IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

Performance Analysis of AODV & MAODV Protocol in Mobile ADHOC Networks

Gurjeet Singh[1*], Vijay Dhir[2]

[1,2]Dept of CSE, Sant Baba Bhag Singh University, Jalandhar

[*]*Corresponding Author: hi_gtech@rediffmail.com*

Available online at: www.ijcseonline.org

Accepted: 22/Dec/2018, Published: 31/Dec/2018

Abstract- A Mobile Adhoc Network made up of mobile nodes which are wireless. Mobile Adhoc Network is self organized and self configurable. In MANET mobile nodes move randomly. Like a router, the mobile nodes in MANET can forward and receive packets. Routing is a critical issue in MANET. A recent trend in adhoc network routing is the reactive on-demand philosophy where routes are established only when they are required. Most of the protocols in on -demand category are not associating with proper security features. The adhoc environment can be accessed by both legitimate network users and attackers. It has been monitored that different protocols need different security strategies. This paper will discuss the performance analysis of AODV & MAODV protocol in Mobile Adhoc Network.

Keywords- AODV, MAODV, Mobile Adhoc Networks

I. INTRODUCTION

An Ad hoc network [1] is a collection of mobile nodes, which forms a temporary network without the aid of centralized administration or standard support devices regularly available as conventional networks. These nodes generally have a limited transmission range and, so, each node seeks the assistance of its neighboring nodes in forwarding packets and hence the nodes in an Ad hoc network can act as both routers and hosts. Thus a node may forward packets between other nodes as well as run user applications. By nature these types of networks are suitable for situations where either no fixed infrastructure exists or deploying network is not possible. Ad hoc mobile networks have found many applications in various fields like military, emergency, conferencing and sensor networks. Each of these application areas has their specific requirements for routing protocols.

Since the network nodes are mobile, an Ad hoc network will typically have a dynamic topology, which will have profound effects on network characteristics. Network nodes will often be battery powered, which limits the capacity of CPU, memory, and bandwidth. This will require network functions that are resource effective. Furthermore, the wireless (radio) media will also affect the behavior of the network due to fluctuating link bandwidths resulting from relatively high error rates. These unique desirable features pose several new challenges in the design of wireless Ad hoc networking protocols. Network functions such as routing, address allocation, authentication and authorization must be designed to cope with a dynamic and volatile network topology. In order to establish routes between nodes, which are farther than a single hop, specially configured routing protocols are engaged.

II. PROTOCOL

The Multicast AODV is developed to be used in networks that contain a number of mobile nodes that move around and therefore create situations, where the network topology changes continuously[2]. Multicast AODV is based on bi-directional shared trees that are created and terminated as the multicast receivers join and leave the multicast groups. MAODV protocol is specified in [5].

2.1 Data forwarding

For each multicast group, a bi-directional tree is created. The tree contains members of two distinct classes. Member can be either a node that has joined the multicast tree or a node that is has not joined the multicast group but is forwarding the multicast messages towards other nodes in the tree. These intermediate nodes are still members of the tree and all multicast packets pass through them. Therefore they may suffer from extra load but this is inevitable in ad hoc networking.

2.2 Message types

MAODV uses four different message types for creation of the multicast routing table. These messages are;

- Route request (RREQ)
- Route reply (RREP)
- Multicast activation (MACT)
- Group hello (GRPH)

Of these messages, RREQ and RREP are also used in the unicast operation of AODV. The others are used only for MAODV.

All the MAODV messages use IP/UDP as their carrier protocols. Port number 654 [6] is reserved for this purpose. The distribution of these control messages in the ad hoc network is limited by IP TTL field which is set per message.

2.3 Control tables

AODV keeps a routing table for unicast routes. Similarly MADV has a routing table for the multicast routes. The entries in this table have the following attributes;

- Multicast group IP address
- Multicast group leader IP address
- Multicast group sequence number
- Next hop(s)
- Hop count to next multicast group member
- Hop count to multicast group leader

Each next hop entry has the following fields;

- Next hop IP address
- Next hop interface
- Link direction
- Activated flag

In addition, a node may also keep a multicast group leader table, which is used to optimize the routing. This has the following fields;

- Multicast group IP address
- Group leader IP address

2.4 Creation of the multicast tree

Normally the first node that wants to join the multicast group, selects itself as the multicast group leader. The sole purpose of this node is that it keeps count of the sequence number that is tied to the multicast group address[3]. The basis for the formation of the multicast tree is illustrated in figure 1.

