AMWN: A New Approach of Multicasting in Ad hoc Wireless Networks

Yash Pal Singh[1], **Vishal Nagar**[2],

[1]BIET Jhansi, [2]CSE Jhansi, (India)

SUMMARY

The aim if this paper is to introduce a new approach for multicast in Ad-Hoc Wireless Network. It is designed to work independent of underlying unicast protocols. In this a unique multicast ID is assigned to every participating node based on demand. The ordering in the flow of multicast is based on the ID assigned, and the sparseness among them used for quick connectivity repair. The delivery of multicast tree is rooted at a node called Sid joins up which is participating in the multicast session. There is a relation between the id-numbers (and the nodes that own them) and Sid, that the id-numbers increase in numerical value as they radiate from Sid in the delivery tree. This approach of id-numbers of a node leads to dynamically leave and join a session, also it adapts rapidly to changes in link connectivity. A message is provided to the region to recover from a breakage in case of any failure.

Key words: Multicast, DARPA, session-id, Neighbour node-Status, JOIN-REQ, JOIN-ACK

I.INTRODUCTION

In the current cellular wireless networks, the fixed infrastructure is used, for example the base stations are used to provide wireless access. It has wireless access in a single final hop, where users are connected and can communicate to base station through wireless network and their data routed is through some backbone connected to the base station. This approach is in contrast to multi-hop wireless networks (a.k.a. ad hoc networks) where this type infrastructure normally does not exist. The roots of Ad hoc networks exist in DARPA packet radio networks [1][2] from the 1970s. this new development in mobile computing, including wireless technologies, have changed the interest in the use and deployment of these networks.

This dynamic approach in the ad hoc networks means that current routing protocols [3][4] that have been developed for fairly static networks and cannot be operated well when deployed over ad hoc networks. There are some multicast routing protocols developed for ad hoc networks can be seen in [5][6]. However, they are based on an underlying unicast routing protocol.

In this paper, a multicast routing protocol is proposed that is developed for ad hoc networks, and that is free from the existing unicast routing protocol. Part II presents a general approach of the protocol. Part III discusses the simulation

model used to evaluate ad-hoc multicast, and their results are present in part IV.

II.AD-HOC MULTICAST IN WIRELESS NETWORK(AMWN)

It is an on-demand protocol which is based on a shared delivery tree to support multiple senders and receivers within a multicast session. Basic idea behind the current proposal and the existing protocols is unique that differentiates it from other multicast routing protocols, that is every member node in the multicast session has a sessionspecific multicast session member id (herein known as session-id). The session-id is provided to each node with a mark of its "logical height" in the multicast delivery tree. Every node other than the root must have one parent that has a logical height (session-id) that is smaller than it. Every participant finds its initial session-id dynamically during the Initial phase, which is initiated by a special node called Sid, who has the smallest session-id. Sid is normally elected from among the set of senders if there is more than one. The relationship between the session-id (and the node that owns it) and Sid (which is also the root of the tree) is that the session-ids increase in numerical value as they radiate away from Sid. The session-ids allow nodes that have broken off from the delivery tree to re-join the delivery tree in a local manner without creating permanent routing loops. The unique feature of AMWN is that it does not depend on the unicast routing protocol to provide routing information to other nodes. maintains a Neighbour-Status table which stores the list of existing neighbours and their session-ids. Each node sends a periodic beacon to signal their presence to neighbouring nodes. The beacon contains the session-ids that each node presently has.

AMWN consists of two main mechanisms: Tree

Starting and its Maintenance. Initialization of tree is the process by which a multicast routing session is created and circulated among nodes in the ad hoc network. The interested Nodes(In-Node) who want to join the multicast session, then join in the Initial phase. The Nodes which are not interested in joining this multicast session are herein known as Un-Nodes. It is important to note that Un-Nodes may join multicast session subsequently when it is necessary for them to function as "intermediate" nodes within the delivery tree to forward multicast traffic. The Maintenance of tree is the mechanism where nodes that become "disconnected" from the multicast delivery tree rejoin the tree to continue access the multicast network, by executing a Branch Reconstruction (BR) routine. The nodes which do not join the multicast session during the initial phase also make use of BR to join the tree. AMWN uses a soft state beacon approach to determine if a link has broken between two neighbouring nodes.

