SURVEY OF Ad-Hoc MULTICAST PROTOCOLS

SUBMITTED BY

HARKIRAT SINGH

ON DEC'7TH, 2000

ADVANCE NETWORKS

INSTRUCTOR – DR. SURESH SINGH

TABLE OF CONTENT

1. INTRODUCTION

Ad Hoc Networks (AHNs) are envisioned to have dynamic, sometimes rapidly changing, random, multihop topologies that are likely composed of relatively bandwidth-constrained wireless links. Individual nodes and even whole networks of nodes may continuously move, or disappear, leading to highly dynamic topologies. Multicast protocols for AHNs have to be designed keeping aforesaid characteristics of the nodes. Most of the Multicasting protocols are either tree based or mesh based. These protocols can be further divided into two categories as:

1.1 Proactive Protocols

Proactive protocols enforce maintenance of the network topology even if there is no traffic. Most of the proactive protocols are tree-based. It means that the shortest paths between different nodes form a tree-like structure, which is maintained in the nodes. Convergence is slow for topology changes but because of tree structure, the efficiency is very high. The other alternative is to maintain information about multiple links between nodes. Links form a mesh-like structure, though overhead increases but mesh-based solution provide more robustness.

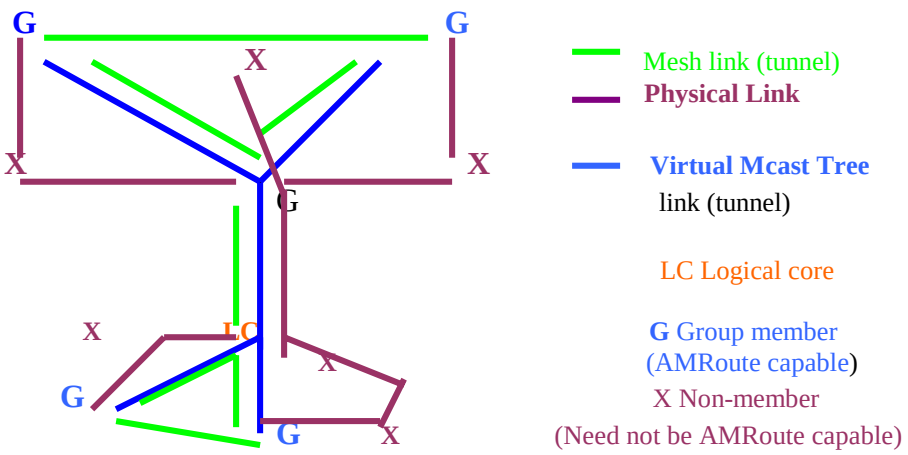
1.2 Reactive Protocols

Reactive protocols do not require the maintenance of the network topology when there is no traffic. The state information is acquired when needed (on-demand). It means that the reactive protocols have to maintain the state of the network, and rapid changes in the topology may cause problems.

2 Adhoc Multicast Routing Protocol (AMRoute)

2.1 Introduction

AMRoute was modeled by Bommaiah, McAuley, Liu and Talpade, a joint effort of Bellcore and University of Maryland. AMRoute is of proactive protocol category and motivated from Core based Trees (CBT) and PIM-SM Multicast routing protocols, however, the core is not the central point of data distribution, and it can change dynamically which makes it less vulnerable to link breakages. AMRoute is based on user-multicast trees and dynamic cores. Data is distributed using a bi-directional shared-tree where only the group senders and receivers can be tree nodes. Neighbors in the tree are connected using unicast tunneling through the intermediate routers (IP-in-IP tunnel). Thus, AMRoute need only be supported by nodes that want to be part of multicast group, which limits the maintenance of state by group members only. AMRoute is independent of specific unicast protocol, so it can operate transparently over separate domains that might use different routing protocols. The key characteristic of AMRoute is its usage of virtual mesh links to establish the multicast tree. Therefore, as long as routes between tree members exist via mesh links, the tree need not be readjusted when network topology changes.



2.2 Mesh Creation Phase

Initially, each group member declares itself as core for its own group of size one. Each core periodically floods JOIN_REQ <source IP, Multicast Group IP, message id, TTL> to discover other disjoint mesh segments of the group. When a member node receives a JOIN_REQ from a core of a same group but different mesh segment, it replies with a JOIN_ACK <source IP, Multicast Group IP, message id, TTL> and mark that node as a group neighbor. The node that receives a JOIN_ACK also marks the sender of the packet as its mesh neighbor, thus a new bi-directional tunnel is established between the core and the responding node of the other mesh. Due to mesh mergers, a mesh will have multiple cores. One of the cores will emerge as the "winning" core of the unified mesh due to the core resolution algorithm (node in the mesh become aware of existence of multiple cores during tree creation phase, one simple core resolution algorithm could be, pick the winning core to be the one with the highest IP address). Only logical core nodes initiate the discovery of other disjoint meshes, while both core and non-core nodes respond to the discovery messages, thus limit the bandwidth usage. However, to avoid the situation where every merger adds a link to a core (which might result in too many links from the core), non-core nodes can participate in the mergers by responding to discovery messages received from other cores.

If a node leaves a group, it sends out a single JOIN_NAK <source IP, Multicast Group IP, message id, TTL> message to its neighboring nodes. If it subsequently receives any data or signaling message for that group, it can again send out further JOIN_NAK messages.

2.3 Tree Creation Phase

While the mesh establishes connectivity across all the members of a group, a tree is needed for forwarding a data. The core is responsible for initiating the tree creation process by sending TREE_CREATE <source IP, Multicast Group IP, message id, Seq.#> messages along all the links incident on it in the mesh, (TREE_CREATE message is sent along the unicast tunnels in the mesh and are processed by the group members only, while JOIN_REQ messages are broadcast messages that are processed by all network nodes). Based on mobility of the mesh nodes the subsequent period between two

TREE_CREATE messages can be increased or decreased. So newer and more optimal trees might be created when TREE_CREATE messages are sent out. Group members receiving non-duplicate TREE_CREATEs forward it on all mesh links except the incoming, and mark the incoming and outgoing links as tree links.

2.4 Maintenance of Tree and Mesh

If a link is not going to be used as part of the tree, the TREE_CREATE message is discarded and a TREE_CREATE_NAK is sent back along incoming links. On receiving a TREE_CREATE_NAK < source IP, Multicast Group IP, message id, Seq. #>, a group member marks the incoming link as a mesh link and not a tree link. Thus, each non-core node considers the link (note: all tree links are bi-directional tunnels) along which a non-duplicate TREE CREATE message was received and every other link along which no TREE_CREATE_NAK message was received to be part of the tree for a specific group, though the core considers every incident link to be a part of the tree.

Situation can arise when a failure of node can lead to partition of the mesh and this way only one of the mesh will be having the core (logical core). Each node in the mesh expects to periodically receive TREE_CREATE message, in case message is not received within a specified period, the designate itself to be the core after a random time. The node whose timer expires earliest succeeds in becoming the core and initiates the process of discovering other disjoint meshes as well as tree creation. Multiple cores that may arise in this case are resolved by the core resolution procedure.

2.5 Analysis

Data packets are forwarded through the tree nodes (which are AMRoute supported) and all the other non-member nodes are not involved. This may lead to delay, as the chances are that optimal route can be achieved using non-member nodes. The worst case overhead occurs in case of a simple star network with a non-member central node directly connecting n member nodes: a tree using the central node will take n-hops, while a user multicast tree will take 2*(n-1) hops.

In case some of the node members of the tree are mobile then reconstruction of the tree will take place and it may lead to loops and creation of the sub-optimal tree, as some of the nodes will be forwarding data according to the stale tree and others according to newly built tree. The existence of loop may lead to collision, congestion, and retransmission of packets.

The loss of control packets (TREE_CREATE, JOIN_ACK, etc.) leads to segmentation of tree and loss of data therefore. AMRoute has assumption of equal reachability of both ends of bi-directional link nodes, however, when mobility speed increases the chances are that a node can reach a neighboring node, but not necessarily vice versa.

During tree creation phase a mesh-branch will be picked up based on receipt of first TREE_CREATE message and any duplicate TREE_CREATE message will be discarded (based on sequence number). This process does not ensure tree with most bandwidth efficient (e.g., using minimum number of total hops) or lowest latency.

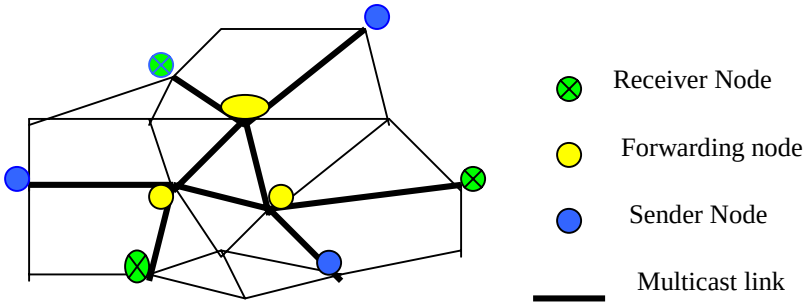
3 On Demand Multicast in Mobile Networks (ODMRP)

3.1 Introduction

ODMRP is of reactive protocol category, modeled by Mario Gerla, Pei and Lee at UCLA. It creates a mesh of nodes which forward multicast packets via flooding, thus provide path redundancy. It uses a soft state approach in-group maintenance (i.e. sender/ receiver can leave the group without informing anyone). Similar to on-demand unicast routing protocols, a request phase and a reply phase comprise the protocol. It does not maintain the state of the network but uses on-demand procedures to dynamically build routes and update the multicast group membership. ODMRP can coexist with any unicast routing protocols such as DSR, ZRP, TORA, AODV and ABR. It can also operate very efficiently as unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol.

3.2 Forwarding Group

ODMRP uses a concept of "Forwarding Group (FG)" instead of traditional tree infrastructure. FG is a set of nodes, which are responsible for forwarding (broadcast) multicast packet of G. All the neighbors can hear it, but only the neighbors who are in the FG will first determine if it is not a duplicate and then broadcast it in turn. This scheme can be viewed as "limited scope" flooding.



The key features of FG infrastructure are reduced storage overhead, allows loose connectivity among multicast group members, which makes protocol more scalable for large networks and more stable for mobile wireless networks.

3.3 Forwarding Group Setup

Forwarding Group setup is primarily based on two actions:

3.3.1 Join Request

When a multicast source has data to send, it periodically floods a member advertising packet. This packet called a "Join Request" packet <Source IP, Multicast Group IP, Previous hop IP, Seq. #, TTL, Hops>. This periodic transmission refreshes the membership information and updates the route (to take care of node movement). When a node receives a Join Request packet, it updates the entries of "Message Cache"<Multicast

Group IP, Source IP, Seq. #, Previous Hop IP>. When node receives a Join Request Packet, the processing is done as:

- If it is a duplicate by comparing the (source IP and Seq. #) in Message cache, if so then discard the packet.
- If it is not a duplicate then insert an entry into Message cache <Multicast Group IP, Source IP, Seq. #, Previous Hop IP>.
- Increase the hop count and if hop count does not exceed the TTL value, then set the node's IP address into Last Hop IP filed and relayed.
- If the node is a receiver member of the multicast group, then insert/update the information into the Member table (for each multicast group a receiver is participating maintain <Multicast Source IP, Time_Stamp> to detect expired source, if no Join Request is received from a source in the Member Table within MEM_TIMEOUT, it is removed from member Table) and originate a "Join Table".

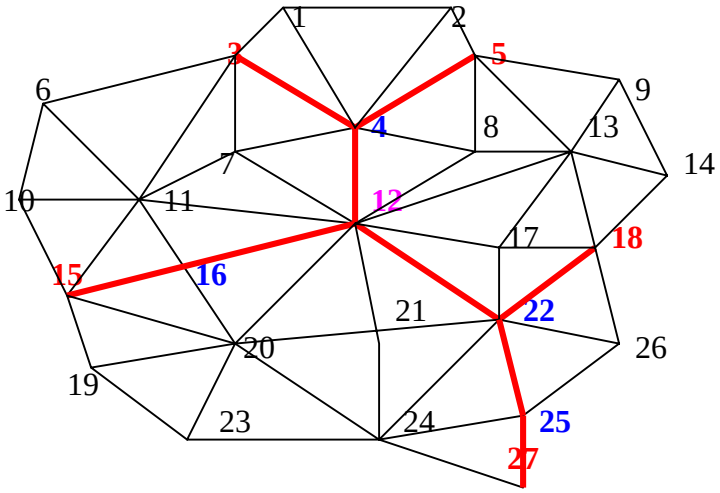
3.3.2 Join Table

A multicast receiver broadcasts a "Join Table" packet <Multicast Group IP, Sender IP, Next Hop IP> every Join_Table_REFRESH as long as there are any entries in its Member Table. The transmission of Join table is triggered by the update of its Member Table.

When a node receives a Join table, it checks if the next node ID is its own ID. If it does, the node realizes that it is on the path to the source thus is part of the forwarding group; it sets the FORWARDING_GROUP_FLAG and broadcasts it's own Join Table built upon matched entries. The next node ID is set to the previous node ID field of the Join Request packet just received (which can be obtained from message cache). This way, each forward group member propagates the join table until it reaches the multicast source via the shortest delay path. This process sets up (or updates) the route from source to each receiver and builds a mesh of nodes, the forwarding group.

3.4 Maintenance

No explicit control packet needs to be sent to leave the group. If a multicast source wants to leave the group, it simply stops sending any Join Request packets since it doesn't have any multicast data to send to the group.



Sender = {12} Join_Table (at node 27) = {sender, next hop}
Receivers = {3,5,15,18,27} = {12, 25}
FG = {4,16,22,25}

If a receiver no longer wants to receive from a particular multicast group, it removes the corresponding entries from its Member Table and does not forward the Join Table for that group.

3.5 Data Packet Flow

Multicast sources broadcast the data packets, when receiving the multicast data packet, a node forwards it only when it is not a duplicate and the setting of FORWARD_GROUP_FLAG has not expired. Thus minimizes the traffic overhead and prevents sending packets through stale routes.

3.6 Analysis

ODMRP builds per-source meshes. If the number of senders increases, more JOIN_PACKET are propagated and control packet overhead grows accordingly. Due to this fact ODMRP suffers from poor scalability and may not be as efficient if network has large number of nodes as multicast senders (more than 100).

In networks where GPS is available, ODMRP can be made adaptive to node movements by utilizing mobility prediction. By using location and mobility information supported by GPS, route expiration time can be estimated and receivers can select the path that will remain valid for longest time. With the mobility prediction method, sources can reconstruct routes in anticipation of route breaks and thus make protocol more resilient. Because ODMRP multiple redundant routes for the data delivery chances are that more data packets will be transmitted (for single packet delivery) than tree based protocols.

4 Ad Hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)

4.1 Introduction

AMRIS is modeled at National university of Singapore and Georgia Tech., it is of proactive protocol category. The key idea is that each participant in the multicast session has a session specific multicast session member id (msm-id), which provides each node with a heuristic height in the multicast delivery tree. AMRIS constructs a shared delivery tree (DAG) to support multiple senders and receivers within a multicast session, rooted at special node, Sid (generally it's the source node or among the source nodes that has the minimum msm-id among all nodes). AMRIS is independent of underlying unicast routing protocol. Maintenance of broken links is done locally without need of any central authority thus reducing overhead of node mobility.

4.2 Tree Initialization

Initially, a special node called Sid broadcast a NEW_SESSION packet <msm-id, multicast session id (e.g. class D IP), routing tables>, all the nodes that receive the NEW_SESSION message generate their own msm-id by computing a value that is larger and not consecutive, so that there are gaps between the msm-ids of a sender and receiver; these gaps are useful for quick local repair of the delivery tree. A node may receive multiple NEW_SESSION (NS) packets to avoid this a jitter is introduces between the receipt and subsequent rebroadcast of the NS packet. A receiver before broadcasting the NS packet will replace msm-id field with its own msm-id and routing metrics (computed based upon receipt of multiple NS packets). The NS message thus travels outward from the Sid in an expanding ring fashion and eventually every node will have assigned to themselves a msm-id (msm-id of the nodes which are not interested in multicast session will be returned to the msm-id pool for future use).

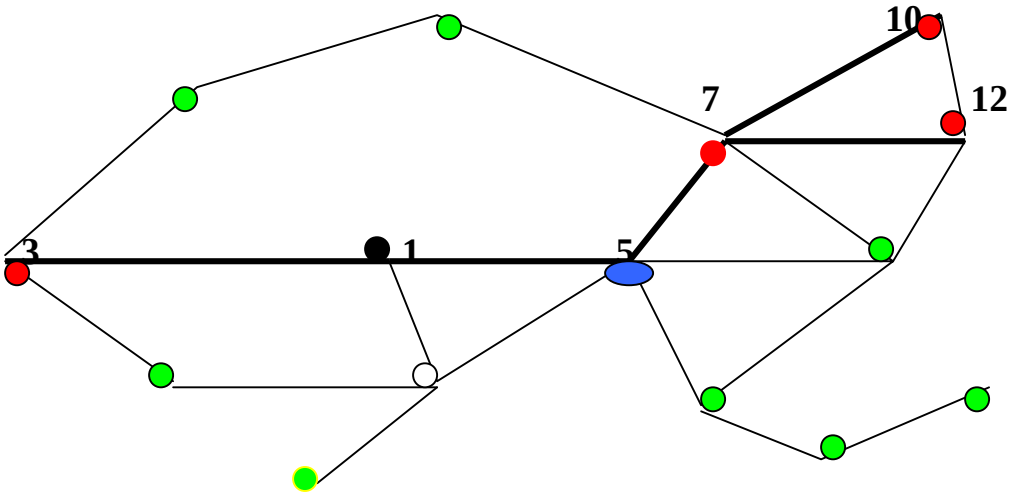
AMRIS maintains a Neighbor-Status table <Neighbor unique-id (IP), msm-id, relation (parent/child), Remaining timeout value, routing metric>, Neighbor-Status table is updated by the contents of NEW-SESSION packets. Each node sends a periodic beacon <Node unique-id (IP), msm-id and status (member/non-member), Parent msm-id (who's msm-id is less than node's own msm-id), Child msm-id (who's msm-id is more than node's own msm-id)> to signal their presence to neighboring nodes.

