Implementation of MAODV and tree maintenance in overlay Multicasting Protocol for MANETs

P. G. V. Suresh Kumar

Associate Professor, Department of IT & SC, AAiT, Addis Ababa University
Addis Ababa, Ethiopia

Abstract---*Distributed Computing foundation and implementation, plays a vital role in current research domain. An effective, efficient and a potential foundation for developing Distributed Applications is Content-based routing (CBR). The communication model of CBR, are primarily vested on inherent addressing, encouraging decoupling among the inter-communication among components, therefore meeting the requirements of several dynamic state of affairs, including mobile ad hoc networks (MANETs). Sadly, the properties of the CBR model are seldom met by existing systems that typically presume that application-level routers are prearranged in a tree-shaped network with a fixed topology. The paper proposes a new protocol, to organize the MANET nodes in a tree-like network that proposes eligibility to automatically repair to bear topological reconfigurations typical of MANETs and to attain this objective through repair strategies that diminish the changes that may impact the CBR layer exploiting the tree. This paper exhibits the implementation of MAODV and tree maintenance in overlay multicasting protocol Performance Analysis of the simulated scenarios.*

Keywords—*Distributed Computing, Content-Based Routing (CBR),Multicast Ad-hoc On-Demand Distance Vector(MAODV), Mobile Ad-hoc Networks (MANETs), Application Level Routers, On-Demand Multicast Routing Protocol(ODMRP).*

I. Introduction

CONTENT-BASED routing (CBR) differs from classical routing in that messages are addressed based on their content instead of their destination. In conventional systems, the sender explicitly specifies the intended message recipients using a unicast or multicast address. Instead, in CBR, the sender simply injects the message in the network, which determines how to route it according to the nodes' interests. These identify the relevant classes of messages based on their content, for example, using key-value pairs or regular expressions. Therefore, in CBR, it is the receiver that determines message delivery, not the sender[1].

MANET applications, such as emergency searches, rescues, and military battlefields where sharing of information is mandatory, require rapid deployable and quick reconfigurable routing protocols. I literature, there are two types of overlay structure for multicasting in ad hoc networks. A tree-based multicast routing protocol establishes and maintains a shared multicast routing tree to deliver data from a source to receivers of a multicast group. Two well-known examples of tree-based multicast routing protocols are the Multicast Ad hoc On-demand Distance Vector routing protocol (MAODV)[2], and the Adaptive Demand-driven Multicast Routing protocol (ADMR). But a mesh-based multicast routing protocol sustains a mesh consisting of a connected component of the network containing all the receivers of a group. Examples of mesh-based multicast routing approaches are the Core Assisted Mesh protocol (CAMP) and the On-Demand Multicast Routing Protocol (ODMRP). Former structure is vulnerable to high mobility, high load and large multicast group. Later one faces the problem of excessive control messages over the network. Some other multicasting protocols aim to restrict flood of control packets over the multicast network. Position-Based Multicast (PBM) routing protocol ignores the maintenance of distribution structure (e.g. tree or mesh).

It assumes that sender knows the location of destinations and each node has the position knowledge of its direct neighbours and its own as well. Multicast for Ad Hoc Networks with Swarm Intelligence (MANSI) is a biologically inspired protocol that adopts swarm intelligence to reduce number of nodes to construct the overlay structure. This report is organized into three groups according to the importance.

II. CONCISE OVERVIEW OF MULTICAST ADHOC ON-DEMAND DISTANCE VECTOR PROTOCOL

The MAODV (Multicast Ad-hoc On-Demand Distance Vector**)** routing protocol discovers multicast routes on demand using a broadcast route-discovery mechanism. A mobile node originates a Route Request (RREQ) message when it wishes to join a multicast group, or when it has data to send to a multicast group but it does not have a route to that group[2]. Only a member of the desired multicast group may respond to a join RREQ. If the RREQ is not a join request, any node with a fresh enough route (based on group sequence number) to the multicast group may respond. If an intermediate node receives a join RREQ for a multicast group of which it is not a member, or if it receives a RREQ and it does not have a route to that group, it rebroadcasts the RREQ to its neighbors. As the RREQ is broadcast across the network, nodes set up pointers to establish the reverse route in their route tables.

