Academic Open Internet Journal www.acadjournal.com Volume 16, 2005

SCALABLE MULTICASTING
ADAPTIVE CORE BASED APPROACH

Baskaran Kaliaperumal
Senior Lecturer, EEE
A. Ebenezer Jeyakumar
Principal
Government College of Technology, Coimbatore

Abstract - Multicast is efficient way to distribute information from single source to multiple destination or many-to-many in communication networks. Mobile ad-hoc network needs special multicast routing protocol to adapt its characteristics including local broadcast capacity, arbitrary topology change, and bandwidth constraint and power limitation. A multicast routing protocol for MANET should find compromise between routing overhead and data transmission efficiency so that it can efficiently use bandwidth and power. For this aim, this paper proposes a new multicast routing protocol called Adaptive Core based Multicast Routing protocol (ACMP) which constructs and maintains a group-shared tree using adaptively selected core only when group traffic exists. ACMP attempts to react more quickly to broken tree edge by detecting link failures during data forwarding. Once a link failure is detected, this protocol uses local route recovery to establish a temporary route and periodical tree refreshing to maintain an optimal multicast tree. The performance of ACMP is evaluated via simulation.

Keywords: MANET, multicast routing, Core, Mtree, RAR, RAA

1. Introduction

A Mobile Ad-hoc NETwork (MANET) is a collection of wireless mobile nodes forming a dynamical temporary network without the use of any existing network infrastructure or centralized administration. Different from traditional wireless networks, nodes have to use other mobile nodes in the network instead of some fixed router or an infrastructure as relay if the destination is not in their coverage area. The nature of mobile nodes decides the features of MANET such as broadcast capacity, dynamic topology, bandwidth and power constraints, etc. This type of networks is suitable for supporting the applications like virtual classroom, emergency search and rescue-operation, data exchange during conference, meeting, etc. Eventually, MANETs need also multicast routing protocols to establish many-to-many communication for efficient data delivery among a group[1]. The properties of MANET make multicast routing protocol for wired network (ex. DVMRP [2], MOSPF [3] and PIM [4]) not suitable for MANET because at the same time they cannot face group membership dynamics and frequent topology changes with limited bandwidth.

Lots of multicast routing protocols have been proposed for MANETs such as AM Route [5], AMRIS[6] CAMP [7], LAM [8], MAODV [9], MZR [10], NSMP [11] and ODMRP [12]. These protocols can be classified to *tree-based (*[5], [6], [8], [9] and [10]) and *mesh-based* ([7], [11] and [12]) according to routing structure. The *tree-based* approach consists of creating and maintaining a multicast tree to deliver data. Once a tree is established, packets or messages are sent to all the routers in the tree only once. For a tree containing N nodes, N-1 links are needed to forward multicast packets with point-to- point links. In the case that the network has broadcast links using a single channel, only internal tree nodes forward multicast packets. With the help of broadcast capability of MANET, the latter approach proposes to use *mesh* structure a connected graph for multicast traffic forwarding. Multicast packets are broadcast to the neighborhood of nodes. Only the neighbors that are also mesh members will react to non-duplicated multicast packets. Mesh structure offers redundant routes for data delivery gives rich connectivity but it involves more routers for forwarding than tree structure. Therefore, tree-based approach is more efficient in data transmission than mesh when traffic is long-term. Here, data transmission efficiency means the number of data packets transmitted in the network for delivering a data packet. Tree is sensitive to node mobility since it provides minimal connectivity among multicast group members. Routing messages are needed to repair tree branches every time topology changes touch them. On the contrary, mesh offers more connectivity. Hence, it is robust against topology changes. Mesh can tolerate link failure between two mesh members if they can get data from other mesh members. Hole will disappear in the next period of mesh refreshing. So, in terms of routing overhead, mesh-based approach is more efficient than tree-based approach in high mobility network. Multicast routing protocols can also be categorized into *source-oriented* and *group-shared* according to construction mechanism. Group-shared mechanism gives one structure per group and all sources use this structure to distribute their data. On the other hand, source-oriented mechanism can construct a structure for each source according to various criteria (ex: the shortest path, various QoS requirements). Therefore, it is more efficient in terms of data transmission than group-shared mechanism when a group has multiple sources. However, this mechanism can easily suffer from scalability problem as the number of groups and/or sources per group increase. Moreover, source-oriented mechanism should send much more routing message per group than group-shared mechanism since it should maintain all structures of the group. These issues make group-shared mechanism more suitable for the MANET environment. This paper proposes a multicast routing protocol, Multicast Protocol with

Adaptive Core (ACMP) for MANETs, which constructs a group-shared multicast tree, called Mtree, for a multicast session on demand. In the rest of this paper, Section 2 presents the main design principles of the ACMP. Section 3 describes in details the creation and maintenance of multicast tree. Section 4 evaluates the performance of ACMP in different traffic loads and mobility. Section 5 provides our conclusion.