A. Initialization of Multicast Tree

Before the formal initialization of tree starts, it is important to determine which node will assume the role of Sp_id. In case of system where we may have singlesender and multiple-receiver, the significance of Sp_id is normally for the single-sender. For a multi-sender and multi-receiver scenario, Sp_id may be chosen from several the senders. The identification of Sp_id election are different then the identification of AWMN.

The initialization of tree starts with Sp_id broadcasting a NEW-SESSION(N_SESS) message to its neighbours. The N_SESS will contain, among other things, Sp_id'sSession_id and other parameters. All nodes which gets the N_SESS message generate their own Session-id by computing a value that is larger and not consecutive, so that there are gaps between the Session-ids of a sender and a receiver; these gaps are useful for quick local repair of the delivery tree. The node that receives that message then changes the Session-id of the message with their own and other routing paraments, before broadcasting the message again. Information derived from the N_SESS message is kept in the Neighbour-Status table for up to T1 seconds. A random uneven delay is introduced between the receipt of a N_NESS message and its subsequent retransmission to prevent broadcast storms. A participating node may receive multiple N_NESS messages from different nodes. If that particular node has not transmitted any messages yet, it will keep the message which contains the best routing metrics and calculate its session-id based on the values from that message.

Otherwise the received messaged are discarded.

A New Terminal N then joins the session by first determining from the NEW_SESS and messages received which neighbouring nodes have smaller session-ids than N. Such new terminals form the set of potential parent nodes. A new unicast J_ REQ is then transmitted to one of the potential parent nodes. When this potential parent M receives a unicast JOIN-REQ, it checks if M itself is already on the delivery tree. If so,

M will send a JOIN-ACK immediately back to N.

Otherwise, M too will try to find a potential parent for itself and send a J_REQ to it. This method is repeated until a node can satisfy the requirements of being a parent node. The node which propagates back along the reverse path towards M will send a J_ACK, grafting a branch from the tree to M. the process of joining is first attempted through contacting a neighbouring node; if that is unsuccessful , a local broadcast method is then used. In case the next neighbouring nodes are already on the multicast tree, then this 1-hop 'peek' approach is very fast and efficient. The application of session-ids helps a node to identify a neighbour (who as a potential parent) provides a higher likelihood of a successful join. In case the node is unable to find any potential parents, then the node who made this request will execute the Branch Reconstruction (BR) process in its continued attempt to rejoin the tree.

B. Maintenance of Multicast Tree

The process of tree maintenance mechanism works continuously in the background to confirm that a terminals remains connected to the multicast session delivery tree. When a connection between two nodes gets disconnected, the node with the larger Session-id is responsible for re-joining. A disconnected node tries to reconnect with the tree by executing the Branch Reconstruction (BR), which has two main subroutines, BRec1 and BRec2. BRec1 is processed when the node has neighbouring potential parent nodes that it can attempt to join to; BRec2 is executed when the node does not have any neighbouring nodes that can be potential parents. The actual work of BR1 as follows: The node N procession BR1 finds a potential parent node M from the set of potential parents. Now, it sends a JOIN-REQ to M; if M is already a registered member on the multicast tree and has a smaller msmid than N, it will send a JOIN-ACK back to N, acknowledging its request, and N has now successfully rejoined the tree. If Y is not yet a member on the tree, then it repeats the process of sending out its own JOIN-REQ to join the tree, provided it has at least one neighbouring potential parent node. Else, it transmits a JOIN-NEGACK back to X. If X gets a JOIN-NEGACK or timeouts on the reply, it will proceed to join with the next best potential parent node. If none are available, X executes the BR2 subroutine.

BRec2 is processed when a node X is unable to find out any neighbouring potential parent nodes. In place of sending a unicast JOIN-REQ to a single potential parent node (as in BRec1), X transmits a broadcast JOIN-REQ. The transmitted JOIN-REQ has a range field R that has only nodes within R hops of X are allowed to rebroadcast the JOIN-REQ. The objective of the R is to localize as much as possible the effects of a BRec routine without resorting to a network-wide broadcast in searching for new potential parent nodes. When a node Y receives a broadcasted JOIN-REQ, it checks if it can satisfy the request. If so, Y sends a JOINACK on the reverse path set up back to X. Since, the terminal Y does not forward multicast traffic to X yet, since X may receive more than one JOIN-ACK in response to its broadcast JOIN-REQ. As soon as X gets the JOINACKs (it may receive more than one from different nodes), this terminal will choose one of them to become the parent node and send a JOIN-CONF to that parent node. Now, the parent of this node receives the JOINCONF, it will now forward any multicast traffic it receives to its new child. If this terminal does not have a valid Session-id and wishes to join, it first uses neighbouring Session-ids to compute an msm-id for itself, then execute the BR routine to join the session.