A node receiving a RREQ first updates its route table to record the sequence number and the next hop information for the source node. This reverse route entry may later be used to relay a response back to the source. For join RREQs, an additional entry is added to the multicast route table. This entry is not activated unless the route is selected to be part of the multicast tree. If a node receives a join RREQ for a multicast group, it may reply if it is a member for the multicast group's tree and its recorded sequence number for the multicast group is at least as great as that contained in the RREQ. The responding node updates its route and multicast route tables by placing the requesting node's next hop information in the tables, and then unicasts a Request Response (RREP) back to the source node. As nodes along the path to the source node receive the RREP, they add both a route table and a multicast route table entry for the node from which they received the RREP, thereby creating the forward path [2].

The protocol exploits four kinds of messages:

Route request (RREQ): This is broadcast by a node willing to join a specific multicast group, repair a branch of the tree, or merge two network partitions. It contains the identifier of the target group and the most recent sequence number known for it. When used to repair the tree, RREQ contains also the last measured hop distance from the leader to the sender.

Route reply (RREP): This is unicast toward a node that previously broadcast an RREQ to inform it that its request can be satisfied. RREP contains the identifier of the target group, its most recent sequence number known at the responding node, the identifier of the leader, and the current distance between the RREP sender and the leader. This information, along with the number of hops traveled by the RREP, is used to infer the new distance of the requesting node from the leader.

Multicast activation (MACT): This is unicast to explicitly activate a particular route toward the multicast tree. Furthermore, specific flags are used to implement operations such as identifying a new group leader after a failed repair, pruning a node from the tree, and updating the nodes' distance from the current leader. Node pruning is required when a forwarder node becomes a leaf. Instead, the nodes' distance from the leader must be updated whenever the tree topology changes (for example, when a broken link is replaced by a new one).

Group hello (GRPH): This is periodically broadcast by each group leader and rebroadcast across the whole network. Its main purpose is to disseminate the group sequence number and let each group member verify its distance (in hops) from the leader. It is also used to update the information at group members in case the group leader has changed, using a proper flag.

When a source node broadcasts a RREQ for a multicast group, it often receives more than one reply. The source node keeps the received route with the greatest sequence number and shortest hop count to the nearest member of the multicast tree for a specified period of time, and disregards other routes. At the end of this period, it enables the selected next hop in its multicast route table, and unicasts an activation message (MACT) to this selected next hop. The next hop, on receiving this message, enables the entry for the source node in its multicast route table. If this node is a member of the multicast tree, it does not propagate the message any further. However, if this node is not a member of the multicast tree, it will have received one or more RREPs from its neighbors. It keeps the best next hop for its route to the multicast group, unicasts MACT to that next hop, and enables the corresponding entry in its multicast route table. This process continues until the node that originated the RREP (member of tree) is reached. The activation message ensures that the multicast tree does not have multiple paths to any tree node. Nodes only forward data packets along activated routes in their multicast route tables.

The first member of the multicast group becomes the leader for that group. The multicast group leader is responsible for maintaining the multicast group sequence number and broadcasting this number to the multicast group. This is done through a Group Hello message. The Group Hello contains extensions that indicate the multicast group IP address and sequence numbers (incremented every Group Hello) of all multicast groups for which the node is the group leader. Nodes use the Group Hello information to update their request table. Since AODV keeps hard state in its routing table, the protocol has to actively track and react to changes in this tree. If a member terminates its membership with the group, the multicast tree requires pruning. Links in the tree are monitored to detect link breakages.

When a link breakage is detected, the node that is further from the multicast group leader (downstream of the break) is responsible for repairing the broken link. If the tree cannot be reconnected, a new leader for the disconnected downstream node is chosen as follows. If the node that initiated the route rebuilding is a multicast group member, it becomes the new multicast group leader. On the other hand, if it was not a group member and has only one next hop for the tree, it prunes itself from the tree by sending its next hop a prune message. This continues until a group member is reached. Once separate partitions reconnect, a node eventually receives a Group Hello for the multicast group that contains group leader information that differs from the information it already has. If this node is a member of the multicast group, and if it is a member of the partition whose group leader has the lower IP address, it can initiate reconnection of the multicast tree.

