Global Journal of Computer Science and Technology
Network, Web & Security
Volume 12 Issue 13 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

MALMR: Medium Access Level Multicast Routing for Congestion Avoidance in Multicast Mobile Ad Hoc Routing Protocol

By G.S.Sreedhar & Dr.A.Damodaram
Academic Audit Cell, JNTUH, Hyderabad

Abstract - This paper is focused on a new solution for congestion avoidance in ad hoc multicast routing by bearing the congestion situations. As the routing strategy belongs to Medium Access Level, the routing strategy is named Medium Access Level Multicast Routing short MALMR. MALMR is aimed at Congestion Avoidance in Multicast Mobile Ad hoc routing protocol. The present MAC level routing strategy is independent which can work with any multicast routing protocol irrespective of tree or mesh structure. During the study of MALMR performance, the MALMR tested along with On-Demand Multicast Routing Protocol where simulation results proved that MALMR raises the performance of ODMRP in order of magnitude.

Keywords : Multicast, on-demand routing, congestion control, ad hoc network, MALMR.

GJCST-E Classification: C.2.2

MALMR MEDIUM ACCESS LEVEL MULTICAST ROUTING FOR CONGESTION AVOIDANCE IN MULTICAST MOBILE AD HOC ROUTING PROTOCOL

Strictly as per the compliance and regulations of:

MALMR: Medium Access Level Multicast Routing for Congestion Avoidance in Multicast Mobile Ad Hoc Routing Protocol

G.S.Sreedhar [α] & Dr.A.Damodaram [σ]

Abstract - This paper is focused on a new solution for congestion avoidance in ad hoc multicast routing by bearing the congestion situations. As the routing strategy belongs to Medium Access Level, the routing strategy is named Medium Access Level Multicast Routing short MALMR. MALMR is aimed at Congestion Avoidance in Multicast Mobile Ad hoc routing protocol. The present MAC level routing strategy is independent which can work with any multicast routing protocol irrespective of tree or mesh structure. During the study of MALMR performance, the MALMR tested along with On-Demand Multicast Routing Protocol where simulation results proved that MALMR raises the performance of ODMRP in order of magnitude.

Keywords : Multicast, on-demand routing, congestion control, ad hoc network, MALMR.

I. Introduction

Great numbers of routing protocols for Ad Hoc network are presented by classifying from many aspects. Protocols are of three types such as reactive protocols, proactive protocols and composite protocols that integrate the discovering process of the above ones. Depending on the structure of network topology, the protocols are divided in two types. They are plane ones and clustering ones. Depending on load balance mechanism, the protocols are grouped as single path ones and multi-path ones. A great number of routing protocols such as Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector Routing (AODV), Destination-Sequenced Distance-Vector Routing (DSDV), Temporally Ordered Routing Algorithm (TORA) and Zone Routing Protocol (ZRP), are identified but the possibility of using them and their efficiency remained doubt in view of the only min-hop metric as routing selection criterion. MANET has no difference between a host and a router, because all nodes are senders or receivers and also forwarders of traffic where all MANET members can deleted easily. Having high mobility nature, MANETs can be used in the environments, which need robust and reliable capacity like military battlefield, emergency rescue, vehicular communication, mining operations etc. For these applications, multicast is paramount and helpful in holding down network bandwidth and resources, as one message from one source can be sent to multiple receivers at a time. The main risk for multicast routing in MANETs is maintaining of robust capacity even in the condition of frequent, mainly high-speed agility and nodes outages. So mesh-based protocols builds a mesh for forwarding multicast packets for sending even in the presence of links breaking, and reaches robustness and reliability demands with path repetition owing to meshes on networks. Present multicast routing protocols for MANET is divided into two types: tree-based and mesh-based protocols. The tree based ones, i. e. MAODV (Multicast of Ad hoc On Demand Distance Vector) generally have tree-based schemes, unfits high-speed ad hoc networks. Common mesh-based multicast routing protocol is ODMRP (On-Demand Multicast Routing Protocol)[2], that uses the concept of forwarding group, builds multicast mesh that is done in soft state and acquires high performance [3, 4]. In [5], V. Kumar, et al. obtains comparative conclusions about MAODV and ODMRP based on the simulation results. Even though the performance of all multicast protocols degrade in terms of packet sending and group reliability as node mobility and traffic load augments, mesh-based protocol ODMRP do better job than tree-based protocol MAODV. ODMRP can bring forth decent robustness based on its mesh structure. MAODV performs less when compared to other protocols in packet delivery ratio and group reliability.

