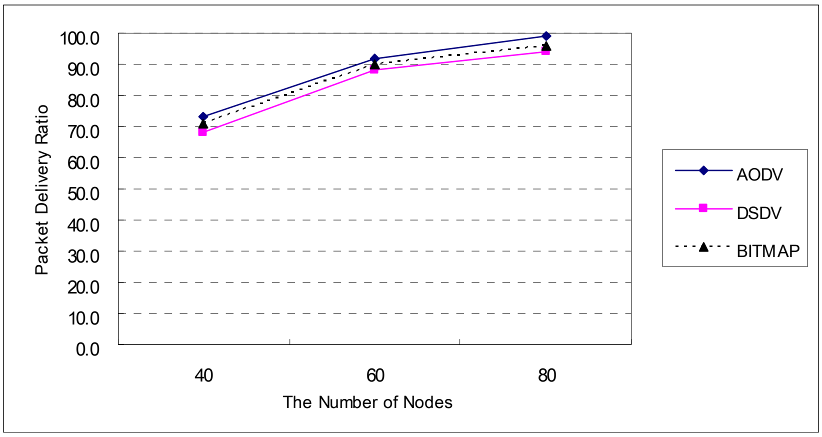

Figure 12: Packet Delivery Ratio (no mobility, 3 connections)

Figure 12 shows the packet delivery ratio of AODV, DSDV and BITMAP. The differences between these protocols are not large.

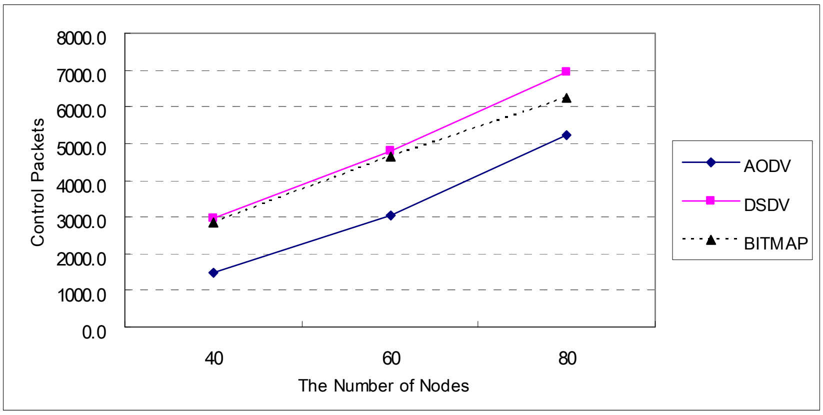

Figure 13: Control Packet Overhead (no mobility, 3 connections)

Figure 13 shows the control packet overhead required for transferring the routing information packets. AODV has less control packet overhead when the number of nodes is few. However, with respect to an increasing number of nodes, there is a decrease in the difference between the control packets of BITMAP and those of AODV.

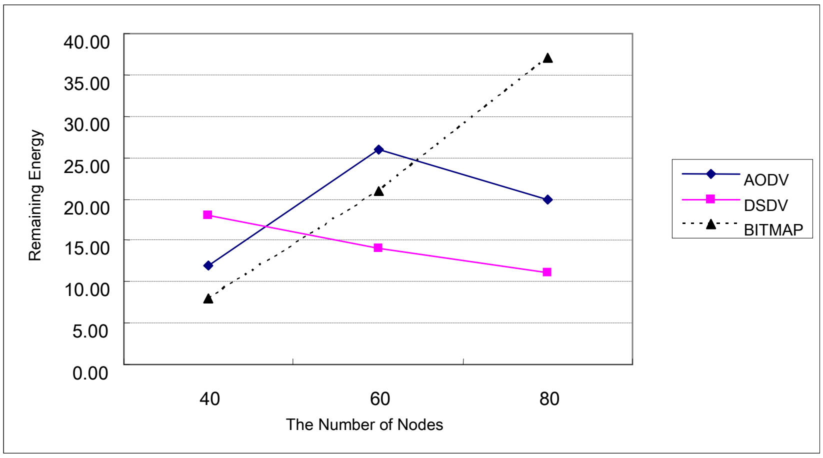

Figure 14: Average Remaining Energy (no mobility, 3 connections)

Figure 14 shows the average energy remaining for each protocol. The value of the remaining energy shows the mean value of each node at the end of simulation. Remaining energy in BITMAP is higher than that of AODV and DSDV when the number of nodes is greater.