2. ACMP

The aim of Multicast Routing protocol with Adaptive Core (ACMP) is trying to find tradeoff between routing overhead and data transmission efficiency in MANET. ACMP uses a tree structure to connect all group members on demand. ACMP uses core to limit the control traffic needed for group members to join multicast group. But it differs from other group-shared tree-based multicast routing protocols, which also use the concept of *core* ([6], [7] and [8]). In our protocol, *core* is the first source of a multicast session. This choice guarantees that the node that acts as core is a multicast group member and is interested in transmitting multicast traffic. If no core exists in the network, it is not necessary to construct and maintain tree and all receivers would remain silent. This property can be named as “on demand” compared with other group-shared tree-based multicast routing protocols. In those protocols, tree should be maintained even when no multicast traffic is present. Another advantage of this choice is that ACMP reduces to source-oriented in the case of a single source, and is group-shared in the case of multiple sources.

All tree-based multicast protocols use *timeout-based path monitoring* to detect link failure. A link failure is detected when a node has not heard its tree neighbor during last period. If a link failure is detected, downstream (the direction from root to leaf nodes) node rejoins tree to maintain tree connection. This mechanism reacts slowly to link failure according to the time period value. Consequently, data transmission is blocked until the branch is repaired. Another disadvantage is that this mechanism needs to send extra routing message to maintain tree when there is no traffic in the group. ACMP proposes to use *reactive path monitoring* in which a tree member supervises links when it forwards data to other next tree members. Hence, it can immediately discover a broken branch 2 and react to this failure. The shortcoming of this maintenance is that it increases distance between tree members from one hop to multiple hops and cannot recover tree structure in some cases. This problem can be solved by periodical tree refreshing mechanism in which core periodically broadcasts a message to the whole network to reconfigure the multicast tree. Therefore, routing overhead caused by tree maintenance (topology change) is reduced with on-demand local route recovery. On the other hand, data delivery efficiency is improved by periodical tree refreshing.

3. Protocol description

The control part of ACMP consists of two aspects: *Tree construction* and *Tree maintenanc*e. *Tree construction* is the aspect by which a core is selected and advertised to the network. Nodes that are interested in the multicast session join the tree. *Tree maintenance* is the aspect where tree members detect broken branches and repair the failure to continue multicast traffic delivery in Multicast tree (Mtree). Mtree maintenance also takes care of departures of group members.

i. Tree Development

Each node in MANET possesses a multicast routing table, called MRTable, which stores multicast routing information. The existence of an entry in the table, which corresponds to a multicast session tree, means that traffic is present in that group. A source can know whether it should act as core of this multicast session or just to participate the multicast tree as a normal source by checking if there is an entry of that group. Similarly, a receiver can decide if it should run the procedure to join the tree or keep being silent by consulting multicast routing table. A multicast routing entry has two states: active and inactive. Nodes that are tree member have an active entry and other nodes have inactive one. Tree construction is based on the following mechanism: a core broadcasts a *Core Advertisement* (CA) message to the network to announce a multicast session. The reverse path to core is established as the CA propagates in the network. A node that is interested in the multicast group runs RAR/RAA procedure to join Mtree upon receiving the CA. In this procedure, the node sends a *Route Active Request* (RAR) message towards core and waits for a *Route Active Acknowledge* (RAA). A CA message contains a multicast group address, a core address and a sequence number. The sequence number, together with the group address and the core address, uniquely identifies each CA. Upon receiving a non-duplicated CA broadcast by a core, all the other nodes create an inactive route entry in their MRTable and store the group address, the core address and the sequence number. A node discards all duplicated CA messages. Then they record next hop information for the core to construct a reverse path. This path may be used to send or forward RAR messages towards core. If a node is a member of this multicast group, it initiates a RAR/RAA procedure. At last, the node updates the next hop information to its address and rebroadcasts the packet to its neighbors. Consequently, after all nodes in the network receive the CA, they know the creation of a multicast session and own a route to the core. A RAR/RAA procedure is initiated by a group member when it wants to take part in the Mtree. This group member generates a RAR packet, sends this packet towards the core and at the same time, it becomes a potential Mtree member. The content of RAR that gives the information about potential downstream Mtree member is updated each time when this RAR passes a node. A RAR message is sent to the next hop for the core. Upon receiving the RAR, the node becomes potential Mtree member. This node stores the potential downstream member as downstream node and updates the content of the CA to its node address then it continues to relay this RAR to the core. A potential Mtree member does not send or relay other RARs when it is waiting for RAA. But it still records the content of these RARs. So that, as a RAR propagates in the network towards core, a potential Mtree branch is formed. This type of branch begins from a potential Mtree leaf that is a group member and ended by meeting an Mtree member or a potential Mtree member. The first Mtree member that receives the RAR breaks off the forwarding of the RAR and replies a RAA to activate route entries of nodes on the potential branch(es). When a potential Mtree member receives a RAA, it sets the route entry to be active and sends RAA to all its direct potential downstream Mtree members. A RAR/RAA procedure is finished when RAA arrives at the potential leaf of the branch. After a timeout if a potential Mtree member does not receive a RAA message, it backs to normal state and informs its all potential downstream members about this failure. A group member that meets this type of failure would wait until next CA to join Mtree.

ii. maintenance of Tree

The tree maintenance aspect in ACMP contains four parts: *local route recovery*, *periodical Mtree refreshing*, *group members' departures* and *core concurrence*. Local route recovery reacts quickly to link failure and demands a few of routing overhead. On the other hand, periodical Mtree refreshing overcomes the link failures that cannot be solved by local route recovery and gives an optimal tree structure.

During data transmission, an Mtree member finds a branch broken when it cannot forward data packet through a tree edge. It runs local route recovery to try to solve this problem. When a broken branch is found, the upstream Mtree member of the broken branch sends a *Joining Invitation* (JI) message to invite the downstream node to rejoin the Mtree. JI message is broadcast in n-hops away using an *Expanded Ring Search* (ERS) to control message propagation. Initially, n is set to 2. After a timeout if the node does not hear from the addressed node, it increases n and rebroadcast JI message. The node repeats this procedure until either receive RAR message from the addressed node or n reach a maximum value, called Greatest-Range. Like CA, JI message also prepares a path from sender to the addressed node. Upon receiving JI, the downstream node takes part in Mtree by running RAR/RAA procedure.

Core is in charge of initiating periodical Mtree refreshing. Every period (PERIOD_REF), core computes a new reference and broadcasts a CA to refresh Mtree and reverse path for the core. Once a node receives a non-duplicated CA, it updates the information of the route entry, empties the field of downstream and sets the entry to inactive. Group members should run RAR/RAA procedure to reconstruct Mtree.

Periodical Mtree refreshing gives multicast group members the possibility of leaving the group implicitly. These nodes can firstly check whether next moment of Mtree refreshing arrives soon. If so, these nodes do not run RAR/RAA procedure when they receive CA so that the branch(es) will be pruned silently. Otherwise, these nodes should explicitly leave tree by sending a message to their upstream nodes. In the case that the core wants to leave group, it checks whether there is another source in the Mtree that can become core. If it is the unique source in the group, it dismisses the tree. Or else the new core will be in charge of sending periodical CA.

Because of network partition or any other reason, a source may make a decision to become a core without hearing a CA from core. Hence, there may be more than one core existing in the network. After the network converges, a core can hear the CAs of other core(s). These cores use core competition algorithm to decide which source is the winner and it should continue acting as core. Cores can use their IP address, identification or any other information to compete. The losers will stop sending periodical CA and will not react to group member join packets so that theirs tree will be dismissed automatically after the multicast route entries time out.

iii. Data Forwarding

Once the Mtree is constructed or the branch is successfully added into Mtree, a source begins to send traffic to the multicast group. When an Mtree member receives non-duplicate traffic packet from an Mtree edge, it sends the packet to up layer if the node is a group member. The node forwards the packet to the other edge of the Mtree. It will mark a broken branch if the other end of the edge is unreachable. There is a great possibility that a data packet should be sent to more than one Mtree edge. The node duplicates the packet and sends them to corresponding directions.