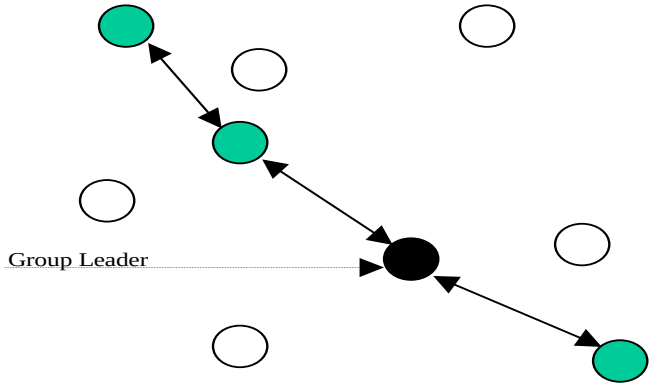

Figure 1: Group Leader in multicast tree

The group leader handles the sequence number by sending periodic Group Hello messages. These are broadcasted through the network. They carry the multicast group, group leader IP address and group sequence numbers. Group Hellos are used for disseminating group information and repairing possibly partitioned multicast trees. When a node wishes to join the multicast group or it wants to send packets to the qroup, it needs to find the route to the group. This is done using two messages; RREQ and RREP in a so-called discovery cycle. The usage of these is explained thoroughly in [6] and therefore I will describe their purpose only briefly.

2.4.1 Route requests

RREQ is used to discover a route towards a multicast (or unicast) destination. The important fields for multicasting of the message are set as follows;

- Source address; the address of the sourcing node.
- Destination address; the address of the multicast group that is the target of the discovery.
- Join-flag; if this is set, then the node originating RREQ wants to join the multicast tree. If it is unset, then the originator is a source of multicast transmission.
- Group Leader Extension; if the originator of RREQ knows the group leader (it has heard Group Hello messages for this multicast group), then the RREQ can be send towards the group leader with this extension. This helps in joining the tree since it is propable that the tree is found from the direction where the leader is.
- Sequence number; the last sequence number known to this multicast group.
- Hop count; set to zero.

When the node sends this message, it initiates a RREP_WAIT_TIMER which has no default value as of writing this but which should be at least latency of a single hop times the diameter of the network times two. If the node does not get an answer, then it retries twice by default. If there is still no answer, then the node selects itself as the group leader if it wants to join the tree[4]. However, if it only wants to send data to the tree and it cannot find the tree, then it silently discards this traffic. RREQs are sent as broadcasts throughout the network. To prevent broadcast stroms, the AODV uses a technique called expanding ring search, where the RREQ is first send with a limited TTL and then the TTL is incremented in subsequent RREQs to reach also nodes further away. Figure 2 illustrates the how initial RREQs are sent.

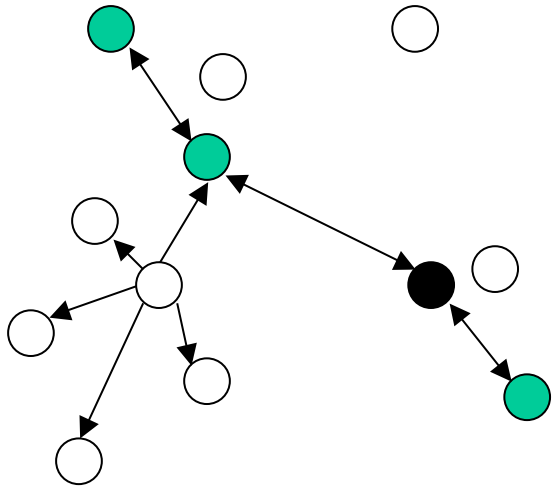

Figure 2: Initial RREQs

2.4.2 Route replies

When a node receives a RREQ for a multicast route, it first checks the Join-flag in the message. If the Join-flag is set, then the node may answer only if it is itself a member of the multicast tree and its sequence number for this tree is at least as great as the number in the RREQ. If the Join-flag is not set, then the node may answer, if it has an unexpired route to the multicast tree and its sequence number is at least as great as the number in the RREQ[5]. If neither of the above is true, then the node must find the route towards the multicast tree itself. This means that it must rebroadcast the RREQ towards the neighbors of itself. In this case, it modifies that RREQ as follows;

- The source IP address of the RREQ is the one of the node rebroadcasting it.
- The hop count is incremented by one.
- The original TTL is decremented by one.