6 Conclusion

We have proposed a knowledge discovery agent for gathering network topology information, to increase performance in ad hoc networks. The concept of the proposed agent, and an algorithm for creating knowledge about the network topology, are provided. The bit-map knowledge from collected information, that includes routing information about the network, is created and distributed by the agent. Maintaining and using the bit-map knowledge at each node, is the main issue. Computer simulation of various network topologies shows that the suggested knowledge discovery agent method of various topologies enables sources to find destinations with 100% probability. The bit-map table could be also used to search multi-paths.

Further research should aim to design a concrete, practical protocol, and compare its performance with those of the existing ad hoc network routing protocols in various environments. An effective shortest path finding method also has to be studied.

Acknowledgements

This research was supported by the Yeungnam University research grants in 2006.

References

[Chakeres et al, 02] Chakeres, I., Belding-Royer, E.: “The utility of hello messages for determining link connectivity”; The 5th International Symposium on Volume 2, Oct 2002.

[Clausen et al, 03] Clausen, T., Jacquet, P.: “Optimized link state routing protocol”; RFC3626, Experimental, Oct 2003.

[Johnson et al, 03] Johnson, D., Maltz, D., Hu, Y-C.: “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)”; IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 15 Apr. 2003.

[Kim et al, 06] Kim, C., Talipov, E., Ahn, B.: “A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks”; EUC Workshops 2006, LNCS 4097.

[Lee et al, 07] Lee, S., Muhammad, R.M., Kim, C.: “A Leader Election Algorithm within candidates on Ad Hoc Mobile Networks”; International Conference on Embedded Software and Systems (ICESS-07), LNCS4523, 2007.

[Lu et al, 03] Lu, Y., Weichao, W., Zhong, Y., Bhargava, B.: “Study of Distance Vector Routing Protocols for Mobile ad hoc networks”; Proceedings of the First IEEE International Conference 2003.

[Nasipuri et al, 01] Nasipuri, A., Castaneda, R., Dasl, S.: "Performance of Multipath Routing for On-demand Protocols in Mobile Ad hoc Networks"; Kluwer Academic Publishers Mobile Networks and Applications, 2001.

[NS, 2004] NS, The UCB/LBNL/VINT Network Simulator (NS), http://www.isi.edu/nsnam/ns/, 2004.

[Ogier et al, 03] Ogier, R.G., Lewis M.G., Templin, F.L.: “Topology Broadcast Based on Reverse-Path Forwarding”; draft-ietf-manet-tbrpf-08.txt, Apr 2003.

[Perkins et al, 94] Perkins, C., Bhagwat, P.: “Highly Dynamic Destination-Sequenced Distance-Vector Routing(DSDV) for Mobile Computers”; Proceedings of ACM SIGMOMM 1994.

[Perkins et al, 03] Perkins, C.: Nokia Research Center, E. Belding-Royer,: “Ad hoc On-Demand Distance Vector (AODV) Routing”; RFC 3561, 2003.

[Vasudevan et al, 04] Vasudevan, S., Kurose, J, Towsley, D,: “Design and Analysis of a Agent Election Algorithm for Mobile Ad Hoc Networks”, UMass Computer Science Technical Report 03-20, 2003.

[Vasudevan et al, 04] Vasudevan, S., Kurose, J, Towsley, D.: “Design and Analysis of a Leader Election Algorithm for Mobile Ad Hoc Networks”; Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP), 2004.