III. RELATED WORKS

Multicast communication

The work described here adapted the topology maintenance mechanisms of MAODV to a CBR scenario. However, here, we report about other proposals in the field of MANET multicast that are close to our requirements, that is, maintaining a flat (that is, no hierarchies or backbones) acyclic network in the presence of mobility. A comprehensive survey on the subject can be found in [3] and [4]. One way of achieving multicast communication in MANETs is to implement it on top of the MAC layer, therefore tackling mobility and link disruptions directly at the network layer. Alternatively, one can rely on some underlying multihop unicast mechanism providing point to-point communication and let this deal with mobility and reconfigurations.1 Notice how the second approach creates a layer of indirection hiding many aspects related to reconfiguration. Instead, we want to retain control of mobility, to tailor the broker tree reconfiguration to our needs. Inevitably, this implies removing any intermediate layer between the topology maintenance mechanism and the network itself. In the Ad Hoc Multicast Routing protocol utilizing Increasing ID-numbers (AMRIS), a bidirectional shared Tree is built by exploiting a ranking order among group members. The link repair process is somehow similar to MAODV, with the downstream node trying to reconnect by looking for a new parent node. The Core Assisted Mesh Protocol (CAMP) [5] and ODMRP [6] exploit mesh like topologies. With respect to the tree-shaped network provided by MAODV, they provide redundant paths at the expense of additional processing for maintaining multiple routes and discarding duplicates. Similar to MAODV, CAMP and Ad Hoc Multicast Routing (AM Route) require at least one special node for reconnecting lost partitions. ODMRP and MAODV have been extensively compared in [7], showing that the former provides better packet delivery at the expense of higher network traffic and, thus, reduced scalability. DCMP [8] is another source initiated multicast protocol that exploits a mesh topology similar to ODMRP. However, in this case, the control overhead is improved by dividing sources into active and passive. Active sources are responsible for creating a shared mesh also on behalf of the passive ones associated to them. ODMRP (On-demand Multicast Routing Protocol) [7] is mesh based, and uses a forwarding group concept (only a subset of nodes forwards the multicast packets). A soft state approach is taken in ODMRP to maintain multicast group members. No explicit control message is required to leave the group. In ODMRP, group membership and multicast routes are established and updated by the source on demand. When a multicast source has packets to send, but no route to the multicast group, it broadcasts a Join-Query control packet to the entire network. This Join-Query packet is periodically broadcast to refresh the membership information and update routes.

IV. COMPARING MAODV VS. ODMRP

The two on-demand protocols share certain salient characteristics. In particular, they both discover multicast routes only in the presence of data packets to be delivered to a multicast destination. Route discovery in either protocol is based on request and reply cycles where multicast route information is stored in all intermediate nodes on the multicast path. However, there are several important differences in the dynamics of the two protocols, which may give rise to significant performance differences.

First, MAODV uses a shared bi-directional multicast tree while ODMRP maintains a mesh topology rooted from each source. In MAODV, the tree is based on hard state and any link breakages force actions to repair the tree. A multicast group leader maintains up to date multicast tree information by sending periodic group hello messages. ODMRP provides alternative paths and a link failure need not trigger the re computation of the mesh, broken links will time out (soft state). Routes from multicast source to receivers in ODMRP are periodically refreshed by the source. However, a bi-directional tree is more efficient and avoids sending duplicate packets to receivers. Also, depending on the refresh interval in ODMRP, the control overhead from sending route refreshes from every source could result in scalability issues

Second, ODMRP broadcasts the reply back to the source while MAODV unicasts the reply. By using broadcasts, ODMRP allows for multiple possible paths from the multicast source back to the receiver. Since MAODV unicasts the reply back to the source, if an intermediate node on the path moves away, the reply is lost and the route is lost. However, a broadcasted reply requires intermediate nodes not interested in the multicast group to drop the control packets, resulting in extra processing overhead.

Third, MAODV does not activate a multicast route immediately while ODMRP does (unless mobility prediction is enabled). In MAODV, a potential multicast receiver must wait for a specified time allowing for multiple replies to be received before sending an activation message along the multicast route that it selects.