4. Performance simulation

i. Simulation Environment

To analyze the performance of ACMP, we use a network simulator, *ns-2* [13]. The simulation models use 60 wireless nodes forming an ad hoc network. These nodes move in a $1000\text{m} \times 1000\text{m}$ flat space for 300 seconds simulation time. The movement model of nodes is the random waypoint model [13] without pause time. Constant Bit Rate (CBR) traffic source is used in simulation. Multicast sources generate two 512-byte data packets per second.

A number of movement scenarios and group scenarios are used as inputs to the simulations. Each movement scenario file determines movements of 50 nodes and the speeds of mobile nodes are uniformly distributed up to a maximum speed. Each data point in figures presents an average of 10 movement scenarios of same maximum speed since the performance results are sensible to node mobility and network topology. Group scenario files determine which nodes are group members and among group members which ones are sources. Group scenario files also decide when sources start sending packets. Nodes join the multicast session at the beginning of the simulation and remain as members throughout the simulation. Each group scenario files has one multicast group.

The following metrics are used to analyze the performance of ACMP:

- **Packet delivery ratio**: The number of data packets correctly delivered to receivers over the number of data packets sent by sources.
- **Number of control bytes transmitted per data byte delivered**: A ratio of routing bytes transmitted to data bytes delivered. This metric is used to investigate how efficiently routing packets are utilized in delivering data. Routing packet headers are also included in calculation.
- **Number of data and routing packets per data packet delivered:** This metric shows the efficiency of multicast routing protocol in terms of bandwidth utilization.

ii. Simulation Results

Period of Mtree refreshing

We first examine the impact of PERIOD_REF on data delivery ratio of ACMP since the performance of the protocol depends on this value. The maximum node speed is varied from 0m/s (stable) to 30m/s (high mobility). There are 10 group members and among them two are sources. In Fig. 1, we can observe that in all mobility cases, ACMP can deliver 82% of traffic packets to receivers. This is due to Mtree maintenance mechanism that can quickly detect link failures and try to solve them immediately. The probability of dropping a packet increases with mobility speed. Tree structure offers the unique route to distribute data packet of from sources to receivers. Once topology changes touch Mtree, packets transferred on the broken branch will be dropped. High mobility causes high topology changes that in turn give high drop rate. The results also show that this protocol is scalable to mobility especially with small PERIOD_REF. The most routing overhead of ACMP comes from CA since it is periodically broadcasted in the whole network. As shown in Fig. 2, the big PERIOD_REF values create smaller number of routing messages to construct and maintain Mtree. Consequently, the congestion in the network and Mtree is much lower. Therefore, the big PERIOD_REF values give a better packet delivery ratio in low mobility cases. But, in high mobility cases, Mtree should be refreshed more frequently against topology changes. So the packet delivery ratio of low PERIOD_REF values is better than that of high values but with the price of creating more routing overhead. In the rest of simulations, we use 5 seconds as PERIOD_REF since it can deliver more than 90% data packet in all case but creates less than 5% routing overhead.

Group size

This experiment is to test the performance of ACMP when group size increases. There are 2 sources per group. The number of multicast group member is varied in the set {5, 10, 15 and 20}. Fig. 3 shows the packet delivery ratio of ACMP as a function of mobility speed. The result indicates that ACMP delivers at least about 90% of data packets.

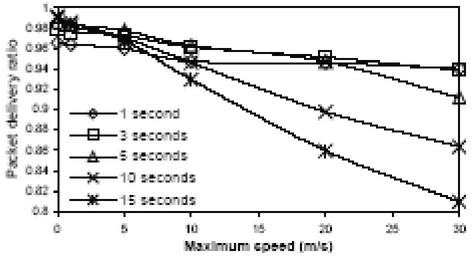

Fig.1 Packet delivery ratio with different PERIOD_REF

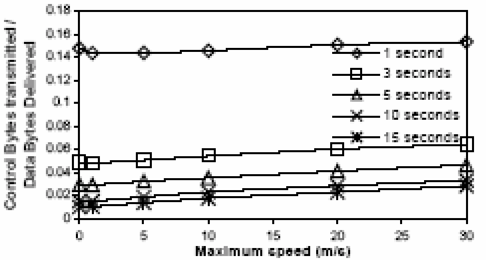

Fig. 2 Number of Control bytes transmitted per data byte delivered with different PERIOD_REF