II. Related Work

Based on the researches on real life Ad Hoc network and references, it is seen that a prodigious of real life Ad Hoc networks works without following rules which are included in our theoretical analysis. In contrast to above, selfish nodes hinder the network process. The selfish nodes are intended in participating of the natural network information exchange procedure like routing discovery, routing maintenance and packets forwarding etc. The reason for selfishness of these nodes comes from the various advantages of various organizations who own the various groups of nodes. Because of the existence of selfish nodes, a few relay nodes remains as “hot spots” which leads to “death” due to power decrease resulting in total disabling of entire network.

Author α : Professor, Department of CSE, K.O.R.M.College of Engineering, Kadapa. E-Mail : gssreedhar9@gmail.com
Author σ : Professor of CSE & Director, Academic Audit Cell, JNTUH, Hyderabad. E-Mail : damodarama@rediffmail.com

Prashant Dewan et al. said in [6] that, particular nodes in an Ad Hoc network might become antagonistic and thus refuse to cooperate with each other. In addition to, Ad hoc network posses semiautonomous nodes owned by different entities may not be distributed with common goal, and thus the nodes may not work together which is supposed to do. In [7], Buttayan et.al; presented "Nuglets" protocol for reducing the impact of nodes selfishness on entire network performance. The effectiveness gets "rewards" and in efficiency will be given "penalties"; In [8],"SPRITE" protocol is designed to control the selfishness by constructing a credit clearing service(CCS) server which provides a credit to every node in the network. The node selection depends on its credit for path. The above mentioned protocols focused on selfishness of nodes and how to overcome this selfishness. Whatever may so, implementation contains complexity in the channel fading, retransmission and collision etc. Thus these protocols remain incompetent.

III. Medium Access Level Multicast Routing (MALMR)

The core point of MALMR is reliability transmission of every packet to every neighbor. The presented MALMR designed using a transmission window structure that influenced by IEEE 802.11 transmission structure, thus a brief operational overview of 802.11 transmissions is as follows.

a) IEEE 802.11 Transmission Overview

IEEE 802.11 used a collision avoidance scheme including RTS/CTS/ACK control frames for transmission of unicast packets. In 802.11, the Distributed Coordination Function (DCF) shows the basic access method that mobile nodes uses for sharing wireless channel. The scheme combines CSMA with Collision Avoidance (CSMA/CA) and acknowledgement (ACK). The mobile nodes based on need they can use the virtual carrier sense mechanism which provided RTS/CTS exchange for channel reservation and fragmentation of packets in situations. The CSMA/CA works in transmission of senses the channel. If the channel is free for a time equal to the DCF Inter Frame Space (DIFS) interval, the node transmits. If the channel is busy, the node enters a state of collision avoidance and backs off from transmitting for a specified interval. In the collision avoidance state, the node sensing the channel busy will suspend its back off timer, only resuming the back off countdown when the channel is again sensed free for a DIFS period. Common sequence of exchanges in 802.11 utilizing the virtual carrier sensing mechanism contains the source node first sensing the channel utilizing CSMA/CA. After the execution of CSMA/CA, RTS is transmitted by the source node, which follows responding of node with CTS, after responding the source node sends the data frame and subsequently with the conformation of destination node with an ACK to the source node. Receiving RTS, CTS or data frame is not real destination of any node but it should complete the data exchange is real destination of node. For broadcast packets, IEEE 802.11 nodes simply execute collision avoidance and then transmit the data frame.

b) MALMR Transmission Window Structure

MALMR, multicast node nm need to manage two lists, target nodes list (TNL), which is hop level destinations, Frames Sent (FS), transmission history. Receptions of frames (RTS/CTS/DATA/ACK/HELLO) are used by nm to maintain track of its targeted nodes. Every nm also maintains a FS. The FS contains copies of the frames which are already transmitted which may be need even later for retransmission. After receiving by neighbor a copy will be removed from the FS. FS size must be larger than targets number for any nm. In addition to the FS, there is possibility of storing yet to be transmitted packets which is called. Every target node maintains a list for frame received which is symbolized as FR. FR stores when a target node receives a new frame, it records the frame's sequence number in FR. When a nm node transmits RTS to a destination node specifying a range of (from and to) sequence numbers, the destination node examines its FR to determine whether it is missing any previous sequence numbers in the specified range. If so, the destination node replies with the missing sequence number in the CTS response. Generally in MALMR, If an "nm" has to transmit a packet, it should first test the channel and then a collision avoidance (CSMA/CA) step like that of 802.11. After the collision avoidance step completion the channel becomes free, the nm sends RTS to its target picked from TNL from particular range of sequence numbers which are already sent where the present sequence number is to be transmitted. All the process will be achieved by pulling the least sequence number from the FS and defining it into the RTS frame with the present sequence number which is expected by the source node. After receiving the RTS, the determine target test its FR and decides the needed sequence numbers. CTS response frame will react if the target node doesn't find precedent sequence number.