V. EVALUATION AND SIMULATION

To evaluate the effectiveness of COMAN, we measured its performance in both simulated and real settings. Therefore, we separate the evaluation into two parts. This section presents the results we obtained in simulated scenarios, with the goal of showing how COMAN is indeed more reliable and better suited to CBR than MAODV's tree maintenance strategy. Instead, Section 5 reports about our real implementation, demonstrating that our solution can be easily integrated into an existing content-based publish subscribe middleware and evaluating the performance obtained in a small-scale real deployment scenario.

Settings

We implemented our protocol using the NS-2 simulator[9]. It summarizes the most significant Parameters, along with their default values. Although most of them are typical of simulations in MANETs and do not require further discussion, it is worth detailing the strategy we adopted for modelling traffic. Indeed, our goal is to evaluate the ability of COMAN to maintain the CBR broker tree even in the presence of real network traffic beyond that needed for tree maintenance but regardless of any specific CBR strategy. Accordingly, we decided to have each node flood the entire network with "dummy" packets at a given rate. This traffic generates contention of the wireless medium and, therefore, collisions and message losses that stress COMAN's operations. At the same time, this does not require any specific assumption about the specific CBR strategy adopted. We run all simulations until a periodic evaluation of the variance of all measures is below 1 percent, which happens around 980 simulated seconds. It is known that this approach gives more precise results than simply repeating simulations with different seeds. As a mobility model, we employed Random Waypoint [8], as this was the model used in the MAODV papers.

Simulation Environment:

Number of nodes	50-100
Mac type	IEE802_112Mb/s
Channel	wireless channel
Network interface	wireless physical interface
Routing Protocol	MAODV
Flooding traffic rate	0.5Msg/s
Communication range	50m
Node Speed	5 m/s
Message Size	256 Bytes
Simulation area	1,250m
Warm-up time	60 s

Evaluating the Broker Tree

Before evaluating the CBR-specific features of COMAN, it is necessary to assess its ability to keep the tree connected at an acceptable cost. Therefore, we consider the following measures:

The percentage of time the tree remains fully connected (TC), that is, with all the nodes connected in a single tree. A link is considered broken when an underlying beaconing mechanism recognizes the absence of a neighbour.

The average number of control messages sent per tree repair (MS), This includes the RREQ, RREP, MACT, and GRPH messages used to repair a broken link along with the messages not strictly involved in the repair process, for example, the MACT and GRPH messages needed to update the distance from the leader.

The average number of nodes involved in a tree repair (NI), that is, the nodes that sent or forwarded at least one RREQ or RREP message. This measure gives an indirect measure of the amount of processing overhead our protocol imposes on the network.

Results:

Fig. 9a shows the trends with respect to the network density and the number of nodes in the system. As the network density decreases, the performance obviously degrades. In particular, Fig. 9a shows that TC rapidly decreases in sparse settings due to the lack of overall connectivity [1].

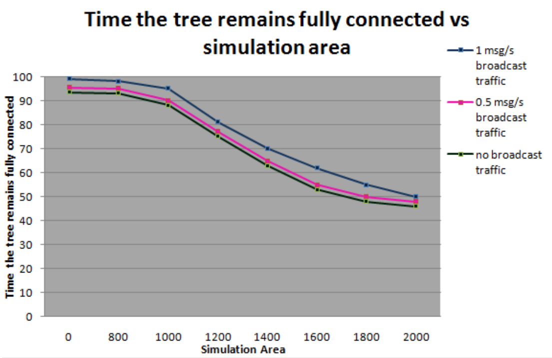

Fig.9a TC versus side of simulation area

On one hand, more nodes are involved when the network is dense because, being closely located; they are likely to hear the same RREQ message. On the other hand, we already pointed out how messages often travel farther in sparse networks, again involving more nodes [1]. The values between 1,250 m and 1,500 m represent the best trade-off between these two extremes. The network traffic does not seem to affect significantly the performance of our protocol, as the curves for various message rates shown in Figs. 9a and 9c are quite close to each other.

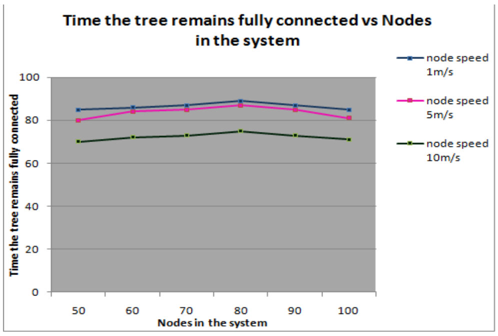

Fig. 9b TC versus nodes in the system and their speed

Conversely, Fig. 9b shows how TC varies with respect to the number of nodes in the system and their speed. TC initially increases with density until the network becomes so dense that packet collisions start to affect the protocol's ability to carry out the repair processes[1]. The same behaviour is exhibited at different node speeds. However, although there is little difference between scenarios with speeds of 1 m/s and 5 m/s, a speed of 10 m/s shows a more marked gap.

NI and MS are not shown in Fig. 9b as they turned out to be essentially independent from node speed. This indirectly supports our claims about the limited reconfiguration impact of our solution. Indeed, NI and MS are relative to single repair processes, and speed generally influences only their number. Therefore, our measures would vary with speed only in the case of concurrent overlapping link repair processes. However, the ability of our protocol to confine reconfigurations to a small portion of the tree makes the probability of concurrent overlapping link repair processes very low [1].

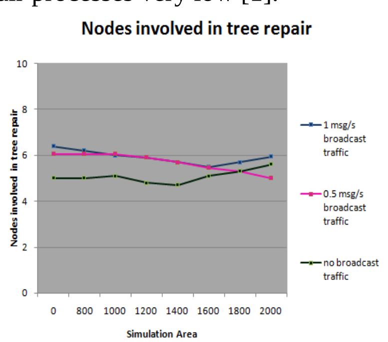

Fig 9c NI versus side of simulation area

Fig 9c The average number of nodes involved in a tree repair (NI), that is, the nodes that sent or forwarded at least one RREQ or RREP message. This measure gives an indirect measure of the amount of processing overhead our protocol imposes on the network.

VI. Conclusion and Future Work

In this paper, we presented and evaluated MAODV, a protocol for maintaining a tree-shaped network interconnecting the brokers of a CBR network in a MANET scenario**.** Moreover, it is also designed to minimize the number of brokers whose routing information is affected by topological changes, therefore improving the efficiency of the CBR network as a whole. It builds upon the tree maintenance algorithm found in the MAODV multicast protocol for MANETs. Results show that the protocol we propose meets the requirements for use in a CBR network and yields good performance [1]. The latter is significantly better than the original MAODV tree maintenance strategy, therefore showing that our solution does have a strong impact in achieving the desired properties of the broker network.

ACKNOWLEDGEMENT

We would like to convey our gratitude to Sri. P. Bhaskara Rao, Retd. Teacher, India, Nune Srinivas, faculty of SECE, Mrs. Pendem Padmaja, India, a special thanks to Mr. P.V. Subrahmanyeswara Rao, before their technical support to realize the this proposal discussed in this paper.

REFERENCES

[1] A Self-Repairing Tree Topology Enabling Content-Based Routing in Mobile Ad Hoc Networks Luca Mottola, Student Member, IEEE, Gianpaolo Cugola, and Gian Pietro Picco, Member, IEEE

[2] E. Royer, and C. E. Perkins, "Multicast operation of the ad hoc on demand distance vector routing protocol", In the proceedings of MobiCom, pages 207-218, Aug. 1999.

[3] C. Cordeiro, H. Gossain, and D. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," IEEE Network, vol. 17, no. 1, 2003.

[4] S.J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," Proc. IEEE INFOCOM '00, vol. 2, 2000.

[5] J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core Assisted Mesh Protocol," IEEE J. Selected Areas in Comm., vol. 17, no. 8, 1999.

[6] S.J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," ACM/Baltzer Mobile Networking and Applications, vol. 7, no. 6, 2002.

[7] K. Viswanath, K. Obraczka, and G. Tsudik, "Exploring Mesh and Tree-Based Multicast Routing Protocols for MANETs," IEEE Trans. Mobile Computing, vol. 5, no. 1, 2006.

[8] S.K. Das, B.S. Manoj, and C.S.R. Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks," Proc. ACM MobiHoc, 2002.

[9] NS-2 Simulator Web Page, www.isi.edu/nsnam/ns, 2006.