Group size increases cause only 1 to 4 percent degradation of packet delivery ratio. So ACMP is robust to group size also in high mobility case. The average number of control bytes transmitted per data byte delivered with different group members is shown in Fig. 4. We can see that ACMP efficiently utilized control message in delivering multicast data. Only CA message is periodically broadcast in the whole network, all other messages are locally broadcast or sent in unicast way and only when they are needed. Because of the properties of tree structure, routing overhead linearly increases in the function of mobility speed. As expected, the efficiency improves when the number of multicast members grows larger. Although more RAR, RAA messages are sent when more nodes participate in a multicast group, the number of data delivered increases since more members receive the data. Fig. 5 shows the average number of total packets transmitted per data packet delivered with different group members. Similar to Fig. 4, the number increases in the function of maximum movement speed and the protocol becomes more efficient when there are more multicast members.

Number of sources

In this experiment, the performance of ACMP with different sources in multicast group is examined. The multicast group size is set constant at 10. Node mobility speed is varied from 0 to 30 m/s. The number of multicast sources ranges from 1 to 5.

Packet delivery ratio of ACMP with different sources is shown in Figure 6. Similar to Figure 3, ACMP delivers high percentage of data.

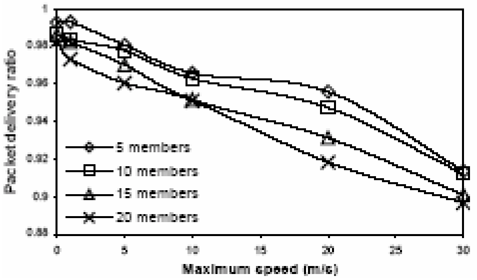

Fig. 3 Packet delivery ratio in different group size cases

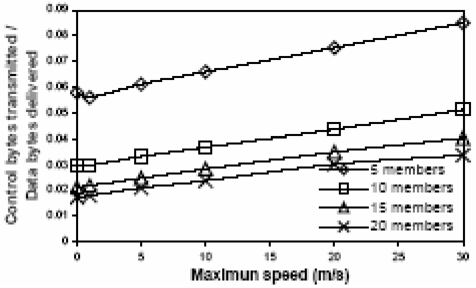

Fig. 4 Number of Control bytes transmitted per data byte delivered with different group members

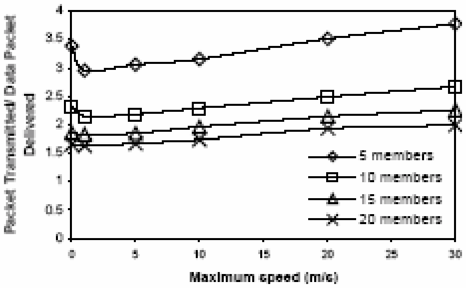

Fig. 5 Number of Total Packet Transmitted per Data Packet Delivered with diff. group members

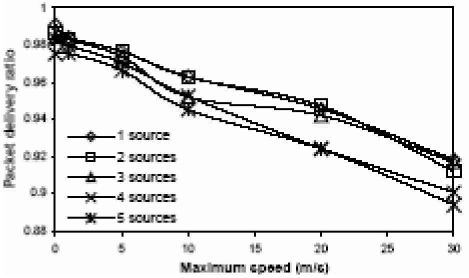

Fig. 6 Packet delivery ratio in different source cases

The performance becomes worse when number of source increase. Source number increasing means more traffic existing in multicast group that brings higher possibility of congestion especially around root. But from the result, we can find that the degradation is less than 2.5%. The average number of control bytes transmitted per data byte delivered with different group members is shown in Fig. 7.

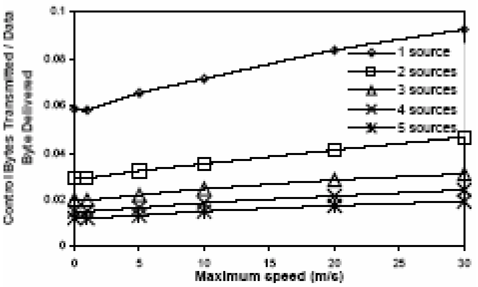

Fig. 7 Number of Control Bytes Transmitted per Data Byte Delivered

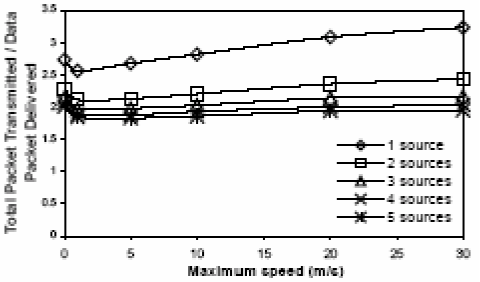

Fig 8 Number of Total Packet Transmitted per Data Packet Delivered

The value rises when node mobility increases and source numbers decreasing because of the nature of group-shared tree. Figure 8 shows the average number of total packets transmitted per data packet delivered with different source number. Compared to Figure 7, because of the same reason, these two figures have the same tendency.