III.SIMULATION ENVIRONMENT

In experiments on AMWN using an ad hoc network simulator written in PARSEC[7], which is a discrete event simulation language developed at UCLA. This multicast network consists of 100 mobile nodes moving about randomly (Brownian motion model) in a 1000x1000m two-dimensional space. In this the Radio transmission range was set at 150m. in this, the program simulated a CSMA MAC layer with a free space propagation model. Data rate was set at 2Mb/s. The first goal for the simulation was to understand the protocol's routing behaviour and detect any major flaws it has. Hence, a relatively light traffic model is used to minimize congestion effects. Every packet had a data portion (excluding headers) of 100 bytes and was generated at a rate of 1 per 100ms. The values on packet varied were the beacon interval (from 500ms to 4000ms), the number of I-Nodes per multicast session (from 25 to 100, one of which was randomly chosen to be Sid), and the maximum movement speed (from 1 to 20m/s). The values measured were packet delivery ratio (pdr), routing overhead and end-to-end delay. Every run simulated 200 seconds of simulation time.

IV.SIMULATION RESULTS

Each sample point given in each graph is an average from 20 simulation runs. Fig. 1 shows the packet delivery ratio with varying beacon intervals, membership sizes (membership size refers to the number of members in a multicast session.) and mobility rates. The delivery of packet ratio is fairly good, with most figures in the upper quartile range. In general, with increases in mobility, the packet delivery ratio decreases for all cases. This is because of the soft state nature of the protocol, which uses timeouts to determine that a neighbouring member node (which may be a parent or a child node) is no longer around.

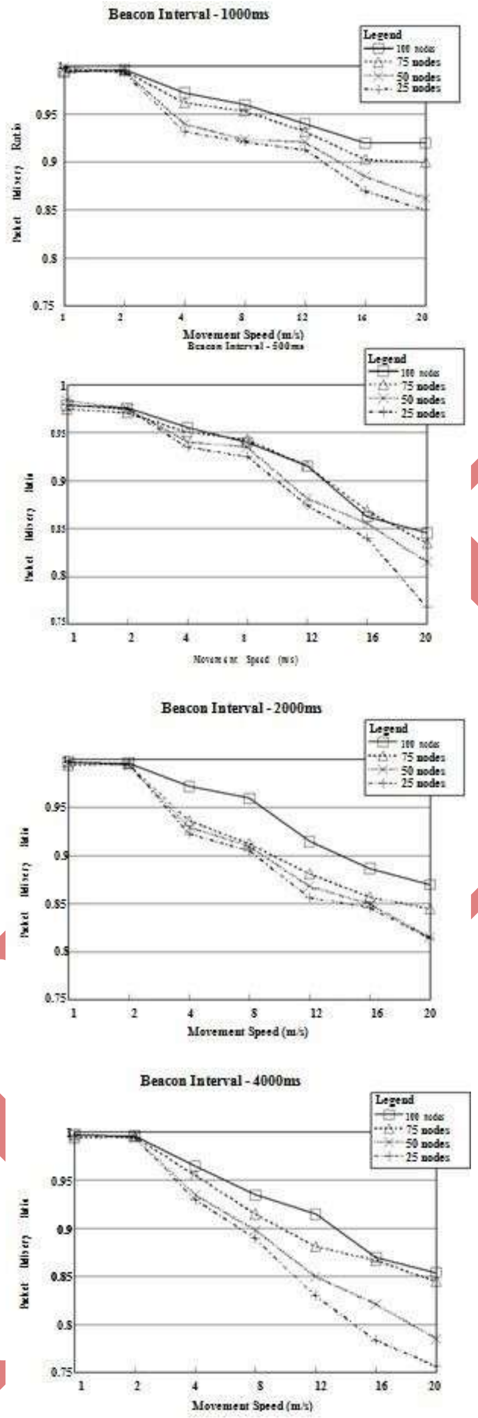

Fig.1. Packet Delivery Ratio

Because larger beacon intervals are used, the packet delivery ratio drops significantly more at higher mobility rates. The value off timeout value that determines a neighbouring node is set as a multiple of the beacon interval (we set it at 3). Hence, a large beacon interval, a node takes significantly longer to realize that its neighbour parent node has moved away. This results into a significantly higher number of packets not received at large beacon intervals. For small membership sizes, drop is larger because there is a smaller number of potential parent nodes around a node when it tries to rejoin the tree. In case if membership sizes is large, when a node discovers the breakage, it can usually find a parent node nearby, and can rejoin the tree more quickly. With more I-Nodes the Multicast sessions also generally perform better than those with less I-Nodes since nodes are better able to quickly find neighbours that are already registered on the tree. The Nodes which are close to Sid have a higher pdr than those further away since they are usually within a single hop from Sid. The Sessions which has more I Nodes also have more nodes that are close to Sid, thus increasing the pdr.