A node X then joins the session by first determining from the NEW_SESSION and beacon messages received which neighboring nodes have smaller msm-ids then X (these could be set of potential parents), a unicast JOIN_REQ is sent to one of the potential parent nodes. When the potential parent Y receives a unicast JOIN_REQ, it checks if Y itself is already a part of multicast session. If so, Y will send a JOIN_ACK otherwise Y will try to locate a potential parent by sending JOIN_REQ_PASSIVE to its potential parent. If a node fails to receive a JOIN_ACK or receives a JOIN_NAK after sending a JOIN_REQ, it performs "Branch Reconstruction (BR)".

The key feature is that if immediate neighboring nodes are already on the multicast tree, then '1-hop' approach is very fast and efficient and BR is broadcast approach, which is used if '1-hop' approach fails.

4.3 Tree Maintenance

This mechanism operates continuously in the background to ensure that a node remains connected to the multicast session tree. When a link between two nodes breaks, the node




Non-participant Node


I-node (interested node in multicasting session)  Sid


U-node (node not interested in Mcast session, but forced to join the sessions because required to relay/intermediate node on the multicast delivery path)

with the larger msm-id (child) is responsible for rejoining. In case the parent is 1-hop away and is on the multicast Tree then it is simple (as said before), otherwise BR approach is adopted.

A node X will broadcast JOIN_REQ with R hops field (to made broadcast localize), when node Y receives a broadcasted JOIN_REQ, it checks if it can satisfy the request, if so, Y sends a JOIN_ACK on the reverse path set up back to X. However, Y does not forward multicast traffic to X till Y receives a JOIN_CONF from X (X may receive multiple ACKs).

If a node does not have a valid msm-id and wishes to join, it first uses neighboring msm-ids to compute an msm-id for itself, then execute BR routine to join the session.

4.4 Data Flow

Data forwarding is done by the nodes in the tree. Only the packets from the registered parent or registered child are forwarded. Hence, if the tree link breaks, the packets are lost until the tree is reconfigured.

4.5 Analysis

BR search is three way hand-shake kind of process and the parent will not forward the multicast packet to the desired child till the child sends a JOIN_CONF packet and in case JOIN_CONF is lost the tree will remain broken till next BR search with increased hop-count (R+1). Potential assumption that multicast application are long lived.

Nodes send beacons every second, and neighbors are considered to have moved away if 3 consecutive beacons are not received. Thus, in the best case, it takes 3 seconds after the link break for AMRIS to start tree readjustment. A number of packets can be lost during this period.

5 Core-Assisted Mesh Protocol (CAMP)

5.1 Introduction

CAMP is modeled at UC Santa Cruz, its proactive protocol category. CAMP creates a

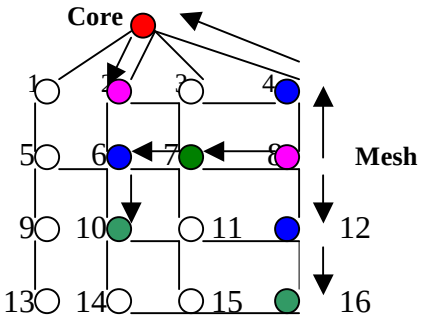
multicast mesh which is a subset of the network topology that provides at least one path from each source to each receiver in the multicast group, which is shortest path. Cores are used to limit the control traffic flow needed for receivers to rejoin the multicast group / maintenance of multicast group. CAMP relies on an underlying unicast protocol which guarantees correct distances to all destinations within finite time.

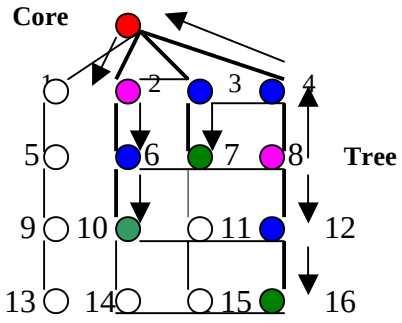
5.2 Mesh creation and maintenance

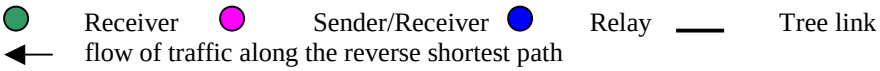
CAMP supports multicasting by creating a shared mesh structure. All nodes in the network maintain a set of tables with membership and routing information. More-over, all member nodes maintain a set of caches that contain previously seen data packet information and unacknowledged membership requests. CAMP classifies nodes in the network as duplex or simplex members, or non-members. Duplex members are full members of the multicast mesh, while simplex members are used to create one-way connections between sender-only nodes and the rest of the multicast mesh. “Cores” are used to limit the flow of JOIN_REQUEST packets.

A node wishing to join a multicast mesh first consults a table to determine whether it has neighbors which are already members of the mesh. If so, the node announces its membership via a CAMP UPDATE. Otherwise, the node either propagates a JOIN_REQUEST towards one of the multicast group “cores,” or attempts to reach a member router by an expanding ring search of broadcast requests. Any duplex member of the node can respond with a JOIN_ACK, which is propagated to the source of the request.

Periodically, a receiver node reviews its packet cache in order to determine whether it is receiving data packets from those neighbors which are on the reverse shortest path to the source. If not, the node sends either a HEARBEAT or a PUS JOIN message towards the source along the reverse shortest path. This process ensures that the mesh contains all such reverse shortest paths from all receivers to all senders. The nodes also periodically choose and refresh their selected “anchors”(in given fig router 6 uses router 7 as an anchor for the group) to the multicast mesh by broadcasting updates. These anchors are neighbor nodes, which are required to re-broadcast any non-duplicate data packets they receive. A node is allowed to discontinue anchoring neighbor nodes which are not refreshing their connections. It can then leave the multicast mesh if it is not interested in the Multicast session and is not required as anchor for any neighboring nodes.







5.3 Analysis

CAMP can work with certain unicast protocols only (developers of this protocol prefers WRP), routing protocols that based on the Bellman-Ford algorithm ca not be used with CAMP. In CAMP paths to the distant destination have fewer redundant paths than those closer to the center of the mesh, during mobility they are more prone to occasional link breaks preventing a vital “anchoring” node from successfully receiving packets. Thus more packet retransmissions are required for packets destined to edges than to the center.

CAMP would be having larger control overhead during high mobility than other mesh oriented protocol like ODMRP because of its reliance on the unicast routing protocol WRP, which sends triggered updates. WRP suffers from exponential growth in control traffic overhead under increasing mobility. Moreover, CAMP piggybacks its own update messages onto WRP updates and those packets are responsible for overhead growth.

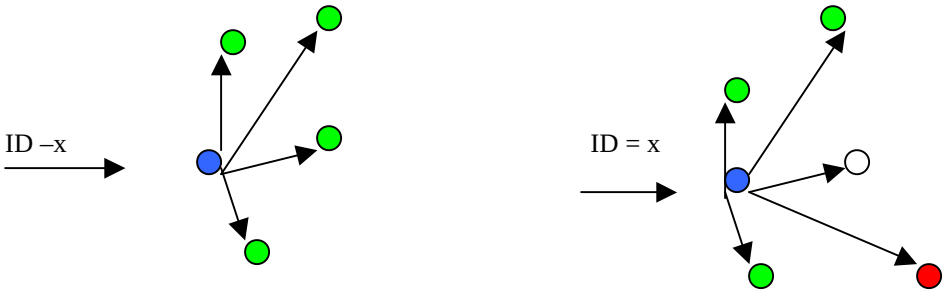
6 Integrated Multicast for Ad Hoc Networks (IMAHN)-Flooding

6.1 Introduction

This protocol is modeled at USC. The researchers of this protocol believe that existing multicasting protocols (some of them discussed above) are state based means routers (nodes in Adhoc networks) will maintain a state. Furthermore, in many Ad hoc Networks (AHNs) like hand-held devices storage capacity and power is a critical issue and limits the storage and processing of states of highly dynamic behavior of AHN where hosts behavior is completely independent, highly mobile (in terms of speed and direction of movement) and high probability of frequent partition. In AHNs specialized flooding can be potential solution, with key features of reliable delivery and minimal state retention, as nodes have to store only their own state (e.g. which packets have been received before). It behaves very well in highly dynamic AHNs.

6.2 Hyper Flooding

Hyper-flooding is not completely state free, each host has to keep a track of its current neighbors (though this information may be borrowed from underneath MAC/Network layer protocol). A multicast packet <Multicast Group id, Unique id (computed as a function of least source address, time, group ID, Seq. #), hops, TTL, Time-stamp> is broadcasted by the sender and all of its receiver will further broadcast it. Each node maintains a cache of Ids for recently processed multicast packets so to avoid broadcast storm. However, when a host receives a previously seen multicast and, in the meantime, some new neighbors have been acquired, then a re-broadcast is required with the new-neighbor id to avoid broadcast storm.



Time T1 Time T2

[At time T2 a copy of same packet x arrives, and in the mean time host (blue) has got new neighbor (red) so if host does the rebroadcast then other nodes (green) will have unnecessary broadcast (at least four), to avoid this superfluous broadcast the host can inform that it is re-broadcast for new neighbor (red)]

A two-tier approach may be suitable for large, widely distributed AHN where AHNs are inherently hierarchical (e.g. Military). In such scenario the AHN is partitioned into clusters and each cluster is assigned a cluster head, intra-cluster (tier-1) multicast is provided with hyper-flooding and inter-cluster (tier-2) can be accomplished by unicasting among cluster heads.

6.3 Analysis

Hyper flooding works well in small highly dynamic AHN however shows poor results in large networks for which it has to adopt to different multicasting protocol. In case the AHN is stable Hyper-Flooding can show poor results in terms of bandwidth and efficiency if compared with any tree based multicasting protocol like ODMRP. Hyper-flooding does not ensure the reliability (due to limited re-broadcast) of the delivery of the packet, hence it is not suitable for AHNs as battlefield where messages are not too frequent but their reliability is more important.

7 Summary

IETF MANET working group has suggested some of the metric for the evaluation of routing / multicasting protocols for AHNs [7]. Some of these metrics are:

Packet deliver ratio: The ratio of the number of data packets actually delivered to the destinations versus the number of the data packets supposed to be received. These numbers presents the effectiveness of a protocol.

Number of control bytes transmitted per data bytes delivered: this measure tells the control overhead in the delivery of the packet, it also shows the protocol's channel access efficiency, which is very important in ad hoc networks since link layer protocols are typically contention oriented.

Other metrics are security, effect of mobility, effect of topology change, sleep etc.

Protocols	AMRoute	ODMRP	AMRIS	CAMP	Flood
Category	Proactive	Reactive	Proactive	Proactive	-
Confoguration	Tree	Mesh	Tree	Mesh	Mesh
Loop-free	No	Yes	Yes	Yes	Yes
Dependency on Unicast Protocol	Yes	No	No	Yes	No
Periodic Messaging	Yes	Yes	Yes	Yes	No
Control Pkt Flood	Yes	Yes	Yes	No	No Control Pkt
Salient feature	IP-in-IP tunnel	Per source mesh	Shared DAG	Shared mesh	Shared mesh
Assumptions	Existence of Unicast routing protocol	NO	Multicast apps are long lived	Prefers to work with WRP (uni-cast routing p'col)	Target small dynamic AHNs
Able to handle node mobility	No (creates temporary loops)	Yes	May loose some packets	control packet overhead grows	Yes
Works well when?	Stable AHN	less mulicast senders (<100)	Stable AHN	Performance increases in AHN with >100 senders	small dynamic AHN

7.1 Packet Delivery Ratio

During mobility of nodes the chances are that ODMRP would not effect the packet delivery ratio due to the fact that it provides redundant routes with the mesh topology, however, AMRoute will suffer badly because it leads to generation of loops during tree re-construction phase, so due to congestion / collision packet delivery ratio will be degraded. In CAMP if the group core is part of the temporally unreachable nodes then multicast routing updates regarding mesh maintenance will be postponed which may affect the packet delivery ratio in negative side. In AMRIS ideal period between two consecutive beacons (say t) is very important and in case this period is reduced then there would be too many beacons and beacon processing overhead and in case t is increased then during tree readjustment phase neighbors can miss beacons which may cause of loss of multicast packets and may effect the packet delivery ratio. Flood is targeted for highly dynamic networks.