V. Conclusions

A new on-demand multicast routing protocol using adaptive core for MANET has proposed in this paper. ACMP constructs on-demand one source oriented group-shared tree and is designed to provide a trade-off between data transmission efficiency and routing overhead. Adaptive core can reduce routing overhead. This protocol selects the first source of a multicast session as core to give the indication of multicast data so that a multicast structure will be constructed and maintained only when there are requirements. Core also limits the control traffic needed for group members to join multicast group. Tree structure is employed because of its efficiency in data transmission. And routing message takes a little percentage of traffic in the network, so data transmission efficiency is more important. To improve the performance of tree structure, ACMP detects link failure during data forwarding, and uses two phases, local route recovery and periodical Mtree refreshing, to maintain an optimal multicast tree.

We analyze the performance of ACMP under *ns-2* simulator with different traffic scenarios and movement scenarios. The results show that ACMP is scalable to node's mobility, group size and source number. And it is economic in bandwidth and power consumption since it sends a few packets for delivering a data packet to receivers.

Reference

[1].Dr.A.Ebenezer Jeyakumar, K.Baskaran, V.Sumathy, “Blocking Probability of Adaptive Group Multicast with time Driven Priority Scheduling,’ DAMS-2003 International Conference,CIT, Coimbatore, January-2003.

[2]. S. Deering, C. Partridge and D. Waitzman, “Distance Vector Multicast Routing Protocol”, RFC-1075, November 1988

[3]. J. Moy, “Multicast Routing Extensions for OSPF”, *Communications of the* ACM, vol. 37, no. 8, pages 61-66, 114, August 1994.

[4]. S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C. –G. Liu and L. Wei, “The PIM Architecture for Wide-Area Multicast Routing” IEEE/ACM Transactions on Networking, vol. 4, no. 2, pages 153-162, April 1996.

[5]. E. Bommaiah, M. Liu, A. McAuley, R. Talpade, “AMRoute: Ad-Hoc Multicast Routing Protocol”, Internet Draft, August 1998

[6]. C.W. Wu, Y.C. Tay, “AMRIS: a Multicast Protocol for Ad Hoc Wireless Networks”, In *Proceeding IEEE MILCOM’9*9, Atlantic City, November. 1999

[7]. J.J. Garcia-Luna-Aceves, E.L. Madruga, “ A Multicast Routing Protocol for Ad-Hoc Networks” , Proceedings of IEEE INFOCOM’99, New York, pages 784-792, March 1999

[8]. L. Ji, M.S. Corson, “ A Lightweight Adaptive Multicast Algorithm” , In *Proceeding IEEE GLOBECOM’9*8, Sydney, pages 1036-1042, November 1998

[9]. E.M. Royer, C.E. Perkins, “Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol”, In *Proceedings of ACM/IEEE MOBICOM’9*9, Seattle, pages 207-218, August 1999

[10]. V. Devarapalli, A. A. Selcuk, D. Sidhu, “MZR: A Multicast protocol for Mobile Ad Hoc networks” Internet-draft, November 2000

[11]. S. Lee and C. Kim, “Neighbor Supporting Ad hoc Multicast Routing Protocol”, In *the first Annual Workshop on Mobile ad hoc networking & Computing*, *MobiHoc 200*0, Boston, pages 37-44, August 2000.

[12]. S. Lee, W. Su and M. Gerla, “On-demand multicast routing protocol”, In *IEEE WCNC’9*9, pages 1298-1304, New Orleans, LA, September 1999.

[13]. J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu and J. Jetcheva. “A performance comparison of multi-hop wireless ad hoc network routing protocols”. *In Proceedings of the 4 th International Conference on Mobile Computing and Networking (ACM MOBICOM’98*), page 85-97, October 1998.

[14]. Kevin Fall and Kannan Varadhan, editors, “ns notes and documentation”, The VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC, February 2000, Available at http://www-mash.cs.berkeley.edu/ns/.than2.5%.