Similarly, CTS response frame will also react even in case of present sequence number. All other targets hearing the RTS will yield long enough for the CTS/DATA/ACK transmission. Upon the receiving of the CTS, the "nm "transmits the DATA (packet) according to the sequence number determined in the CTS frame. After receiving the DATA, the target node updates its FR and answer with an ACK. Remaining neighboring nodes which receive the DATA updates their FR. After receiving the ACK, if the DATA sent DATA is wrong but obtain from the buffer, the source node its process with the destination node with another RTS until the present

DATA is sent from the queue. After transmission of the present DATA and acknowledged, the source node then buffers the packet and chooses the next neighbor in its NEIGHBOR LIST and repeats the whole process over again then the collision avoidance step is neglected. In MALMR, ordered first strategy is used that picks a target node from TNL chronologically. During this process target nodes order changes depending on their current ingress ability status. The ordered first strategy sends packets and works comfortably. If, there are no packets for transmission of queue, the ordered first process will be stopped until next target in the TNL received all the broadcast DATA until there is a new packet to send. For preventing this, MALMR utilizes flag cs that set to true and then next node in the TNL will be selected and the ordered first process repeats. In between if new sequence numbers joined then flag cs will be set false then remaining targets are visited in the ordered first process without considering the current sequence numbers if any. Upon the completion of RTS to all targets in ordered first process if still no new sequence numbers are identified then ordered first process stops the RTS process till there a new packet is ready for transmission. If new sequence numbers is identified then the flag cs sets wrong and ordered first process repeats.

IV. CONGESTION AVOIDANCE IN ODMRP USING MALMR

A few multicast routing protocols contains AMRoute[9], AMRIS[10], CAMP[11] , multicast AODV[12], and the On-Demand Multicast Routing Protocol (ODMRP)[13, 14, 15]. ODMRP distracts multicast packets on a mesh in place of the traditional multicast tree. By utilizing a mesh, ODMRP bring out excess to combat packet loss in ad hoc networks where channel noise, collisions and mobility are universal. With low traffic load, ODMRP does efficiently. Nevertheless, as traffic load augments, ODMRP continuously suffers from network congestion. Though this disadvantage is not limited to ODMRP it is wide spread among other multicast protocols. The present paper introduces a new MAC protocol, MALMR that allows reliable MAC broadcast in ad hoc networks. In addition to, by excess using MALMR, it is said that congestion control in ODMRP decreases network load when contention is high. This MALMR is not limited only to ODMRP but can apply even on other multicast protocols, like multicast AODV. ODMRP protocol is explained in the sub section I and ODMRP with our congestion avoidance scheme MALMR is explained in the sub section ii where simulation results are provided in section 4. Subsequently, section 5 explains the conclusion of the paper.

a) On-Demand Multicast Routing Protocol

ODMRP creates a group-shared forwarding mesh for every group. Every source carries out periodic flood-response cycles creating multicast forwarding state without depending on present forwarding state. The frequent state discovery helps the protocol to find the present simple paths between every source and the multicast receivers and develops the boisterous protocol due to multiple forwarding paths may present between group members. Due to this ODMRP's packet send number of sources and receivers per multicast group augments and even sometimes increases mobility: the repeat forwarding state devises ODMRP's packet produces ability due to it behaves error correction, and does the protocol robust to mesh. Nevertheless, the frequent identification produces and great number of data transmissions identically augments network load.

Ever multicast source for a group G in ODMRP regularly moves the network with a JOIN QUERY packet that forwards by all nodes in the network. REFRESH INTERVAL, e. g., every 3 seconds send by this packet. Every multicast receiver reacts to this flow by delivering a JOIN REPLY packet that is forwarded in a simple path back to the multicast source that started the QUERY. Anterior of forwarding the packet, every node waits for JOIN AGGREGATION TIMEOUT, and mixes all JOIN REPLYs for the group received during this time into one JOIN REPLY. Every node that forwards the REPLY packet generates (or refreshes) forwarding state for group G.

Every node with forwarding state for G forwards every data packet delivered by a multicast source for G. A data packet use the simplest paths to the multicast receivers within the forwarding mesh, it may even forwarded to other sources of the group who are group members. Forwarding state is ceased after a multiple of regular breaks to assure that in the event that some number of forwarding nodes' multicast state is not refreshed due to packet loss, the forwarding state generated from a earlier flood is also authenticated. This mechanism develops the boisterous protocol, where many overlapping trees will be activated in the network parallel; everything is produced finally by JOIN QUERY flood [16].

i. *Multicast route discovery*

In MALMR, route finds is started and managed by the source. When the source contains packets for transmission for a certain multicast group, the source first decides if there is a route to the group. If a route is not present, MALMR tries to create one through the route finding process. The route finding process is equal to on-demand unicast routing protocols like AODV [17] and DSR [18]. Route discovery process has two steps.

a. Request Round

In the request round, the source moves the network with a member advertisement packet with the data piggybacked which is named as JOIN QUERY. These JOIN QUERY packets regularly broadcast to the total network to refresh membership information and recreate new multicast routes. After receiving a non-duplicate JOIN QUERY, a node inserts or updates in its ROUTING TABLE the upstream node indicates as the next node to the source node. The ROUTING TABLE can also be utilized even in a JOIN REPLY depending on need of the source during the reply round, called as backward learning [19].

b. Reply round

After reaching a non-duplicate JOIN QUERY to multicast member the reply round starts. In the reply round, the multicast member generates and broadcasts JOIN REPLY packet to the network with the address of the node the member receives the JOIN QUERY from stamped in the JOIN REPLY. After receiving the JOIN REPLY, a node decides if its address is stamped in the JOIN REPLY. If so, the node knows it is on the path to the source and set the path FORWARDING_GROUP_FLAG and be a part of the forwarding group. After that, the node resends JOIN REPLY with the upstream node address to the source stamped in the JOIN REPLY. The upstream node address is got from the ROUTE TABLE via backward learning. This process goes on until the JOIN REPLY meets the source. The source receives JOIN REPLY, a mesh of nodes, or forwarding groups, is formed and packets can be sent to the members.

c. Route maintenance

ODMRP manages the group by regular broadcasting JOIN QUERY to the network and receiving JOIN REPLY. The regular broadcast of JOIN QUERY updates the forwarding group nodes and takes membership fluctuations.

b) On-Demand Multicast Routing Protocol With Congestion avoidance

To accomplish congestion avoidance, MALMR is utilized as the underlying MAC layer. MALMR is needed because ODMRP broadcasts data packets to all neighbors in spite of sending them point-to-point to choose individual neighbors, as causally done by multicast protocols. The underlying MAC protocol utilized for broadcast, CSMA avoiding ACK. CSMA, the queue length will denote perfect measure of congestion. Broadcast packets are sending "blindly". If the packet is not reached because of receive-buffer excess flow or channel congestion. it is stopped and no retransmission is done. Accordingly, even in presence of congestion, the queue length is small. In opposite to, the version of the IEEE 802.11 protocol utilized in unicast, point-to-point transmissions is filled with RTS and CTS control packets and ACKs. It is used against receive-buffer overflow and hidden terminals, and thus supplies perfect congestion feedback. This unicast version accordingly is not so attractive for multicast applications because it cannot misuse "broadcast advantage" of the wireless channel, and needs an individual transmission to every multicast member. Hence, MALMR is required to perfect description of the network state via queue lengths as MALMR supply reliable delivery of packets that are broadcasted in the context of multicast.

i. Congestion Avoidance

MALMR effectively eliminates the congestion by adapting to ordered first sequence to cast the packet all target nodes in broadcast manner. Here the MALMR process surveyed with an example.

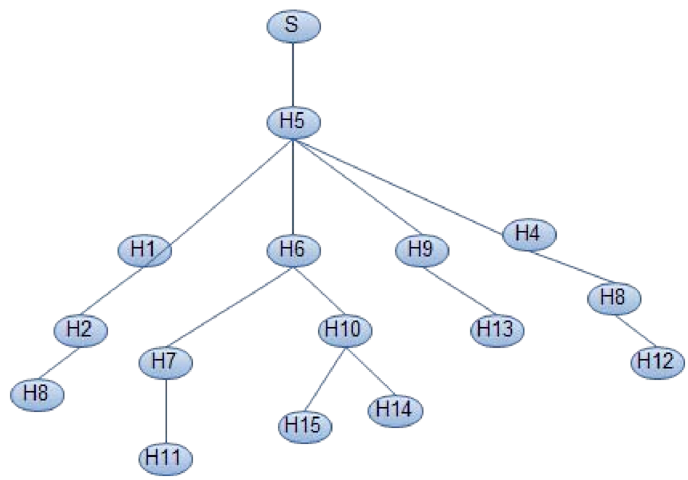


Fig. 1 : Tree Representation of example multicast model

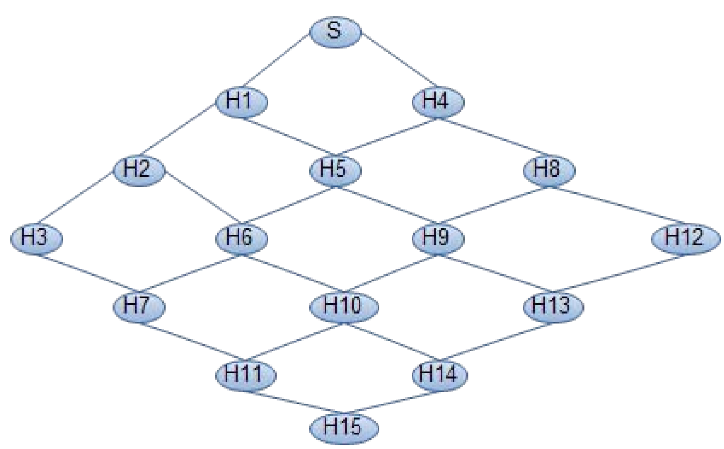


Fig. 2 : Mesh Representation of example multicast model

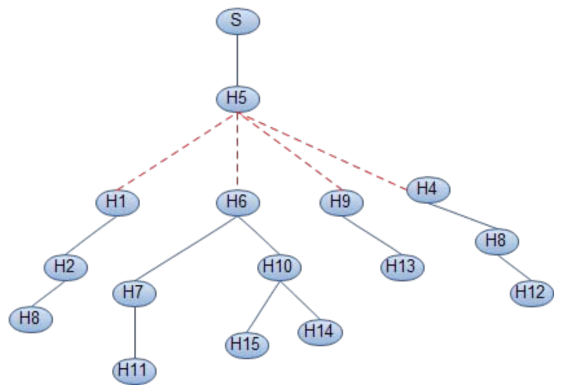


Fig. 3 : An example of group formed in tree based multicasting

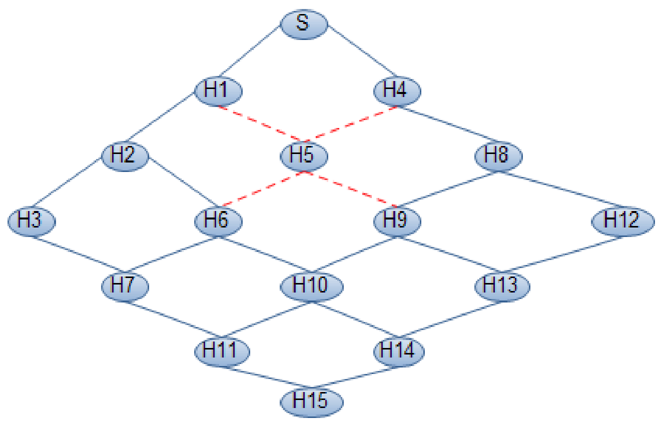


Fig. 4 : An example group formed in mesh based multicasting

The concept of MALMR is explained with an example. First of all we take a tree based multicasting as in figure 1 or mesh based multicasting as in figure 2. The fig 3 and fig 4 denotes a group linked with dotted line formed during path finding. For descriptive convenience same group denotation in tree and mesh (fig 3 and fig 4) is used. We now imagine that node H5 desires to multicast packets to H1, H4, H6, H9 in Figure 3 and fig 4. At the stage of transmitting first packet Node H5 first decides a target node H1 from TNL and sends RTS with sequence numbers ranging from f0 to f0 since no DATA frames have yet been sent. Node H1, upon receiving the RTS frame, responds with sequence number f0 in the CTS frame. Nodes H6, H9, and H4, after receiving the RTS frame, gives for the CTS/DATA/ACK interchange between node H5 and H1. After receiving the CTS frame, node H5 multicasts DATA with sequence number 0 in broadcasting manner. Node 1, after receiving DATA, updates its FR and responds with an ACK. For explanation requirement, a node H6 did not receive the DATA (possibly due to interference from neighboring nodes) while node H9 and H4 received the DATA perfectly. Hence, node H9 and H4 also update their RF. After receiving the ACK, node H5 copies the DATA that was delivered into the FS and goes on to choose nodes from TNL in ordered first form. If we imagine that node H6 chosen in order as immediate neighbor for transmission after executing the collision avoidance round, node H5 delivers RTS with sequence number range f0 to f1. After receiving the RTS, node H6 looks its FR and noticed that frame f0 haven't received. Node H6 then delivers CTS desired sequence number f0. Node H5, after receiving the CTS, gets the DATA with sequence number f0 from the FS and transmits the DATA. After receiving the DATA, node H6 updates its FR and responds with an ACK. Upon receiving the ACK, node H5 delivers RTS repeatedly with sequence number range f0 to f1 since the most recent DATA will not been sent. Node H6, after receiving the RTS, delivering CTS with sequence number 1 after checking its FR. Node H5, upon receiving the CTS, sends the DATA with sequence number f1. Node H6, after receiving the DATA, response with an ACK Again, for explanation process , let's say nodes H1, H9 and H4 receive the DATA and update their respective FR. Node H5, after receiving the ACK, buffers the DATA in FS and selects node H9 as its immediate neighbor. After the collision avoidance round, node H5 transmits RTS with sequence number range f0 to f2. After receiving the RTS, node H9 inquiry its received sequence number list and delivers CTS requesting sequence number f2 (since f0 and f1 were successfully received previously). Node H5, after receiving CTS, transmits DATA with sequence number f2. Node H9, after receiving DATA, transmits ACK and updates its FR. Node H5, after receiving ACK, buffers the DATA in FS, choose node H4 as it's immediate neighbor to transmit to, and the process starts again . In this process if node H5 found that no data with new sequence numbers available, then it set flag cs true and goes on delivering RTS with sequence range already delivered and cached in FS to nodes in TNL in ordered first form. After sending RTS to all the nodes in TNL, checks for data. If still no data with new sequence numbers then this process stops till it discovers data with new sequence numbers. If data is discovered with new sequence number then flag cs sets to wrong and promotes multicast process.

ii. *MALMR Algorithm*

Description of the notations

I. nm ←Node participating in multicasting
II. nu ←Node participating in one of the unicasting path of nm
III. TNL ←Target Node List
IV. bp_{nm} ←Buffer of Packets to multicast at nm
V. FS_{nm} ← Buffer of Frames already sent by nm
VI. FR_{tn} ←Buffer of frames received by target node tn that listed in TNL
VII. cs ← Boolean flag

Input:

$$TNL\ ,\ bp_{nm}\ ,cs \leftarrow true$$

Algorithm:

1. Begin
2. Fetch $\{tn_i \mid tn_i \in TNL\}$ that fetched in ordered first manner for $i = 1....|TNL|$
3. Fetch sequence numbers range $fo,....fl$ of the frames such that $fj \in FS$ for each $j = 0,...l$
4. If bp_{nm} is not empty
5. Begin
6. Set $cs \leftarrow false$
7. Pick next sequence number sn of the packet to be multicast.
8. Send sequence numbers range $\{fo,...fl,sn\}$ to tn_i and wait for response from tn_i
9. Receive the sequence number rsn of the frame from tn_i
10. If $rsn \cong sn$
11. Begin
12. Multicast new packet from bp_{nm} and wait for acknowledgement from tn_i
13. End of block Started at line 3
14. Else if $rsn \in \{f0,...fl\}$
15. Begin
16. Multicast cached frames of range $\{rsn,....fl\}$ in a sequence. And then multicast new data packet from bp_{nm} with sequence number sn
17. End of block Started at line 4
18. End of block Started at line 2
19. Else if bp_{nm} is empty and $cs \neq true$
20. Begin
21. Set $cs \leftarrow true$
22. Fetch $\{tn_k \mid tn_k \in TNL\}$ that fetched in ordered first manner for $k = i....|TNL|$
23. Begin
24. Fetch sequence numbers range $fo,....fl$ of the frames such that $fj \in FS$ for each $j = 0,...l$
25. Send sequence numbers range $\{fo,...fl\}$ to tn_k and wait for response from tn_k
26. Receive the sequence number rsn of the frame from tn_k
27. If $rsn \in \{fo....fl\}$
28. Begin
29. Multicast cached frames of range $\{rsn,....fl\}$ in a sequence.
30. End of block Started at line 7
31. End of block Started at line 6
32. Set $i \leftarrow k$
33. End of block Started at line 5
34. Else if bp_{nm} is empty
35. Halt a time interval ti and go to step 1
36. End of block Started at line 1

In step 12, 16 and 29 all nodes of list TNL also receives those frames and according their respective FR status they update FR, that is if the nodes not found that frame in their respective FR then updates otherwise discards.

In step 12 and 16, if acknowledgement received from target node tn_i then the node nm updates it's FS_{nm} by adding new sequence number to FS_{nm}

V. Simulations and Results Discussion

NS 2 is used in doing experiments. We create a simulation network with hops under mobility and count of 50 to 200. The simulation parameters explained in table 1. Authentication ensures that the buffer is properly allocated to valid packets. The simulation model aims in comparing ODMRP with MALMR and ODMRP. The performance examines of these two brings out against to the metrics displayed in the given table:

Number of nodes Range	50 to 200
Dimensions of space	1500 m × 300 m
Nominal radio range	250 m
Source–destination pairs	20
Source data pattern (each)	4 packets/second
Application data payload size	512 bytes/packet
Total application data load range	128 to 512 kbps
Raw physical link bandwidth	2 Mbps
Initial ROUTE REQUEST timeout	2 seconds
Maximum ROUTE REQUEST timeout	40 seconds
Cache size	32 routes
Cache replacement policy	FIFO
Hash length	80 bits
certificate life time	2 sec

Table1 : Simulation parameters that we considered for experiments

The metrics to verify the performance of the present protocol as follows:

- Data packet delivery ratio: Data packet delivery ratio is calculated as the ratio between the number of data packets that are delivering by the source and the number of data packets that are received by the sink.

- Packet Delivery Fraction: It is the ratio of data packets send to the destinations to those created by the sources. The PDF says about the performance of a protocol that how successfully the packets have been send. Higher the value produces the better results.
- Average End to End Delay: Average end-to-end delay is an average end-to-end delay of data packets. Buffering during route discovery latency, queuing at interface queue, retransmission delays at the MAC and transfer times, may cause this delay. Once the time difference between packets sent and received was recorded, dividing the total time difference over the total number of CBR packets received provided the average end-to-end delay for the received packets. Lower the end to end delay better is the performance of the protocol.
- Packet Loss: It is defined as the difference between the number of packets sent by the source and received by the sink. In our results we have calculated packet loss at network layer as well as MAC layer. The routing protocol forwards the packet to destination if a valid route is known; otherwise it is buffered until a route is available. There are two cases when a packet is dropped: the buffer is full when the packet needs to be buffered and the time exceeds the limit when packet has been buffered. Lower is the packet loss better is the performance of the protocol.
- Routing Overhead: Routing overhead is calculated at the MAC layer which is defined as the ratio of total number of routing packets to data packets.

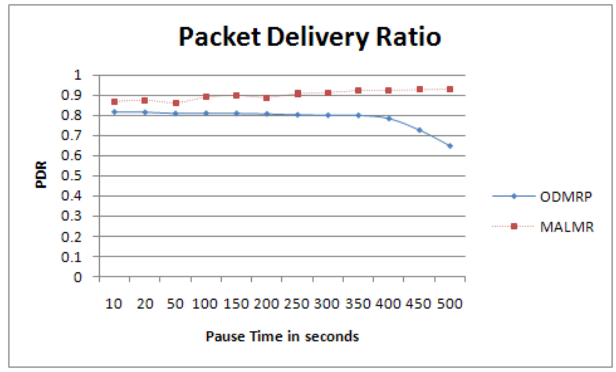
Figure 5(a) displays the Packet Delivery Ratio (PDR) for ODMRP [15] and ODMRP with MALMR. Depending on the results it is clear that MALMR reduces the loss of PDR that observed in ODMRP [15]. The approximate PDR loss recovered by MALMR over [15] is 14. 471%, this is an average of all pauses. The minimum individual recovery observed is 5. 91% and maximum is 30. 345%.

The packet delivery fraction (PDF) is denoted as:

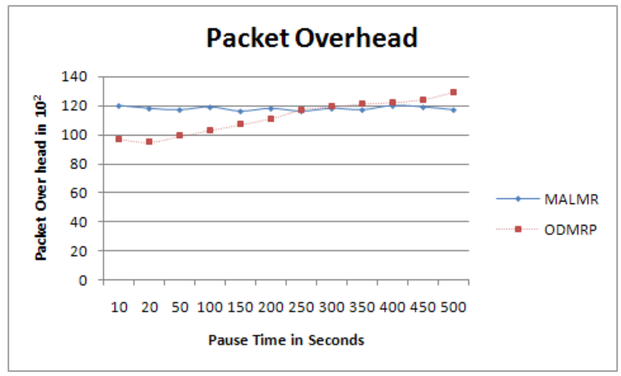
$$P' = \sum_{f=1}^{e} \frac{R_f}{N_f}$$

$$P = \frac{1}{c} * P'$$

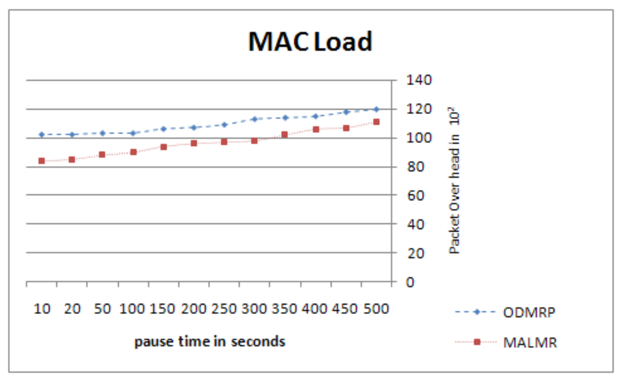
- P is the fraction of successfully delivered packets,
- c is the total number of flow or connections,
- f is the unique flow id serving as index,
- R_f is the count of packets received from flow f
- N_f is the count of packets transmitted to flow f .



(a) Packet delivery ratio



(b) Packet overhead comparison report



(c) Mac load comparison

Fig. 5 : Comparison of performance metric values between ODMRP with MALMR and ODMRP

Figure 5(b) affirms that ODMRP with MALMR have stable packet overhead over ODMRP [15] where the magnitude growth in packet overhead in different pause intervals. Because of congestion avoidance routing mechanism of MALMR this benefit is possible. The average Packet overhead observed for 12 intervals in ODMRP with MALMR is 117. 9 more than packet overhead observed for 12 intervals in ODMRP. But the average growth of the packet over head in ODMRP is 26. 36%, but in the case of ODMRP with MALMR, the average growth in packet over head is 3. 34%. This advantage of ODMRP with MALMR over ODMRP happens due to the collision and congestion avoidance strategy introduced in MALMR.

The advantage of ODMRP with MALMR over ODMRP in MAC load overhead control is shown in the Figure 5(c), . The average growth in MAC load over head in ODMRP with MALMR is 14. 32% is almost equal to MAC load overhead in ODMRP, which are 14. 17%, this resulted due to multicasting of the packets in MALMR to all target nodes unlike in ODMRP, a unicasting packet.

VI. Conclusion

This paper expatiate a MAC level multicast routing algorithm called "Medium Access Level Multicast Routing" i. e. MALMR. The present routing strategy aims at avoidance of congestion at group levels formed in multicast route discovery. This protocol derives an algorithm that transmits the data in multicast manner at group level unlike other multicast protocols, concentrating of data transmission in a sequence to every targeted node. Being independent, the MALMR works with group of either tree or mesh. The present mentioned MALMR is tested by associating with ODMRP where the simulation results indicated that the MALMR improves the PDR and reduces the Packet overhead of ODMRP in order of magnitude. It is further planned to develop an extension to MALMR which can even control the congestion besides avoiding congestion.

References Références Referencias

1. G. Krishna, "Routing protocols in mobile ad-hoc networks," pp. 1-30, 2006 http://www.cs.umu.se/education/examina/Rapporter/Krish naGorantala.pdf
2. L. Sung-Ju, S. William, G. Mario, "On-demand multicast routing protocol in multihop wireless mobile networks," Mobile Networks and Applications, 2001.
3. S. J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A performance comparison study of ad hoc wireless multicast protocols," Proceedings of the IEEE INFOCOM'00, 2000.
4. P. Madhan, J. James, K. Murugan, V. Ramachandran, "A Comparative and Performance Study of On Demand Multicast Routing Protocols for Ad Hoc Networks," 9th International Conference on High Performance Computing (HiPC), 2002.
5. V. Kumar, O. Katia and T. Gene, "Exploring mesh- and tree based multicast routing protocols for MANETs," IEEE Transactions on Mobile Computing, vol. 5, pp. 28–42, 2006
6. Prashant Dewan, Partha Dasgupta and Amiya Bhattacharya; "On Using Reputations in Ad hoc Networks to Counter Malicious Nodes" Tenth International Conference on Parallel and Distributed Systems, 2004; ICPADS 2004. Proceedings; July 2004
7. L. Butty´an and J.-P. Hubaux. Enforcing Service Availability in Mobile Adhoc WANs, In ACM international symposium on Mobile ad hoc networking and computing, pages 87–96, Boston, Massachusetts, 2000. ACM Press.
8. S. Zhong, J. Chen, and R. Yang Sprite: A Simple, Cheat-proof, Credit-based System for Mobile Ad-hoc Networks. In IEEE INFOCOM, San Francisco, USA, 2002. IEEE Press.
9. E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Ad-hoc Multicast Routing Protocol," Internet- Draft, draft-talpade-manet-amroute-00.txt, Aug. 1998, Work in progress.
10. C.W. Wu, Y.C. Tay, and C.-K. Toh, "Ad hoc Multicast Routing Protocol utilizing Increasing id-numberS (AMRIS) Funcational Specification," Internet-Draft, draft-ietf-manet-amris-spec-00.txt, Nov. 1998, Work in progress
11. J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, Aug. 1999, pp. 1380-1394.
12. E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," Proceedings of ACM/IEEE MOBICOM'99, Seattle, WA, Aug. 1999.
13. S.-J. Lee, M. Gerla, and C.-C Chiang, "On-Demand Multicast Routing Protocol," Proceedings of IEEE WCNC'99, New Orleans, LA, Sep. 1999, pp. 1298-1302.
14. S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," Proceedings of IEEE INFOCOM2000, Tel Aviv, Israel, Mar. 2000.
15. S.-J. Lee, W. Su, and M. Gerla, Internet Draft, draft-ietf-manet-odmrp-02.txt, Jan. 2000
16. J. Haartsen, M. Naghshineh, J. Inouye, O.J. Joeressen, and W. Allen, "Bluetooth: Vision, Goals, and Architecture," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 4, Oct. 1998, pp. 38-45.
17. C. E. Perkins and E. M. Royer, "Ad-Hoc On Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA, Feb. 1999.
18. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Pusblishers, 1996.
19. S. Keshav, "An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network", Addison-Wesley, Menlo Park, California, 1997.