It is noteworthy that with a small beacon interval of 500ms, the packet delivery ratio drops significantly at higher mobility rates compared to other beacon intervals. Also, during the investigation it was found that when the beacon interval is small, there is an increase in the number of link breakages being detected. When the nodes move about in a random fashion, they frequently move just out of range of each other for just a short while (micro-term) before moving back into range again. Because of above mentioned link breakages are more evident when the beacon interval is small. This results as the nodes to execute Branch Reconstruction (BR) to rejoin the tree. This increase in packets sent leads to increased packet collisions around those nodes, causing packets to be dropped and decreasing the packet delivery ratio.

The graphs given in Fig.- 1 show that there is an optimum beacon period that should not be too small or too large. More studies need to be done to find out the relationship between this optimum and the node densities, movement speeds and traffic models. Fig.-2 shows the results for routing overhead. Overhead of routing is calculated as the ratio of control packets (e.g. JOIN-REQs, JOIN-ACKs; the beacons are not included as they are a constant overhead) sent versus all (data and control) packets sent. Calculation of is done packets instead of bytes because the control packets are small in size (around 20 bytes) compared to the data packets (100 bytes). If we measure ratio of control bytes versus total bytes, then the routing overhead would be very small. Counting the number of bytes would have ignored the cost of acquiring the medium to transmit a packet, which is relatively independent of packet size once the medium has been acquired. Estimating packets therefore provides a clearer view of routing overhead.

If we consider the value of the beacon interval as small, there is noticeably higher routing overhead. The reason of this is due to more nodes superfluously initiating BR as a result of micro-term link breakages. This hike in packet collisions result in dropped data packets which further increases the ratio of control packets to total packets.