In addition to this rebroadcast, a node does two things;

1) It creates a reverse unicast route for the node which originally send the RREQ.
2) It creates a multicast table entry for the multicast group in question.

The RREPs are send as a unicast message towards the originator of the RREQ message. This is done using the information that was learned when the RREQ was rebroadcasted and a unicast reverse route was created. Intermediate nodes increment the hop count of the message. The contents of the RREP messages are as follows;

- Hop count; set to zero if the sending node is a member of the multicast tree, otherwise set to the value which is the sending node´s distance towards the multicast tree.
- Source address; the address of the node that originated the RREQ.
- Destination address; the multicast group address
- Destination sequence number; the responding node´s knowledge of the sequence number.

Figure 3 illustrates the path of the RREP message.

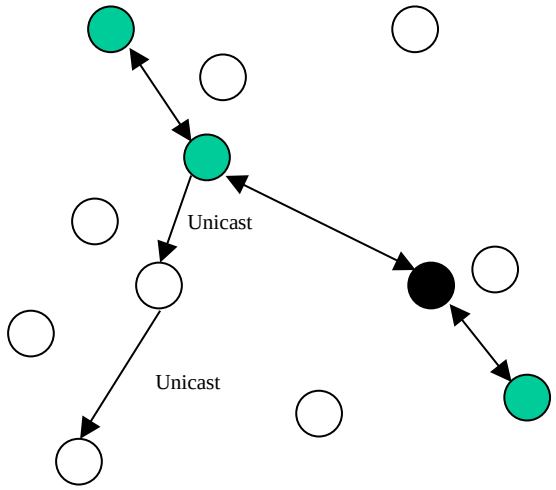

Figure 3: Unicast RREP

2.4.3 Multicast activation

A single node may get multiple replies to the RREQ message. It must choose the best out of these to be used for the multicast tree creation. The reason for this is that the multicast messages are broadcasted in layer two (in radio networks like IEEE 802.11) and therefore loops may occur if there is no control of how the tree is formed[6]. For this reason, the next hop node that is selected by the node wishing to join the multicast tree is informed about this fact by sending a MACT message. The receiver of the MACT message updates its multicast routing table by setting the source of the message as a downstream next hop neighbor. The MACT message has four flags that can be set. These are join, prune, grpldr and update. The join is used, if the node wishes to join the tree (the normal reason for MACT message) and prune is for leaving the tree. The two other messages are used, if the tree breaks and must be repaired.

2.4.4 Leaving the tree

The membership of the multicast group is dynamic. Each node is free to join or leave the group at any time. However, since a node may also act as an intermediate multicast tree hop, it might not be able to leave the tree, even if it does not want to receive the traffic for the group. Actually the fact is that a node may only leave the tree in two cases;

1) If it is a leaf node (no downstream multicast group neighbors).
2) If it is an intermediate tree node and the last downstream node of it leaves the tree.

The leaving of the tree is done by sending the MACT message with the prune-flag set.

2.4.5 Tree partitions

Even if the AODV and MAODV protocols may be used also in fixed networks, it is most likely that the implementations are seen in ad hoc networking. Since ad hoc networks are highly dynamic by nature, this means that also the multicast tree is higly dynamic [7].

The changes in the network topology may lead to two different situations;

1) A link is broken
2) Multicast tree is partitioned

Lets look at each of these cases separately.

A node discovers a link breakage either actively or passively. Active discover means that the MAC layer informs upper layers about reachability problems. Passive discovery happens, if the node has not heard from it´s neighbor for a while. In this case, it might try to ping the neighbor or ask a route towards it via RREQ[8].

Be it either case, when the node discovers connectivity loss with the multicast tree neighbor, then if it is the downstream neighbor, it is responsible for correcting the situation.

What the node does is that it sends a RREQ with a Multicast Group Leader Extension. This extension contains the old distance of the node to the group leader. Only multicast tree member nodes that have distance to the group leader equal or less than the one set in the extension may answer with RREP. This prevents the nodes on the same side of the break as the initiator of the RREQ from answering and thus creating possible loops.

If the repair leads to a situation, where the node´s new distance to the group leader is greater than the old one, then it must inform its downstream nodes about this. This is done with MACT message where the update-flag is set. This MACT message is multicasted to all of the tree members, also upstream. But upstream members see that this message came from a downstream node and therefore discards the message. This local tree repair is illustrated in figures 4 and 5.

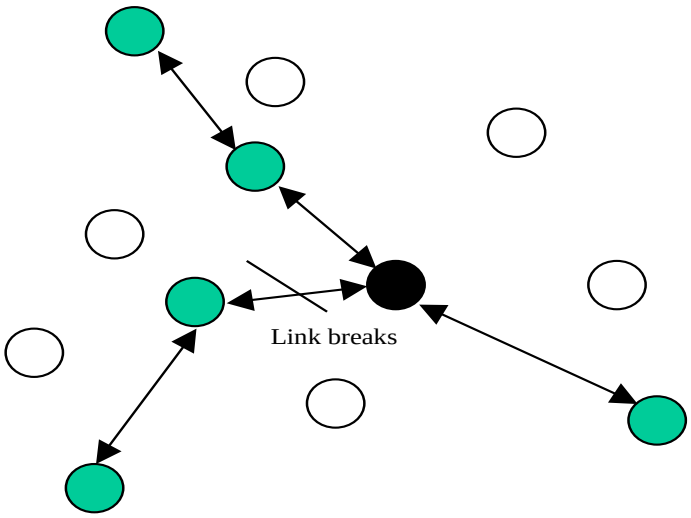

Figure 4: A link breaks

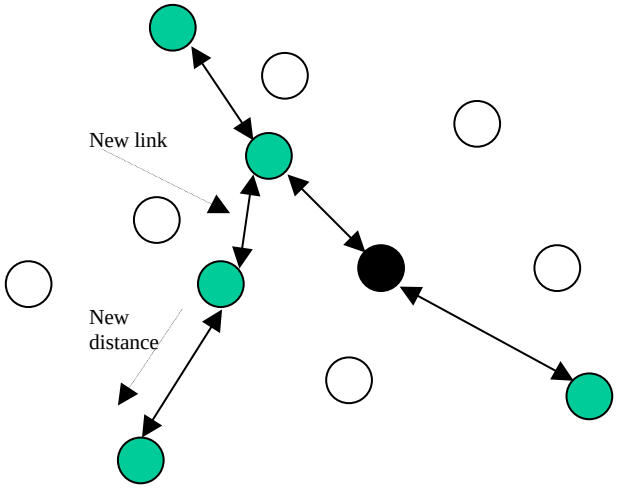

Figure 5: New link formation

The other case is when the whole tree becomes partitioned. This is illustrated in figure 6.

When the node tries to reconnect the disconnected link and does not get an answer to the RREQ message number_of_retries times, then it must assume that the tree is partitioned. If this is the case and it is a member of the group, then it becomes a new group leader. It broadcasts group hello message with update-flag set indicating that there is a new group leader. However, if the node has multiple downstream nodes, then it selects any one of these and sends a MACT message with grpldr-flag set. This indicates that the receiving node should become group leader. If it is a group member, it becomes a leader, otherwise it continues seeking the leader with the previously described methods. When the group leader is finally found, it broadcasts group hello message with update-flag set to indicate that changes has occurred in the network.

If the node trying to repair the break is not a multicast group member, then it must try to find a new group leader from the downstream nodes it has. If there is only one downstream node, then the node prunes itself from the tree and everything begins from the beginning on the next hop downstream node.

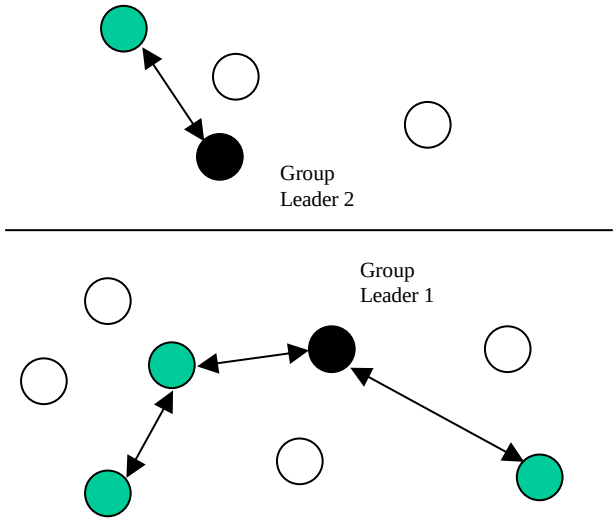

Figure 6: Partitioned network

2.4.6 Merging partitions

When the two network partitions become united once again, there is two multicast group members for a single multicast group. Since this is an illegal situation, it must be corrected. What happens is that the group leader that has numerically lower IP address joins the tree of the other group leader. It does this by sending a RREQ with repair-flag set. This RREQ is unicasted to the other group leader and all the nodes along the way must update their multicast routing tables so that also they begin to use the group leader that has numerically higher IP address. The rest of the tree that was formed with the originator of RREQ is given knowledge of the group leader change by issuing a group hello with update-flag set.

2.5 Simple protocol for Multicast and Broadcast

This protocol is a side development of the DSR unicast routing protocol. The key idea is very simple; in DSR that nodes do not keep any state information such as routing tables for other nodes than the ones that they are communicating with. This simplifies the protocol operation. Multicast (and broadcast) packets are encapsulated into Route Request packets that are send everywhere. The nodes that want to receive the multicast transmission, copy the packet to the application level before re-broadcasting it. Essentially the approach that this protocol is using, leads to very inefficient usage of bandwidth. Especially if the multicast group members are distributed very sparsely in the network. However, in small networks (where this is intended) this kind of flooding approach may very well succeed[9].

2.6 ADMR

ADMR is developed mainly by the same group as the one described in the previous section. It solves the other side of the coin – how multicasting is done on a large ad hoc network. On contrary to the simple protocol for multicast and broadcast, ADMR is build on state information. The state is created for each (S(ource), G(roup)) pair and therefore all the resulting multicast trees are source-based. This differs from the approach of MAODV, where the tree is created based on the group leader. Otherwise ADMR shares the same basic concepts which are included also in MAODV. Namely these include;

- Ability to dynamically create multicast trees even if the nodes change their position and without relaying on external position information source such as GPS.
- Ability to repair locally broken links and recover from network partition.

The protocol documentation seems to be much more complete and ready than the one for MAODV. This means that it is easier to implement and test.

2.7 SRMP

SRMP is another protocol that builds on top of the DSR unicast routing protocol. The concept is totally different than the ideas behind the other multicast routing protocols. Instead of a source- or core based tree, SRMP creates a mesh network for multicast traffic forwarding. According to the SRMP draft, this makes forwarding more reliable and less dependend of the movements of the nodes. As is the case for ADMR, SRMP seems to have better documentation and therefore it is easier to test and implement.

III. PERFORMANCE ANALYSIS

Since there are multiple possibilities for Ad Hoc routing, it is good to describe how the suitability and performance of a routing protocol might be evaluated. This is done in RFC 2501 [10], which states that the criteria might include;

- Distributed operation
- Loop-freedom
- Demand-based operation
- Proactive operation
- Security
- “Sleep” period operation
- Unidirectional link support

Also, any single protocol performance might be measured with different criteria. This may include the following viewpoints;

- End-to-end data throughput
- Route acquisition time
- Percentage out-of-order delivery
- Efficiency

In addition to see the protocol internal efficiency, the following characteristics might be examined;

- Ratio of average number of bits transmitted/delivered
- Ratio of control bits transmitted/data bits delivered
- Ratio of control and data packets transmitted/data packets delivered

When testing a protocol, the following parameters should be altered;

- Network size ie. the number of nodes and the possible movement area
- Network connectivity ie. the average number of neighbors for a node
- Topological rate of change ie. how fast the topology changes
- Link capacity
- Fraction of unidirectional links
- Traffic patterns
- Amount of sleeping nodes

For AODV and MAODV, several simulations have been made. Although the AODV results can not be directly used for MAODV evaluation, they give hints of how the protocol behaves generally.

3.1 AODV Simulations

The basic AODV protocol has been evaluated against multiple other unicast routing protocols. These are Adaptive Distance Vector Routing, Dynamic Source Routing, Destination Sequenced Disctance Vector Routing and Temporally Ordered Routing Algorithm.

Whilst the constants for the simulations vary, the following results can be received from these comparisons;

- From the four protocols, AODV and DSR are the best in higly dynamic network topology, where the nodes are moving a lot. This result does not change within the tested number of sources (10,20 and 30).
- The same protocols create the biggest routing overhead in highly dynamic networks. Also this result is not affected by the number of source nodes.
- Both AODV and DSR suffer from huge delays (> 1 second) when the number of source nodes increases over 30 in high mobility environments.

Immediate result of these comparisons is that when the number of nodes increase to more than a handful and especially if these nodes are highly mobile, the routing protocols have a lot to do to keep up with the current network topology. This influences the packet delivery ratio and distrubs particularly TCP-based connections. Therefore one could argue that the current unicast routing protocols are not ready for large-scale implementations.

Simulation studies show that Kleinrock´s and Silvester´s paper; "Optimum Transmission Radii for Packet Radio Networks" still apply also for the current Ad Hoc networks. Essentially these papers state that optimum level of delivered packets can be obtained with 6-8 neighboring nodes. Whilst the delivery ratio may increase a small percentage with more neighbors, the contention of the usable transmission space is limiting the overall throughput.

3.2 MAODV Simulations

There are not too many simulations for multicast operation of AODV. Especially, no comparisons with other multicast ad hoc routing protocols have been done. Therefore the usable results apply only to the MAODV protocol itself. From the studies, the following issues can be found;

- Throughput ratio decreases if the area that has to be covered with a fixed amount of MAODV nodes increase.
- A change in the speed of mobile nodes does not have a big effect on the throughput ratio.
- Control overhead in not highly dependent of the mobile node movement.

Based on these simulations one can argue that MAODV can be used as a multicast routing protocol in ad hoc networks. However the same limitations as for the unicast AODV apply also on the MAODV side since the protocol operation is the same.

IV. CONCLUSION

Ad hoc networking is a technique that will become more and more important over the next couple of years. The reason for this is the fact that mobile nodes performance improves as does the capacity of wireless networks. Natural way of evolution is to give up regulatory, operator-based wireless networks and step into the era of wireless Internet – the same original idea that was with the wired Internet. This means a network that is controlled and owned by nobody but that can be used by anybody. In this paper, I have detailed the internals of a multicast routing protocol for ad hoc networking. The MAODV is a protocol that is an extension to the unicast AODV routing protocol, which itself is a development of DSDV unicast routing protocol. A review of other multicast routing protocols was also given. Since there is a lot of research activity in mobile ad hoc networking, there are also several different multicast routing protocols. None of these protocols, including the topic of this paper, is ready for a large scale implementation due to the lack of simulation studies and real life testing. However, as is stated in the implementation study of AODV routing protocol, no simulation can replace the real implementation and testing with real equipment in real life situations. Normally these do tend to improve the quality of protocols and improve their simplicity and efficiency. Also, all routing protocols in ad hoc networks suffer from the same phenomenon that is missing from fixed networks; when the mobility of the nodes increase, there is either a lot of routing updates (for topology-based protocols) or route requests (for on-demand protocols). This reduces the amount of effective bandwidth used for the actual data forwarding. And the worst thing is that the more there is control traffic in the network, the more the nodes have to process it. This can severely impact the processing power of the nodes.

REFERENCES

[1] Malkin, G.: RIP version 2, STD 56, November 1998, ftp://ftp.isi.edu/in-notes/std/std56.txt

[2] Moy, J.: OSPF version 2, STD 54, April 1998, ftp://ftp.isi.edu/in-notes/std/std54.txt

[3] Adams, A., Siadak, W. : Protocol Independent Multicast – Dense Mode (PIM-DM) Protocol Specification (Revised), Internet-draft, November 2001, http://www.ietf.org/internet-drafts/draft-ietf-pim-dm-new-v2-00.txt

[4] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I. : Protocol Independent Multicast – Sparse Mode (PIM-SM) Protocol Specification (Revised), Internet-draft, November 2001, http://www.ietf.org/internet-drafts/draft-ietf-pim-sm-v2-new-04.txt

[5] Royer, E., Perkins, C.: Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing, Internet-draft, July 2000, draft-ietf-manet-maodv-00.txt

[6] Port numbers, March 2002, http://www.iana.org/assignments/port-numbers

[7] C. Perkins, E. Royer, and S. Das. Ad hoc on demand distance vector (AODV), Internet-draft, March 2000, http://search.ietf.org/internet-drafts/draft-ietf-manet-aodv-10.txt

[8] Jetcheva, G., et.al., A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks, July 2001, http://www.ietf.org/internet-drafts/draft-ietf-manet-simple-mbcast-01.txt

[9] Jetcheva, G., et.al., The Adaptive Demand-Driven Multicast Routing Protocol for Mobile Ad Hoc Networks (ADMR), July 2001, http://search.ietf.org/internet-drafts/draft-jetcheva-manet-admr-00.txt

[10] Labiod, H., Moustafa, H.: The Source Routing-based Multicast Protocol for Mobile Ad Hoc Networks (SRMP), November 2001, http://search.ietf.org/internet-drafts/draft-labiod-manet-srmp-00.txt