7.2 Energy depletion in AHN with limited senders

In stable networks it is beneficial to use ODMRP from point of view of energy consumption / node as all the nodes will be depleting their energy resources at almost same rate, however, in other protocols as some of the nodes have to do more processing than other nodes that may cause network partition much faster.

7.3 Security

None of the protocols talk about security, most of the protocols give an example of Military as AHNs and it is essential for the Military that security is not compromised in any case. Flooding is definitely not a best solution for such AHNs.

7.4 Number of control bytes transmitted per data bytes delivered vs. mobility

ODMRP will show constant control overhead due to aforesaid facts that redundant paths exist and no need to find out new routes so no need to send more control packets.

CAMP would be having larger control overhead during high mobility because of its reliance on the unicast routing protocol WRP, which sends triggered updates. WRP suffers from exponential growth in control traffic overhead under increasing mobility. Moreover, CAMP piggybacks its own update messages onto WRP updates and those packets are responsible for overhead growth.

To overcome loops AMRoute will show increase in control overheads. AMRIS will also show increased control overheads on account of beacons and BR search.

7.5 Multiple multicast senders

Flooding would not be effective in multiple multicast senders scenario as there would be more collision, congestion and channel contention.

ODMRP performance will also degrade in multiple senders scenario (> 100) as more JOIN_REQUEST and bigger Member table and Message cache (suffer from scalability).

CAMP may show better performance due to increase in the number of anchors that each node requires. Each member node requests every neighbor which is in the reverse shortest path to some source, to rebroadcast multicast update packets it receives initially. Hence increasing the number of sources increases the redundant paths in the mesh.

In very large stable AHN CAMP should be the best choice to use as mesh becomes massive and more anchors are available to a node so more redundant paths are available.

8 Conclusions

Multicast solution for AHNs is different from fixed network due to inherent demanding features of AHNs, under such situation non of the existing multicast protocols is a bullet proof solution, as aforesaid facts show that a protocol works efficiently under one particular condition and as given condition changes (mobility of nodes, joining of more sender and receivers) protocol's efficiency degrades drastically.

One of the solution could be that if nodes participating in AHN dynamically change from one multicast protocol to another based upon the current condition then the best results can be achieved. But again the question is with limited storage and power capability (hand-held devices) how this can be achieved?

9 References

[1] E.Bommaiah, M. Liu, A. McAuley, and R. Talpade, “AMRoute: Ad-hoc Multicast Routing Protocol,” Internet-Draft, draft-talpade-manet-armroute-00.txt., Aug. 1998, Work in progress.
[2] S.-J.Lee, W.Su, and M.Gerla, “On-Demand Multicast Routing Protocols(ODMRP) for Ad Hoc Networks,” Internet-Draft, draft-ietf-manet-odmrp-01.txt, Jun. 1999, Work in progress.
[3] J.J. Garcia-Luna-Aceves and E.L. Madruga, “The Core-Assisted Mesh Protocol,” IEEE Journal on selected areas in Communication, vol. 17, no.8, Aug. 1999, pp. 1380-1394.
[4] J.J. Garcia-Luna-Aceves and E.L. Madruga, “A Multicast Routing Protocol for Ad-Hoc Networks,” In Proceeding of IEEE INFOCOM’99, New York, NY, Mar. 1999, pp. 784-792.
[5] C.W.Wu, Y.C. Tay, and C.-K.Toh, “Ad Hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional specification,” Internet-Draft, draft-ietf-manet-amris-spec-00.txt, Nov 1998, work in progress.
[6] S.Deering, “Host Extensions for IP Multicasting, RFC 1112”.
[7] M.S. Corson and J. Maker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” RFC 2501, Internet Engineering Task Force, Jan. 1999.
[8] Ho C. & Obraczka, K. & Tsudik, G. & Viswanath K., “Flooding for Reliable Multicast in Multi-Hop Ad Hoc Networks.”
[9] Obraczka, K. & Tsudik, G., Multicast Routing Issues in Ad Hoc Networks, IEEE International Conference on Universal Personal Communication (ICUPC '98), October 1998