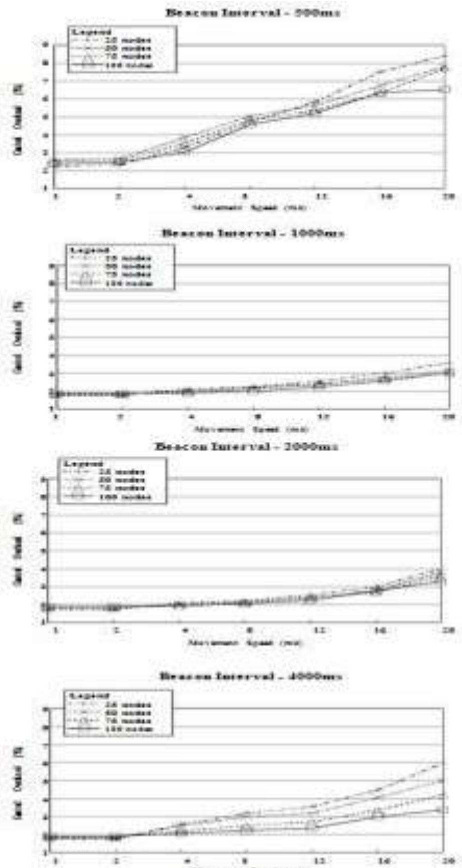

Fig.2. Control Overhead

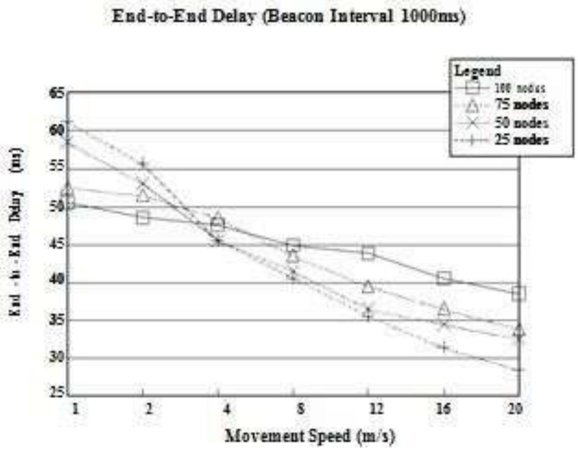

Fig. 3. End to End Delay

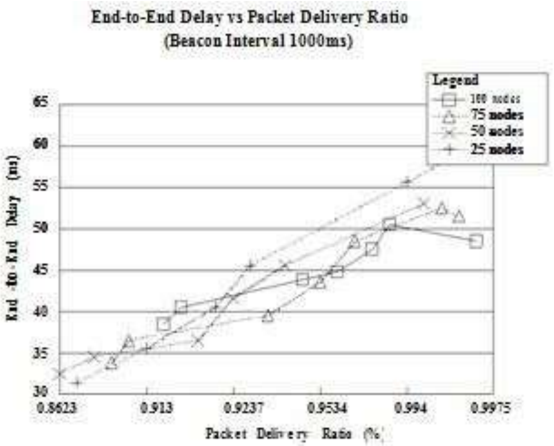

Fig.4. End-to-End Delay vs PDR

Considering a mobility rates, a large membership has lower routing overhead compared to a small membership. Further, this is because of the localized repair feature of BR which queries neighbouring nodes rather than doing a localized nhop broadcast. The overhead due to routing results again show that there is an optimum beacon period.

A delay of End-to-end point is considered as the average time taken by a packet to reach an I-Node from the time it leaves the sender. In practical scenario nodes closer to the sources will usually have a smaller end-to-end delay than nodes further away. The calculation of metric is done as follows: When a data packet is first created by the source, it is tagged with a send time. Further, each if the I-node that receives the packet calculates the end-to-end delay by subtracting the time the packet was received with the initial send time. The average value is then taken from all I-nodes. The delay of end points is thus measured only for packets that are received. Which results, in Fig: 3, as the maximum movement speed is increased, the end-to-end delay actually drops. The results in Fig: 4 shows explicitly this relationship between end-to end delay and packet delivery ratio. The results given in curves given for all four membership sizes are clustered together, thus showing that the relationship between the two metrics is robust with respect to membership.

To move a data packet to travel a hope on average 5ms in needed. The motive for the 5ms per hop (a large value) is because a random jitter with maximum of 50ms is introduced between data packet reception and retransmission. Therefore, with a maximum average end-toend delay of 62ms for 25 receivers, we can estimate the average hop traversed by the data packet along the delivery tree.

V.CONCLUSION

In this paper, we proposed a new multicast routing protocol designed for use in ad hoc networks. AMWN governs the communication between the nodes within the multicast delivery tree logically with a dynamically derived number. This protocol can be used to direct multicast traffic, and the sparseness among these numbers facilitates quick local repair to the delivery tree. The basic results of simulation results show that AMWN has high delivery ratio and low overheads, and is thus feasible as a multicast routing protocol for ad hoc networks.

The simulations of protocol show that some improvements are possible. One of the improvement is the criteria for selecting which potential parent node to send the JOIN-REQ if there is more than one to choose from.

Currently, we choose the node with the smallest Session-id. Although, from the simulation, we observe that this may not give a good route: a potential parent with the smallest Session-id may be further away, so the link may get weaker than if another nearby potential parent is selected.

REFERENCES

[1] Leiner, B.M, Neison, D.L. and Tobagi, F.A, "Issues in Packet Radio Network Design". Proceedings of the IEEE
Special issue on "Packet Radio Networks",, 75,1:6-20, 1987

[2] Jubin, J and Tornow, J.D, "The DARPA Packet Radio Network Protocols", In Proceedings of IEEE, volume 75, 1, pages 21-32, Jan. 1987

[3] Deering, S.E., Partridge, C., and Waitzman, D., "Distance vector multicast routing protocol", RFC 1075, Nov 1988

[4] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., "Protocol Independent Multicast-
Sparse Mode (PIM-SM): Protocol Specification",
RFC2117, June, 1997

[5] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks", ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, vol. 1, no. 2, 1998.

[6] C.-C. Chiang, M. Gerla and L. Zhang, "Shared Tree Wireless
Network Multicast", In Proceedings of IEEE IC3N
'97.

[7] Meyer, R.A., PARSEC User Manual, August 1998.

[8] AMRIS: A Multicast Protocol for Ad hoc Wireless Networks, C.W. Wu, Y.C. Tay National University of Singapore Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE