BUILDING HYBRID MULTICAST BY COMBINING IP AND APPLICATION LAYERS

A Thesis Presented to the Faculty of the Graduate School

University of Missouri – Columbia

In Partial Fulfillment of the Requirements for the Degree

Master of Science

By ALEKSANDRE LOBZHANIDZE

Dr. Wenjun Zeng, Thesis Supervisor

DECEMBER 2007

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled.

BUILDING HYBRID MULTICAST BY COMBINING IP AND APPLICATION LAYERS

Presented by

Aleksandre Givi Lobzhanidze

A candidate for the degree of

Master of Science

And hereby certify that in their opinion it is worthy of acceptance

_____

Dr. Wenjun Zeng

_____

Dr. Yi Shang

_____

Dr. Zhihai He

ACKNOWLEDGEMENTS

To my family, who has given me all the best and enough support to make it through the exchange program and Graduate School.

I would especially like to acknowledge the assistance of my advisor Dr. Wenjun Zeng, who provided me with thoughtful insights, interesting challenges, and wonderful working environment. His guidance and patience have led me to the right direction of identifying and describing the issues that my thesis deals with, his criticism and insight have helped me overcome lots of problems; without his valuable knowledge, time and help my thesis would hardly be done.

I also want to thank Dr. Yi Shang and Dr. Zhihai He for taking time from their extremely busy schedules to serve on my thesis committee and contribute valuable insights.

Lastly, I want to thank all my friends at University of Missouri – Columbia. Two years at graduate school was a great experience and thanks to them I have many memorable moments to take with; they will last my whole life.

TABLE OF CONTENT

Chapter

List of Illustrations

Figure **Page**

Building Hybrid Multicast by Combining IP and Application Layers

Aleksandre Givi Lobzhanidze
Dr. Wenjun Zeng, Thesis Advisor

ABSTRACT

Nowadays, Internet TV (IPTV) is becoming a hot topic and streaming media over the network remains one of the most challenging issues for Internet Service Providers (ISP). Due to the increased number of subscribers for IPTV and other Internet broadcast applications, efficient multicast protocols become more important than ever. Existing multicast protocols have their advantages and disadvantages. IP Multicast is very efficient for multimedia data distribution, but it has deployment issues; most of ISPs do not support IP Multicast. Application Layer Multicast (ALM) is an alternative for content owners, which aims to eliminate the dependency on IP-Multicast enabled routers, but has performance issues. The main drawbacks of ALM are degradation of efficiency (one-to-many delivery function is achieved with multiple unicast calls) and instability (due to the dependency of the distribution trees on end hosts).

In this thesis, we propose a strategy that takes advantage of both existing techniques and maximizes the efficiency of resource usage. Our work explores how to build Hybrid Multicast (HM) by combining IP and Application layers. Our goal is to use available resources as much as possible, and provide a reliable solution for multicast applications.

HM is a software solution that implements the integration between ALM and IP Multicast. The proposed solution always tries to use IP Multicast capability wherever available; each computer on the hybrid network acts as an agent and is ready to extend the network to those nodes that do not have access to IP Multicast. In Hybrid Multicast, the path from Server to End-Node is built with minimum delay algorithm; HM assigns stability coefficient to each node, and based on bandwidth, delay, processing speed and loss rate, gives priority to those nodes that represent themselves more stable on the hybrid network. Unlike most existing multicast protocols that do not consider node failure in advance, HM adopts a proactive approach in which each node on the network has alternate parents which allow it to switch to backup route quickly and smoothly. HM also provides backup routes for IP Multicast enabled nodes. In addition, HM takes advantage of passive members, which are nodes that run hybrid multicast software, but have not joined any group yet.

Simulation results show, that HM performs much better than other multicast protocols. HM distributes the data with smaller delay than ALM, and the recovery time for backup routes in HM is smaller than other multicast protocols. The advantage is generally more significant when the multicast group size is large.

1. Introduction

The research done and innovations introduced in this thesis are tightly related to media streaming. The purpose of introducing a new protocol called Hybrid Multicast is intended to serve as a Multicast protocol for real-time media delivery. Applications like video conferencing need strong software or hardware support. The dilemma with the hardware is that it's expensive, while software is much flexible to install and relatively cheap. Existing transport protocols like TCP provide reliable delivery of data, but do not meet on-time delivery requirement. Hybrid Multicast mainly focuses on time delay and reliable, quality delivery of video data. However Hybrid Multicast can be used with the other types of data as well.

The word 'hybrid' speaks for itself, meaning that we have combination of different layers. IP Multicast and Application layer multicast is out there and people use it, so we don't necessarily have to invent a new wheel. We use both of these multicasting protocols and all we have to do is to make these two different layers talk to each other and accurately exchange information.

One good question before we describe anything about Hybrid Multicast, is why do we need it at all? Or why do we need multicasting at all? There are number of reasons why multicasting is desired, even though we have unicasting.

Until most internet service providers become multicast enabled, we can come up with something new that allows us to use features of multicast without updating hardware,

without paying too much money for the service and with easy updates of software application.

1.1 Hybrid Multicast

In this section we are going to overview Hybrid Multicast (HM), then we present important technical details, e.g. what are the key points in HM, how Join and Leave messages work, retransmission and reliability issues. HM is a bit complicated, we will try to consider all the problems encountered by other Multicast protocols. HM may not be a perfect solution and some aspects could be improved, but it has all features to be the most reliable, cost effective and easy-to-use multicast protocol. Hybrid Multicast is implemented at the application layer; therefore we don’t have hardware deployment issues. HM is an algorithm that uses two existing protocols and implements transition part between these two.

1.2 Overview of Hybrid Multicast

Hybrid Multicast is designed to overcome IP Multicast deployment issues. The basic idea is to combine IP multicast enabled network segments with the application layer multicast (ALM). We dynamically map ALM path to the underlying IP multicast path wherever it is available to optimize the performance. So basically Hybrid Multicast addresses the interoperability of different multicast protocols, depending on what is available at a given segment. IP Multicast performs much better than the application

layer multicast, so we take advantage of IP Multicast whenever available and hybrid multicast interacts two different protocols using hybrid technique.

So far we've learned that Application layer is a better approach, even though the implementation is complicated. The software itself has to be intelligent enough, to handle all kinds of scenarios. The design of Hybrid multicast would be as follows: we have three key platforms: Content Server, Rendezvous Point, and End-Nodes. End-Nodes have their own classification: Active/Passive End-Nodes; and End-Nodes are also broken into Parent and Children nodes (we'll see later how that works). Each of these Nodes has its own task and RP has to tell them what they are. It is important that all the nodes have information about Hybrid Multicast tree; basically RP collects information from all nodes and distributes this information among group members when necessary.

1.3 Why Do We Need Hybrid Multicast?

There are several reasons why we would use Hybrid Multicast

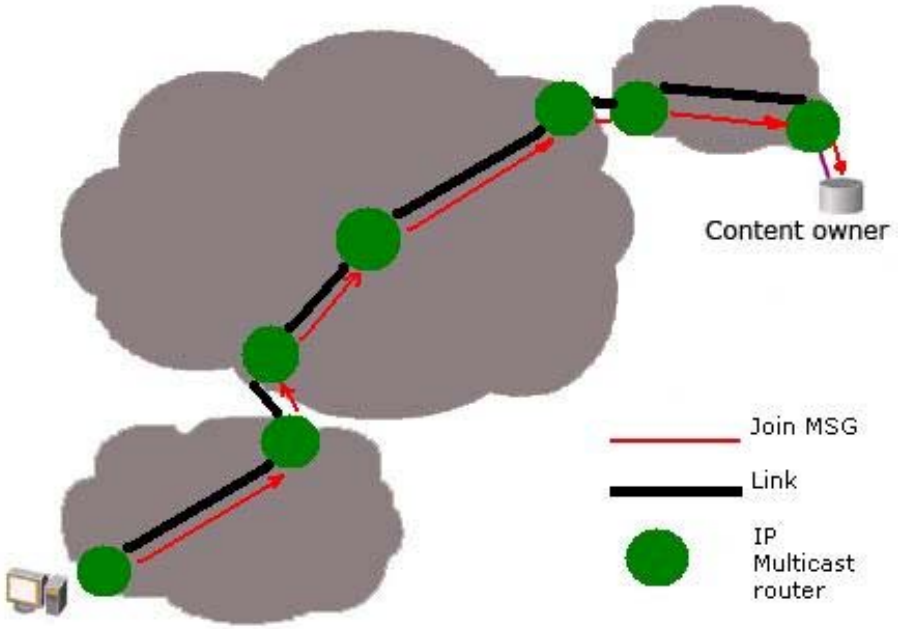
First and an important one is the cost of unicast versus multicast, saying in other words barriers for carriers entering video market.

Transit service providers have a challenge in the Internet where many local service providers are not multicast enabled. We can't enforce content owners to transmit

video and other multicast traffic across their backbone; the cost for the content owners may be prohibitively high if they have to pay for unicast streams for the majority of their subscribers.

The video transport market has great potential, but the barrier to enter is the fact that transit service providers (who do not typically have end users) can only serve new customers (subscribers) that are attached to IP Multicast enabled service providers. Most of Internet Service Providers do not support multicast – they are only running unicast. Multicast however is “all-or-nothing” solution. We have to turn it on everywhere in order to function. Sometimes because of security concerns systems administrators turn it off. Security issue here goes beyond the scope of this thesis, so we’re not going to discuss that.

Figure 1-1. Join in IP Multicast



In Figure 1-1 the multicast enabled service providers offer a multicast service for content owner. A user in the multicast enabled local provider wants to receive content and subscribes to the content. As we know multicast enabled transit provider can efficiently transport the content to the multicast enabled local provider, which sends it to the user(s).

However if a user is in the unicast only network, it will not be able to join this way. In this scenario, the content owner can allow the user to join via unicast – but the content owner will incur an added cost, since it will need extra bandwidth to support the unicast subscribers.

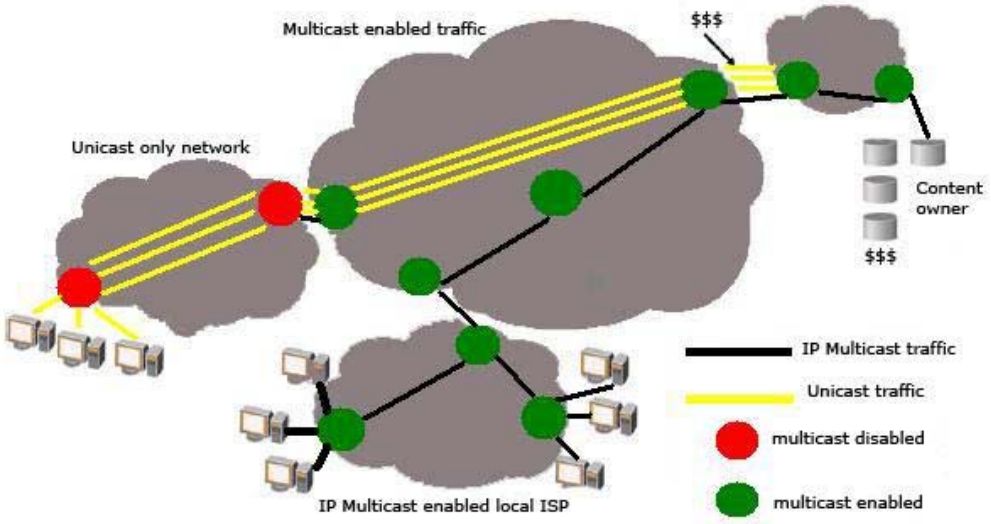
Motivated to maximize its subscribers, the content owner has unicast fallback streams. But as more users try to join from unicast-only networks, the content owner has to spend more money; more money on bandwidth and also on caching servers and splitters to support these users.

Providing broadcast video via unicast will cause scalability problems. Content providers who try to do this incur a significant cost per user. The use of caches only serves to distribute the problem, and the cost to distribute the content – per audience member is just too high to charge.

There's plenty of potential of more subscribers, especially as last mile bandwidth is increased, but that doesn't help the content owners, because their costs scale linearly as new unicast subscribers are added.

With these linear cost increases, content owner frequently can't afford to support their own subscriber base. The problem with multicast across multiple service providers is illustrated in figure 1-2.

Figure 1-2. Multiple Unicast Transmissions



Thus, the problem that transits service providers have with providing video is not the lack of bandwidth to transmit the content; the problem stems from the business model that the content owners face to support unicast-only subscribers is that it's not feasible to broadcast video when the costs is so high.

So here we can agree that multicast IP is one important component of economical video distribution.

The second reason why Hybrid Multicast would be useful is bandwidth optimization. HM uses IP whenever it is possible.

As we mentioned earlier in this thesis, Hybrid multicast is combination of two layers – IP and Application. IP Multicast can not be used all the time. We either have to enable multicast on all nodes, or totally ignore it and switch to Application layer multicast Application which can be enabled on almost every type of network, but even if there is IP Multicast we can not use it. There is no doubt that IP layer performs better than Application layer.

The main difference between IP multicast and ALM is that group management and packet replication are shifted from IP routers (network layer) to end hosts (application layer). Packets are transmitted through standard unicast messages while replication takes place on the end hosts themselves. With most ALM protocols, the underlying physical topology is completely hidden from the tree creation algorithms.

Though, one of the main goals of ALM is to discover as much useful information about the network as possible. Although not as efficient as IP multicast, ALM aims to eliminate the need for additional support from network routers.

Moreover it simplifies a number of other issues such as congestion control, pricing models and protocol interoperability. The main drawbacks of ALM are degradation of

efficiency (one-to-many delivery function is achieved with multiple unicast calls) and robustness (due to the dependency of the distribution trees on end hosts)

Various ALM algorithms exist with different approaches. However there is a set of metrics against which all approaches can be compared. They are:

Link Stress - This metric is defined per node/link and counts the number of identical packets sent by a node over a particular link. For IP multicast this is equal to 1.

Relative Delay Penalty (RDP) - This is the path length of the overlay tree divided by the length of the direct path. For IP multicast this is equal to 1

Stability is a measure of how quickly and how long overlay trees can be made to be "mature". When a node joins an overlay network, it is usually placed at a random location. From that point onward, there is an ongoing procedure which re-assigns overlay neighbors so that the overall cost of the tree is minimized. When no more significant gains can be achieved by shifting neighbors, the tree is considered to be mature.

Locality is a measure of a protocol's ability to minimize network attributes such as latency and cost. The goal is to minimize the tradeoff with IP multicast by considering the closeness of neighbors as a factor when constructing a routing table.

Network Performance is a direct measure of the inefficiency of using application layer multicast. The two metrics, just described, are Link Stress and Relative Delay Penalty. Previous efforts have shown that overlay routing based on locality characteristics has a maximum Link Stress of 2.5 to 3.5 and an RDP value of 1.5 (again, compared to a value of 1 for native multicast). Experiments also show that ALM protocols that do not consider locality have RDP values of between 4 and 5. These results are important as they provide a reference point for our analysis.

Robustness is a measure of the likelihood that key elements are likely to fail. Overlay networks are particularly prone to network failures due to their inherent dependency on end hosts acting as both clients and servers. It is widely assumed that end hosts are less robust than core network routers.

Having listed most important aspects of overlay networks we want to realize that there is no doubt that IP layer performs better than Application layer. Lots of papers have been published that show performance comparison, thus we are not going to repeat everything here. Obviously if any algorithm takes advantage and uses IP layer whenever it is possible that's great. This is exactly what Hybrid Multicast does.

Third, HM is implemented at application layer. We don't change anything at router and we don't modify any low level protocol. This is close to application layer multicast, but ALM ignores existence of IPM and builds purely overlay tree; we know that IP level has better optimization of network recourses, therefore HM is

better than ALM because it uses IP Multicast whenever available, and implements everything at Application layer without need to modify anything at IP layer.

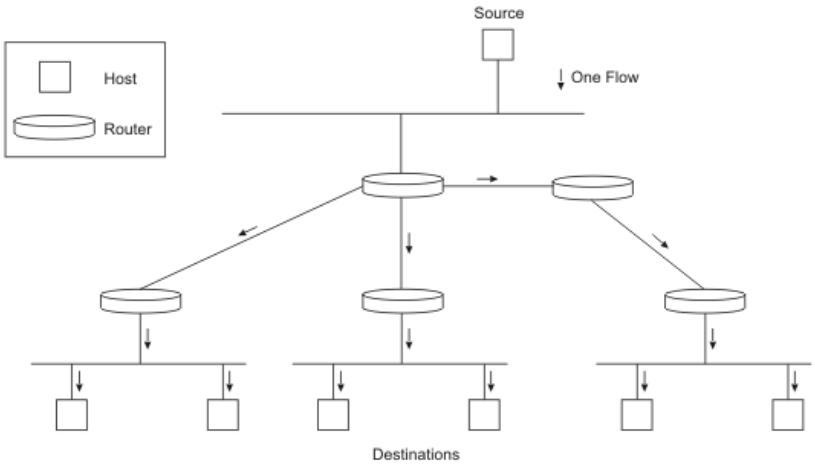
There are sufficient reasons to work on new multicasting algorithm that would solve delay, bandwidth optimization and money related problems. As we discussed Hybrid Multicast uses IP Multicast and Application layer multicast features. First we are going to describe these two protocols, and then provide in-depth Hybrid Multicast technical details.

2. IP Multicast

Internet Protocol (IP) **multicast** is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of corporate recipients and homes. Applications that take advantage of multicast include videoconferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

IP Multicast delivers source traffic to multiple receivers without adding any additional burden on the source or the receivers while using the least network bandwidth among any competing technology. Multicast packets are replicated in the network by routers enabled with Protocol Independent Multicast (PIM) and other supporting multicast protocols resulting in the most efficient delivery of data to multiple receivers. All alternatives require the source to send more than one copy of the data. Some even require the source to send an individual copy to each receiver. If there are thousands of receivers, even low-bandwidth applications benefit from using IP Multicast. High-bandwidth applications, such as MPEG video, may require a large portion of the available network bandwidth for a single stream. In these applications, probably the only way to send to more than one receiver simultaneously is by using IP Multicast. Figure 2-1 demonstrates how data from one source is delivered to several interested recipients using IP multicast.

Figure 2-1. Multicast Transmission, a Single Multicast Packet Addressed to All Intended Recipients



2.1 Multicast Group Concept

Multicast is based on the concept of a group. An arbitrary group of receivers expresses an interest in receiving a particular data stream. This group does not have any physical or geographical boundaries—the hosts can be located anywhere on the Internet. Hosts that are interested in receiving data flowing to a particular group must join the group using Internet Group Management Protocol (IGMP). Hosts must be a member of the group to receive the data stream.

IP Multicast Addresses - Multicast addresses specify an arbitrary group of IP hosts that have joined the group and want to receive traffic sent to this group.

IP Class D Addresses - The Internet Assigned Numbers Authority (IANA) controls the assignment of IP multicast addresses. It has assigned the old Class D address space to be used for IP multicast. This means that all IP multicast group addresses will fall in the range of 224.0.0.0 to 239.255.255.255.

Note: This address range is only for the group address or destination address of IP multicast traffic. The source address for multicast datagrams is always the unicast source address.

2.1.1 Reserved Link Local Addresses

The IANA has reserved addresses in the 224.0.0.0 through 224.0.0.255 to be used by network protocols on a local network segment. Packets with these addresses should never be forwarded by a router; they remain local on a particular LAN segment. They are always transmitted with a time-to-live (TTL) of 1.

Network protocols use these addresses for automatic router discovery and to communicate important routing information. For example, OSPF uses 224.0.0.5 and 224.0.0.6 to exchange link state information. Table 2-1 lists some of the well-known addresses.

Table 2-1. Link Local Addresses

Address	Usage
224.0.0.1	All systems on this subnet
224.0.0.2	All systems on this subnet
224.0.0.5	OSPF routers
224.0.0.6	OSPF designated routers
224.0.0.12	DHCP server/relay agent

2.1.2 Globally Scoped Address

The range of addresses from 224.0.1.0 through 238.255.255.255 is called globally scoped address. They can be used to multicast data between organizations and across the Internet. Some of these addresses have been reserved for use by multicast applications through IANA. For example, 224.0.1.1 has been reserved for Network Time Protocol (NTP).

2.1.3 Limited Scope Addresses

The range of addresses from 239.0.0.0 through 239.255.255.255 contains limited scope addresses or administratively scoped addresses. These are defined by RFC 2365 to be constrained to a local group or organization. Routers are typically configured with filters to prevent multicast traffic in this address range from flowing

outside an autonomous system (AS) or any user-defined domain. Within an autonomous system or domain, the limited scope address range can be further subdivided so those local multicast boundaries can be defined. This also allows for address reuse among these smaller domains.

2.1.4 Glop Addressing

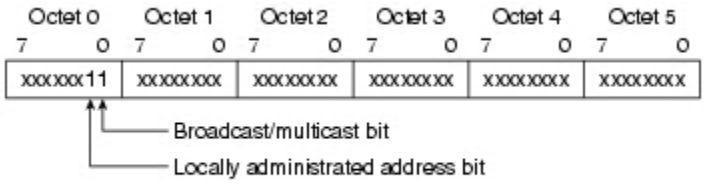
RFC 2770 proposes that the 233.0.0.0/8 address range be reserved for statically defined addresses by organizations that already have an AS number reserved. The AS number of the domain is embedded into the second and third octets of the 233.0.0.0/8 range. For example, the AS 62010 is written in hex as F23A. Separating out the two octets F2 and 3A, we get 242 and 58 in decimal. This would give us a subnet of 233.242.58.0 that would be globally reserved for AS 62010 to use.

2.1.5 Layer 2 Multicast Addresses

Normally, network interface cards (NICs) on a LAN segment will receive only packets destined for their burned-in MAC address or the broadcast MAC address. Some means had to be devised so that multiple hosts could receive the same packet and still be capable of differentiating among multicast groups. Fortunately, the IEEE LAN specifications made provisions for the transmission of broadcast and/or multicast packets. In the 802.3 standard, bit 0 of the first octet is used to indicate a

broadcast and/or multicast frame. Figure 2-2 shows the location of the broadcast/multicast bit in an Ethernet frame.

Figure 2-2. IEEE 802.3 MAC Address Format



This bit indicates that the frame is destined for an arbitrary group of hosts or all hosts on the network (in the case of the broadcast address, 0xFFFF.FFFF.FFFF).

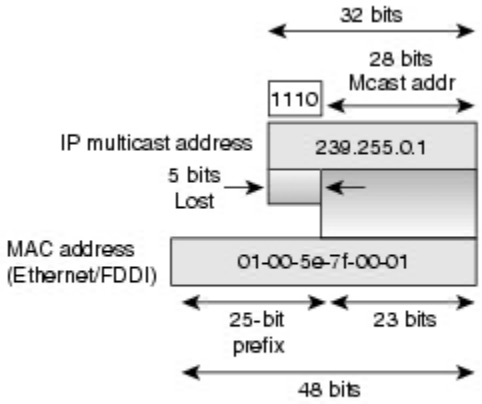
IP multicast makes use of this capability to transmit IP packets to a group of hosts on a LAN segment.

2.1.6 Ethernet MAC Address Mapping

The IANA owns a block of Ethernet MAC addresses that start with 01:00:5E in hexadecimal. Half of this block is allocated for multicast addresses. This creates the range of available Ethernet MAC addresses to be 0100.5e00.0000 through 0100.5e7f.ffff.

This allocation allows for 23 bits in the Ethernet address to correspond to the IP multicast group address. The mapping places the lower 23 bits of the IP multicast group address into these available 23 bits in the Ethernet address (shown in Figure).

Figure 2-3. Mapping of IP Multicast to Ethernet/FDDI MAC Address



Because the upper 5 bits of the IP multicast address are dropped in this mapping, the resulting address is not unique. In fact, 32 different multicast group IDs all map to the same Ethernet address.

Figure 2-4. MAC Address Ambiguities

2.2 Internet Group Management Protocol

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Hosts identify group memberships by sending IGMP messages to their local multicast router. Under IGMP, routers listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

In Version 1, there are just two different types of IGMP messages:

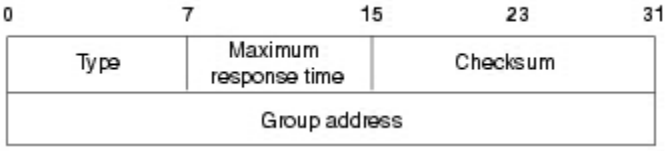
- ✓ Membership query
- ✓ Membership report

Hosts send out IGMP membership reports corresponding to a particular multicast group to indicate that they are interested in joining that group. The router periodically sends out an IGMP membership query to verify that at least one host on the subnet is still interested in receiving traffic directed to that group. When there is no reply to three consecutive IGMP membership queries, the router times out the group and stops forwarding traffic directed toward that group.

2.2.1 IGMP Version 2

RFC 2236 defines the specification for IGMP Version 2. A diagram of the packet format follows in Figure 2-5.

Figure 2-5. IGMPv2 Message Format



In Version 2, there are four types of IGMP messages:

- Membership query
- Version 1 membership report
- Version 2 membership report
- Leave group

IGMP Version 2 works basically the same as Version 1. The main difference is that there is a leave group message. The hosts now can actively communicate to the local multicast router their intention to leave the group. The router then sends out a group-specific query and determines whether there are any remaining hosts interested in receiving the traffic. If there are no replies, the router times out the group and stops forwarding the traffic. This can greatly reduce the leave latency compared to IGMP Version 1. Unwanted and unnecessary traffic can be stopped much sooner.

2.3 Multicast in the Layer 2 Switching Environment

The default behavior for a Layer 2 switch is to forward all multicast traffic to every port that belongs to the destination LAN on the switch. This would defeat the purpose of the switch, which is to limit traffic to the ports that need to receive the data.

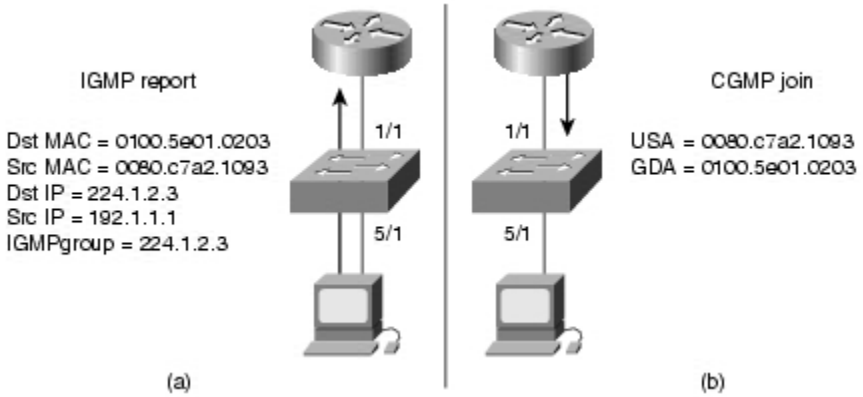
Cisco Group Management Protocol (CGMP) - CGMP is a Cisco-developed protocol that allows Catalyst switches to leverage IGMP information on Cisco routers to make Layer 2 forwarding decisions. CGMP must be configured both on the multicast routers and on the Layer 2 switches. The net result is that with CGMP, IP multicast traffic is delivered only to those Catalyst switch ports that are interested in the traffic. All other ports that have not explicitly requested the traffic will not receive it.

The basic concept of CGMP is shown in Figure 2-6. When a host joins a multicast group (part A), it multicasts an unsolicited IGMP membership report message to the target group (224.1.2.3, in this example). The IGMP report is passed through the switch to the router for the normal IGMP processing. The router (which must have CGMP enabled on this interface) receives this IGMP report and processes it as it normally would, but in addition it creates a CGMP join message and sends it to the switch.

The switch receives this CGMP join message and then adds the port to its content addressable memory (CAM) table for that multicast group. Subsequent traffic

directed to this multicast group will be forwarded out the port for that host. The router port is also added to the entry for the multicast group. Multicast routers must listen to all multicast traffic for every group because the IGMP control messages are also sent as multicast traffic. With CGMP, the switch must listen only to CGMP join and CGMP leave messages from the router. The rest of the multicast traffic is forwarded using its CAM table exactly the way the switch was designed.

Figure 2-6. Basic CGMP Operation



2.4 IGMP Snooping

IGMP snooping requires the LAN switch to examine, or snoop, some Layer 3 information in the IGMP packets sent between the hosts and the router. When the switch hears the IGMP host report from a host for a particular multicast group, the switch adds the host's port number to the associated multicast table entry. When the

switch hears the IGMP leave group message from a host, it removes the host's port from the table entry.

Because IGMP control messages are transmitted as multicast packets, they are indistinguishable from multicast data at Layer 2. A switch running IGMP snooping examine every multicast data packet to check whether it contains any pertinent IGMP information. If IGMP snooping has been implemented on a low-end switch with a slow CPU, this could have a severe performance impact when data is transmitted at high rates. The solution is to implement IGMP snooping on high-end switches with special ASICs that can perform the IGMP checks in hardware. CGMP is ideal for low-end switches without special hardware.

2.5 Multicast Distribution Trees

Multicast-capable routers create distribution trees that control the path that IP multicast traffic takes through the network to deliver traffic to all receivers. The two basic types of multicast distribution trees are source trees and shared trees.

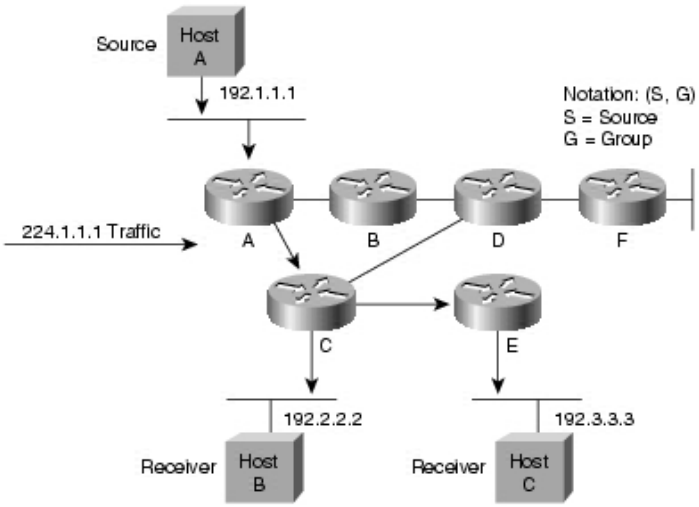
2.5.1 Source Trees

The simplest form of a multicast distribution tree is a source tree whose root is the source of the multicast tree and whose branches form a spanning tree through the

network to the receivers. Because this tree uses the shortest path through the network, it is also referred to as a shortest path tree (SPT).

Figure 2-7 shows an example of an SPT for group 224.1.1.1 rooted at the source, Host A, and connecting two receivers, hosts B and C.

Figure 2-7. Host A Shortest Path Tree



The special notation of (S,G), pronounced "S comma G," enumerates a SPT in which S is the IP address of the source and G is the multicast group address. Using this notation, the SPT for the example in Figure 2-7 would be (192.1.1.1, 224.1.1.1).

The (S,G) notation implies that a separate SPT exists for each individual source sending to each group, which is correct. For example, if Host B is also sending traffic

to group 224.1.1.1 and hosts A and C are receivers, then a separate (S,G) SPT would exist with a notation of (192.2.2.2,224.1.1.1).

2.5.2 Shared Trees

Unlike source trees that have their root at the source, shared trees use a single common root placed at some chosen point in the network. This shared root is called the rendezvous point (RP).

Figure 2-8 shows a shared tree for the group 224.2.2.2 with the root located at Router D. When using a shared tree, sources must send their traffic to the root, and then the traffic is forwarded down the shared tree to reach all receivers.

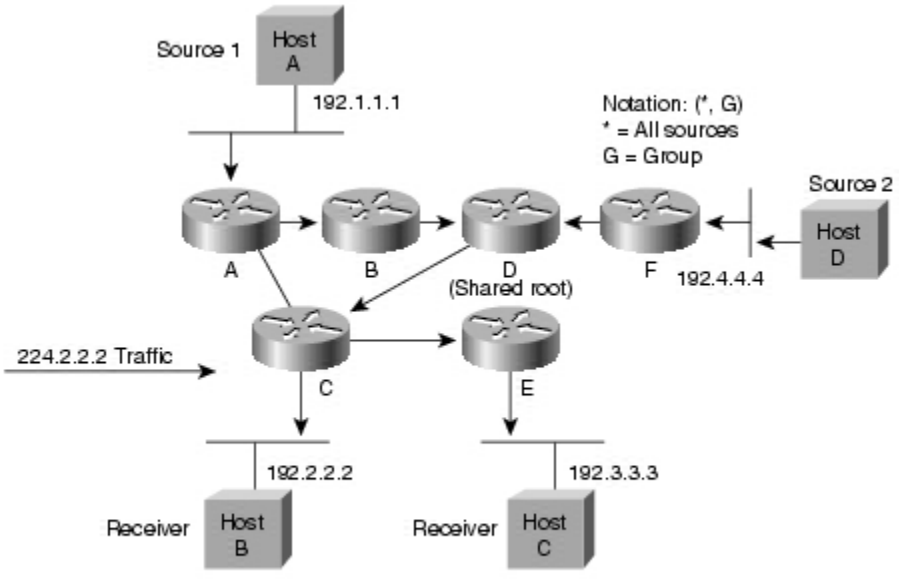


Figure 2-8. Shared Distribution Tree

In this example, multicast traffic from the source hosts A and D travels to the root (Router D) and then down the shared tree to the two receivers, hosts B and C. Because all sources in the multicast group use a common shared tree, a wildcard notation written as (*, G), pronounced "star comma G," represents the tree. In this case, * means all sources, and the G represents the multicast group. Therefore, the shared tree shown in Figure 2-8 would be written as (*, 224.2.2.2). Both SPT and shared trees are loop-free. Messages are replicated only where the tree branches.

Members of multicast groups can join or leave at any time, so the distribution trees must be dynamically updated. When all the active receivers on a particular branch stop requesting the traffic for a particular multicast group, the routers prune that branch from the distribution tree and stop forwarding traffic down that branch. If one receiver on that branch becomes active and requests the multicast traffic, the router dynamically modifies the distribution tree and starts forwarding traffic again.

Shortest path trees have the advantage of creating the optimal path between the source and the receivers. This guarantees the minimum amount of network latency for forwarding multicast traffic. This optimization does come with a price, though: The routers must maintain path information for each source. In a network that has thousands of sources and thousands of groups, this can quickly become a resource issue on the routers. Memory consumption from the size of the multicast routing table is a factor that network designers must take into consideration.

Shared trees have the advantage of requiring the minimum amount of state in each router. This lowers the overall memory requirements for a network that allows only shared trees. The disadvantage of shared trees is that, under certain circumstances, the paths between the source and receivers might not be the optimal paths—which might introduce some latency in packet delivery. Network designers must carefully consider the placement of the RP when implementing an environment with only shared trees.

2.6 Multicast Forwarding

In unicast routing, traffic is routed through the network along a single path from the source to the destination host. A unicast router does not really care about the source address—it only cares about the destination address and how to forward the traffic towards that destination. The router scans through its routing table and then forwards a single copy of the unicast packet out the correct interface in the direction of the destination.

In multicast routing, the source is sending traffic to an arbitrary group of hosts represented by a multicast group address. The multicast router must determine which direction is upstream (toward the source) and which direction (or directions) is downstream. If there are multiple downstream paths, the router replicates the packet and forwards the traffic down the appropriate downstream paths—which is not necessarily all paths. This concept of forwarding multicast traffic away from the source, rather than to the receiver, is called reverse path forwarding.

2.6.1 Reverse Path Forwarding

Reverse path forwarding is a fundamental concept in multicast routing that enables routers to correctly forward multicast traffic down the distribution tree. RPF makes use of the existing unicast routing table to determine the upstream and downstream neighbors. A router forwards a multicast packet only if it is received on the upstream interface. This RPF check helps to guarantee that the distribution tree will be loop-free.

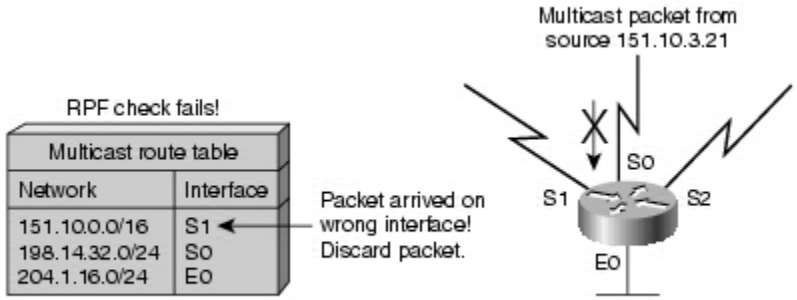
2.6.2 RPF Check

When a multicast packet arrives at a router, the router performs an RPF check on the packet. If the RPF check is successful, the packet is forwarded. Otherwise, it is dropped. For traffic flowing down a source tree, the RPF check procedure works as follows:

Step 1: Router looks up the source address in the unicast routing table to determine whether it has arrived on the interface that is on the reverse path back to the source.
Step 2: If packet has arrived on the interface leading back to the source, the RPF check is successful and the packet is forwarded.
Step 3: If the RPF check in Step 2 fails, the packet is dropped.

Figure 2-9 shows an example of an unsuccessful RPF check.

Figure 2-9. RPF Check Fails



A multicast packet from source 151.10.3.21 is received on interface S0. A check of the unicast route table shows that the interface that this router would use to forward unicast data to 151.10.3.21 is S1. Because the packet has arrived on S0, the packet will be discarded. Figure 2-10 shows an example of a successful RPF check.

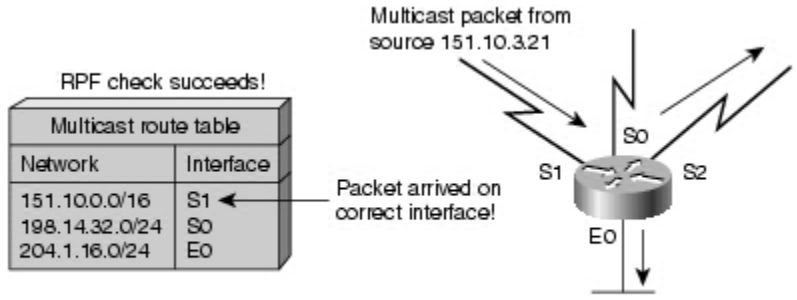


Figure 2-10. RPF Check Succeeds

This time the multicast packet has arrived on S1. The router checks the unicast routing table and finds that S1 is the correct interface. The RPF check passes and the packet is forwarded.

2.7 Protocol-Independent Multicast

Protocol-independent multicast (PIM) gets its name from the fact that it is IP routing protocol-independent. PIM can leverage whichever unicast routing protocols are used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static routes. PIM uses this unicast routing information to perform the multicast forwarding function, so it is IP protocol-independent. Although PIM is called a multicast routing protocol, it actually uses the unicast routing table to perform the reverse path forwarding (RPF) check function instead of building up a completely independent multicast routing table. PIM does not send and receive multicast routing updates between routers like other routing protocols do.

2.7.1 PIM Dense Mode

PIM Dense Mode (PIM-DM) uses a push model to flood multicast traffic to every corner of the network. This is a brute-force method for delivering data to the receivers, but in certain applications, this might be an efficient mechanism if there are active receivers on every subnet in the network.

PIM-DM initially floods multicast traffic throughout the network. Routers that do not have any downstream neighbors prune back the unwanted traffic. This process repeats every 3 minutes.

The flood and prune mechanism is how the routers accumulate their state information—by receiving the data stream. These data streams contain the source and group information so that downstream routers can build up their multicast forwarding tables. PIM-DM can support only source trees—(S,G) entries. It cannot be used to build a shared distribution tree.

2.7.2 PIM Sparse Mode

PIM Sparse Mode (PIM-SM) uses a pull model to deliver multicast traffic. Only networks that have active receivers that have explicitly requested the data will be forwarded the traffic. PIM-SM is defined in RFC 2362.

PIM-SM uses a shared tree to distribute the information about active sources. Depending on the configuration options, the traffic can remain on the shared tree or switch over to an optimized source distribution tree. The latter is the default behavior for PIM-SM on Cisco routers. The traffic starts to flow down the shared tree, and then routers along the path determine whether there is a better path to the source. If a better, more direct path exists, the designated router (the router closest to the receiver) will send a join message toward the source and then reroute the traffic along this path.

PIM-SM has the concept of an RP, since it uses shared trees—at least initially. The RP must be administratively configured in the network. Sources register with the RP, and then data is forwarded down the shared tree to the receivers. If the shared tree is

not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree. This is the default behavior in ISO. Network administrators can force traffic to stay on the shared tree by using a configuration option (lp pim spt-threshold infinity). PIM-SM scales well to a network of any size, including those with WAN links. The explicit join mechanism prevents unwanted traffic from flooding the WAN links.

2.7.3 Sparse-Dense Mode

Cisco has implemented an alternative to choosing just dense mode or just sparse mode on a router interface. This was necessitated by a change in the paradigm for forwarding multicast traffic via PIM that became apparent during its development. It turned out that it was more efficient to choose sparse or dense on a per group basis rather than a per router interface basis. Sparse-dense mode facilitates this ability.

Network administrators can also configure sparse-dense mode. This configuration option allows individual groups to be run in either sparse or dense mode, depending on whether RP information is available for that group. If the router learns RP information for a particular group, it will be treated as sparse mode; otherwise, that group will be treated as dense mode.

2.8 Multiprotocol Border Gateway Protocol

Multiprotocol Border Gateway Protocol (MBGP) gives a method for providers to distinguish which route prefixes they will use for performing multicast RPF checks. The RPF check is the fundamental mechanism that routers use to determine the paths that multicast forwarding trees will follow and successfully deliver multicast content from sources to receivers.

MBGP is described in RFC 2283, Multiprotocol Extensions for BGP-4. Since MBGP is an extension of BGP, it brings along all the administrative machinery that providers and customers like in their interdomain routing environment. Including all the inter-AS tools to filter and control routing (e.g., route maps). Therefore, by using MBGP, any network utilizing internal or external BGP can apply the multiple policy control knobs familiar in BGP to specify routing (and thereby forwarding) policy for multicast.

Two path attributes, MP_REACH_NLRI and MP_UNREACH_NLRI have been introduced in BGP4+. These new attributes create a simple way to carry two sets of routing information—one for unicast routing and one for multicast routing. The routes associated with multicast routing are used to build the multicast distribution trees.

The main advantage of MBGP is that an internet can support noncongruent unicast and multicast topologies. When the unicast and multicast topologies are congruent, MBGP can support different policies for each. MBGP provides a scalable policy based interdomain routing protocol.

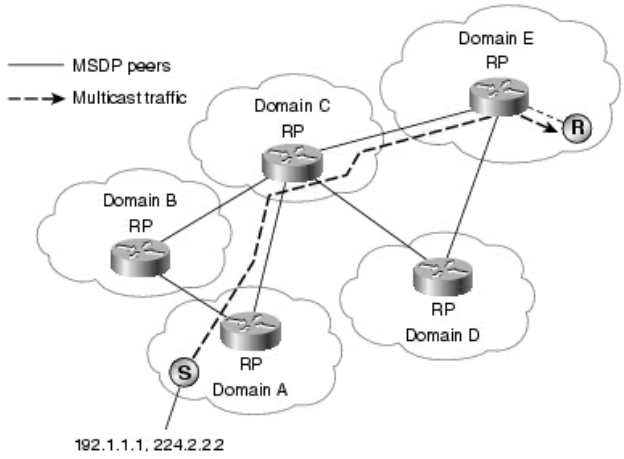
2.9 Multicast Source Discovery Protocol (MSDP)

In the PIM Sparse mode model, multicast sources and receivers must register with their local Rendezvous Point (RP). Actually, the closest router to the sources or receivers registers with the RP but the point is that the RP knows about all the sources and receivers for any particular group. RPs in one domain have no way of knowing about sources located in other domains. MSDP is an elegant way to solve this problem. MSDP is a mechanism that connects PIM-SM domains and allows RPs to share information about active sources. When RPs in remote domains know about active sources they can pass on that information to their local receivers and multicast data can be forwarded between the domains. A nice feature of MSDP is that it allows each domain to maintain an independent RP which does not rely on other domains, but it does enable RPs to forward traffic between domains.

The RP in each domain establishes an MSDP peering session using a TCP connection with the RPs in other domains or with border routers leading to the other domains. When the RP learns about a new multicast source within its own domain (through the normal PIM register mechanism), the RP encapsulates the first data packet in a

Source Active (SA) message and sends the SA to all MSDP peers. The SA is forwarded by each receiving peer using a modified RPF check, until it reaches every MSDP router in the interconnected networks—theoretically the entire multicast internet. If the receiving MSDP peer is an RP, and the RP has a (*,G) entry for the group in the SA (there is an interested receiver), the RP will create (S,G) state for the source and join to the shortest path tree for the state of the source. The encapsulated data is decapsulated and forwarded down that RP's shared tree. When the packet is received by a receiver's last hop router, the last-hop may also join the shortest path tree to the source. The source's RP periodically sends SAs, which include all sources within that RP's own domain. Figure 2-11 shows how data would flow between a source in domain A to a receiver in domain E.

Figure 2-11. MSDP Example



MDSP was developed for peering between Internet Service Providers (ISPs). ISPs did not want to rely on an RP maintained by a competing ISP to service their customers.

MSDP allows each ISP to have its own local RP and still forward and receive multicast traffic to the Internet.

2.10 Anycast RP-Logical RP

A very useful application of MSDP is called anycast RP. This is a technique for configuring a multicast sparse-mode network to provide for fault tolerance and load sharing within a single multicast domain.

Two or more RPs are configured with the same IP address on loopback interfaces—say, 10.0.0.1, for example. The loopback address should be configured as a 32 bit address. All the downstream routers are configured so that they know that their local RP's address is 10.0.0.1. IP routing automatically selects the topologically closest RP for each source and receiver. Because some sources might end up using one RP and some receivers a different RP, there needs to be some way for the RPs to exchange information about active sources. This is done with MSDP. All the RPs are configured to be MSDP peers of each other. Each RP will know about the active sources in the other RP's area. If any of the RP fails, IP routing will converge and one of the RPs will become the active RP in both areas. (Note: The Anycast RP example above uses IP addresses from RFC 1918. These IP addresses are normally blocked at inter domain borders and therefore are not accessible to other ISPs. You must use valid IP addresses if you want the RPs to be reachable from other domains)

3. Application Layer Multicast (ALM)

Also known as Overlay or Peer-to-Peer Multicast; high bandwidth multi-source multicast among widely distributed nodes is a critical capability for a wide range of important applications including audio and video conferencing, multi-party games and content distribution. Throughout the last decade, a number of research projects have explored the use of multicast as an efficient and scalable mechanism to support such group communication applications. Multicast decouples the size of the receiver set from the amount of state kept at any single node and potentially avoids redundant communication in the network.

The limited deployment of IP Multicast, a best effort network layer multicast protocol, has led to considerable interest in alternate approaches that are implemented at the application layer, using only end-systems. In an overlay or end-system multicast approach participating peers organize themselves into an overlay topology for data delivery. Each edge in this topology corresponds to a unicast path between two end-systems or peers in the underlying Internet. All multicast-related functionality is implemented at the peers instead of at routers, and the goal of the multicast protocol is to construct and maintain an efficient overlay for data transmission. Now let us review Overlay Multicast in details.

3.1 Overview of ALM

Multicasting remains a critical element in the deployment of scalable networked virtual simulation environments. Multicast provides an efficient mechanism for a source of information to reach many recipients. Traditional multicast protocols such as those defined by RFC 1075 [35] and RFC 2362 [36], provide mechanisms to support single source to a large number of destinations typically associated with streaming media or distribution of large volume data. In real-time collaborative and virtual simulation environments, the requirement is for many senders to send to the same destination group(s) simultaneously. This is commonly referred to as many-to-many multicast [37].

Even though IP multicast was introduced more than 20 years ago, it is still not widely available as an open Internet service even for one-to-many multicast [38]. The most widely used multicast capability is the MBone [39]. The MBone provides a circuit overlay inter-network that connects IP multicast capable islands by using unicast tunnel connections and is commonly used in university and research environments. Only recently have public carriers started to introduce multicast services, but then only as a private network offering where all interested parties obtain service from the same carrier [40]. These new services are based on traditional multicast services providing one-to-many multicast.

Because open multicast services generally have not been available, there has been a shift to the idea of an end-host service to provide similar capabilities. By organizing end hosts into an overlay to act as relay agents, multicast can be achieved through message forwarding among the members of the overlay using unicast across the underlying network or Internet. Two general approaches have been proposed to accomplish this. One is peer-to-peer networks that were originally designed for information sharing and messaging such as Napster and Gnutella [41]. The second approach has focused on overlay multicasting to support group communications. Here, a transport-layer overlay, on top of the underlying Internet, between the members of a multicast group establishes group communications. [42] The fundamental difference between these two approaches is that in peer-to-peer networks, the topology tends to be random relative to the underlying physical topology which results from the loosely coupled relationship between the peers. The impact on the service is that latency can be very high as information might pass across many peers some of which might be slow as well as have long physical paths between them in the underlying network. Also, large periods of message flooding can occur in peer-to-peer networks which can cause congestion and inefficient use of network capacity. In contrast, an overlay multicast protocol can be more centrally controlled by managing the resources of a service node and by efficiently managing link stress. In this case, link stress is the number of times a message transmits across the same underlying network link.

The overlays are constructed from two different strategies: mesh or tree. The mesh strategy provides for more than one path between a pair of nodes. In the tree case, a single path is established between any pair of nodes. It is also feasible to apply a mesh first, followed by a tree construction algorithm to implement overlay multicast where the idea is to take advantage of both strategies.

There are distinct differences in these two strategies that directly impact the control mechanisms of implemented overlay protocols. Tree overlays are sensitive to partitioning of the overlay because they are acyclic graphs. A graph that contains no simple cycles is defined to be acyclic where a simple path is a path that contains no repeated arcs and no repeated nodes, except the start node (root) and the end node (leaf) are the same [43]. This means that if any non-leaf member of the overlay tree leaves the overlay, voluntarily, or by failure, the tree is broken and there will be no way for members of the multicast group to communicate. The clear advantage of trees is that, inherently, there are no routing loops formed during tree construction. This greatly simplifies the routing algorithm. Mesh based overlays provide multiple or redundant connections between members of the group. This means that the overlay is less likely to be partitioned by node failure or departure. Alternate paths will already exist without the need to re-construct a path as is the case in a tree overlay. This certainly has advantages when considering needs for routing stability and offering quality of service (QoS) in the overlay. The down side to the mesh is that it is necessary to run a routing algorithm for construction of loop-free forwarding paths between group members such as a path vector algorithm. Mesh overlays may also

result in some inefficiencies as more than one copy of a message may use a link in the forward direction, e.g. link stress increases. This is not the case in a tree nor is it necessary to run a routing algorithm once the tree is established in order to prevent loops.

Traditional tree approaches use core based or route point based approaches for forwarding messages. This approach works well for one-to-many multicast. The idea is that a sender that desires to send a message to the multicast group sends the message to the core of the tree or the route point node, which in turn then forwards the message along the tree to all receivers. There is some inefficiency that results because all sender messages must first be routed to the core or route point before distribution across the tree. Current IP layer multicast routing generally uses this approach. The network inefficiency can be overcome by using source based tree algorithms in which each source builds the optimal routing tree from the source to all receivers in the group. However, this approach results in more overhead as each node must now run a routing algorithm and maintain larger amounts of supporting information. Though storing and managing larger amounts of information is easier to accomplish on an overlay host than on an ordinary network router where processing and information resources tend to be more limited. Another important aspect of overlays is whether or not they are constructed with knowledge of the Internet topology. An awareness of the underlying Internet topology improves the efficiency of the overlay. Data forwarding in overlay networks is done at the application level. Therefore, data may traverse the IP network several times before it reaches its

destination or destinations. This may result in inefficient use of network capacity and increased delays compared to transmission at the IP layer. This disadvantage is reflective of all overlay protocols but is least pronounced if the overlay network is constructed with respect to the underlying Internet topology.

There are a couple of factors that influence the scalability of the overlay, where we define the scalability as the achievable size in terms of number of nodes or possibility overall performance like end-to-end latency. The number of nodes for example, is influenced by the amount of information that a node might need to retain. If the information needs of a node grow faster than the number of nodes in the overlay, then this very well becomes the limiting factor. Limiting node information to only known neighbors, not the entire overlay, allows greater scalability. The level of effort required to build and maintain the overlay can also influence scalability. While processing power and network capacity continue to grow, it is important to keep the overhead of the protocol in balance with the stated objective of efficient communications.

Typically multicast paths are unicast paths and are the shortest paths in term of hops. The resulting shortest-path trees are good for best-effort traffic. However, when QoS is considered, such shortest-path trees may not have the resources to support the quality requirement. Therefore, it is desirable to include other resource availability considerations in the overall optimization of best path for offering QoS. Clearly, there are many alternatives and trade-offs for consideration in developing the optimal

overlay multicast protocol. These are represented in a large number of initiatives in this area. The Table below presents a summary of some of these initiatives that are in various states of experimentation and development. One observation is that it seems each of these efforts tends to focus on a specific optimization parameter that is reflective of a unique characteristic of a targeted application environment. While the intent of this research effort is not to draw a conclusion about this observation, it does however support the original proposal of this research. That is, that there are unique characteristics of the RT-DVS application environment that can be explored to enable open network overlay multicast services. The main characteristics of these applications are: real time, many-to-many, and receptive to network communication performance feedback. For example, unlike streaming video or streaming audio which are also real-time where the sender is not necessarily network aware and the transmission is one-to-many. Thus it is imperative to understand which combination of overlay strategies is optimal for RT-DVS such that the end systems cooperate to construct a good overlay structure to support many-to-many multicast.

To help answer that question, it is of value to review some of the most relevant efforts in overlay multicast protocol development. The row headings of the table indicate comparison criteria that reflect key performance elements for an overlay protocol. The criteria used for comparison are:

- Application: A general description of the targeted application environment, e.g. message information exchange, query, conferencing, streaming video.

- Overlay Topology: The term describes the nature of the organization of elements in the network. Examples would be, mesh, tree, ring, or multi-tier.
- Routing Algorithm: The routing algorithm refers to the specific algorithm used to develop the routing rules. Examples are, distance vector, Floyd's shortest path, Steiner tree, etc.
- Group formation: A general description of how groups might be formed and managed in the overlay.
- Scalability measures: A description of the scalability of the protocol and measures used for determination.
- QoS considerations: A description of quality of services that might be offered or are part of the guarantee of the protocol. Included are considerations for priority, message loss, and path failure and recovery mechanisms.
- Consideration for Node characteristics/resources: A discussion of whether the protocol considers the characteristics of a node in the development and dynamic management of the overlay. It is a recognition or consideration given to the ability of a node or host to act as an overlay relay agent.
- Node Join/leave/failures: A discussion of the technique associated with nodes joining and leaving the network either by choice or fault.

The desired outcome is to have a protocol that is QoS [44] sensitive even though the underlying Internet is not able to provide services at a consistent QoS. These comparison criteria are chosen as representative characteristics of a protocol that enable QoS sensitivity while being resource efficient and flexible. The criteria also

represent areas or features that are typically traded off based on targeted application environment.

3.2 Mesh Overlays

Overlay meshes provide the underlays that allow message forwarding between members or nodes of the overlay. Essentially these meshes provide managed tunnels between nodes across the underlying IP network. Various strategies are considered in the performing establishment of these meshes including use of graphical shapes that have well known geometric routing principles as well as information about the underlying network or Internet.

Mithos [24] uses a geometric approach where the network is embedded into a multidimensional space, with every node being assigned a unique coordinate in this space. The geometric approach greatly simplifies routing as routing is easily enabled with knowledge of the local grid coordinates. Hypercast [25] also uses this strategy. This approach uses properties of regular geometric shapes like rectangles, hypercubes, or Delaunay triangles to greatly simplify routing tables. In fact, in the cast of Hypercast, once the overlay is established, no routing protocol is necessary for the overlay.

In the rectangular approach, each node is assigned an enclosing axis-parallel multidimensional rectangle. Message forwarding is easily accomplished by sending to

the rectangle abutting at the point where the vector to the destination intersects with the current node's rectangular boundary.

For the Delaunay triangle [45] links are established according to a Delaunay triangulation of the nodes and forwarding is accomplished similar to the rectangular approach. Delaunay tangles main characteristic is that for each circumscribing circle of a triangle formed by three nodes, no other node of the graph is in the interior circle. While Pastry [26] is a peer-to-peer based protocol, the substrate self-creates a messaging routing overlay on the Internet that operates in a way that makes the overlay look like a mesh. This is accomplished by each node having a unique 128-bit node ID.

Using this unique ID, Pastry routes a message to the active node that is numerically closest. This approach provides a level of reliability since the idea is based on an active node. No further routing protocol is necessary for the local node to make this decision, unlike a tree based approach where tree re-construction is likely required in the case of a node going inactive. Pastry also uses a metric for closest node such as latency so that optimum choice for forwarding is always made.

The HyperCast protocol builds logical overlays based on geometric properties of a logical graph. HyperCast currently implements both the hypercube and Delaunay triangles. In each case, applications communicate with its neighbors in the geometric overlay, both in one-to-many multicast and many-to-one or unicast. The key

advantage of using geometric logical relationships is that once the overlay is established, there is no further need for a routing protocol. The key disadvantage of this approach is that the underlying physical network is completely ignored which makes it difficult to consider end-to-end latency performance. Another disadvantage is that hypercube overlays must be formed sequentially with the result that for a large set of nodes, it is likely that it will take a long time to construct the overlay and also complicate departure or joining of a single node. In the case of Delaunay triangles, overlay construction can be accomplished faster as they can be built in a distributed fashion.

The logical hypercube overlay network topology organizes the applications into a logical n-dimensional hypercube. Each node is identified by a label (e.g., "010"), which indicates the position of the node in the logical hypercube. Message forwarding is easily accomplished by logical reference to nearest neighbor, hence no need for a routing protocol once the overlay is established.

A Delaunay triangulation uses the special characteristic that for each circumscribing circle of a triangle formed by three nodes, no other node of the graph is in the interior of the circle. Each node in a Delaunay triangulation has (x,y) coordinates which depict a point in the plane. This approach allows each application to derive the next hop forwarding information without the need of a routing protocol.

Tapestry is a unicast overlay network that provides the infrastructure for other multicast protocols like Bayeux. The Tapestry routing mechanism is similar to longest prefix routing used in the CIDR IP infrastructure of the Internet RFC 1518.

The routing mechanism is a hash-prefix system which essentially results in every destination node being the root of its own tree that is the unique spanning tree across all nodes. The approach is inherently decentralized. Tapestry includes fault tolerant mechanisms that provide redundant paths to all destinations. This strategy effectively routes around failed nodes, in essence providing a mesh type infrastructure.

3.3 Tree Overlays

oSTREAM [46] is a tree based overlay that is specifically designed for one-to-many on-demand media distribution. The approach is to establish minimum spanning tree and use media buffering at the host to aid the distribution of asynchronous service requests for the same streaming media. While this strategy is not particularly useful for a many-to-many environment, there are some similarities in forwarding the same information to asynchronous requests in the web services model. This might apply, for example, in the case where more static background information such as terrain models or weather information might be asynchronously requested from group members in a simulation. In these types of data distribution requests, there would be link and server efficiency advantages in using this approach.

Yoid [47] employs a single shared tree for all members of the group. The links are unicast multi-router-hop paths. Trees are managed by a concept of child/parent relationship amongst members of the overlay. A member with no parent is the root member, and members with no children are leaf members and stub members are always leaf members. Each member divides the set of all other members into two groups called parent-side and child-side members. The groups are defined such that parent side members are all members reachable via the parent and all others are child-side members. Each member must mange this and understand how to protect from partition and make decisions on a new parent if the tree is portioned.

Topology Aware Grouping (TAG) [48] exploits underlying network topology information to build efficient overlay tree networks among multicast group members. TAG uses information about path overlap among members to construct a tree that reduces the overlay relative delay penalty, and reduces the number of duplicate copies of a packet on the same link. Each member of a TAG multicast session determines the path from the root of the session to itself and determines its parent and children. TAG nodes need only the IP addresses and paths of their parent and children nodes. TAG is unusual in that it constructs its overlay tree based on delay and considers bandwidth to break ties among paths with similar delays when constructing the overlay. This approach has merit for consideration as it provides the opportunity for guaranteeing end-to-end latency performance for the overlay. TAG does this by taking advantage of the underlying network shortest path topology information maintained in the underlying network IP routers.

The Overlay Multicast Network Infrastructure (OMNI) [49] is a two-tier approach to overlay multicast. The lower tier consists of a set of devices or service nodes that are distributed throughout an underlying network infrastructure like the Internet. The lower tier provides data distribution services to any host connected to an OMNI node over a directed spanning tree rooted at the source OMNI node. An end-host subscribes with a single OMNI node to receive multicast data service. The OMNI nodes organize themselves into an overlay which forms the multicast data delivery backbone. For the second layer, the data delivery path from the OMNI nodes to its clients is independent of the data delivery path used in the overlay backbone. This path can be built using network layer multicast, application-layer multicast, or a set of unicast paths.

Overcast [50] is a single source multicast overlay designed for on-demand and live data delivery. The protocol is single source tree based with some added features to provide reliable delivery to multicast groups. Reliability is provided by using TCP e.g., HTTP over port 80.

3.4 Hybrid Mesh-Tree Overlays

As indicated earlier, there are two basic methods for the construction of overlay trees for data delivery. First, one can construct a tree directly by members selecting their parents from amongst group members that they know. The second is to construct a

well connected mesh of the group members and then use standard shortest path tree construction algorithms to establish the minimum distribution spanning tree. Protocols such as Narada [51] apply a two step process where in the first step an overlay mesh is constructed and then a tree is constructed using a shortest path algorithm on the nodes of the mesh.

While the mesh construction can be accomplished in an arbitrary fashion relative to the underlying infrastructure, there is value using knowledge of the underlying structure in building the mesh so as to improve over all performance of the overlay. The hypercube and Delaunay triangle techniques described above, for example, can easily be applied to build meshes without regard to underlying infrastructure. The technique might work well for peer-to-peer information exchange where end-to-end performance constraints are stringent.

Narada, however, tries to build the mesh in recognition of underlying network performance characteristics. Narada applies reverse shortest path algorithm in the second step to establish shortest path minimum spanning trees with each tree rooted at the source node. Several advantages result from this approach:

- Group management can be accomplished at the mesh level and more easily allows for the use of a standard group management protocol like IGMP at the local distributed level.
- Meshes are more resilient than trees; repair and optimization are easier to accomplish as loop avoidance is not required during this process.

- There are many existing algorithms for construction of shortest path trees on top of the mesh.

Another protocol that uses this general strategy is Tmesh [52], which uses an algorithm to determine shortcuts in the tree. The idea is to correlate measurable characteristics with a computed reduction in node-pair latencies attributed to each shortcut. This information is then used in a heuristic to select a shortcut with objective of improving overall latency.

3.5 Peer-To-Peer Overlays

There is another definition of overlay networking that is normally associated with information or message exchange at the application layer without using the services of an intermediary host, such as in a client server application, called peer-to-peer. Peer-to-peer is typically used to define an end point or application layer exchange of information. We have chosen not to treat them specifically as a separate category, as they apply routing principles similar to overlays in general, but the relationship requires some explanation as there are many developments in progress for the implementation of self-organizing and decentralized peer-to-peer overlay networks. These efforts support new distributed applications which require information discover and message exchange across a network of loosely coupled applications.

The point-to-point overlays typically implement distributed hash tables that allow for location of an object within a bounded number of routing hops. They tend to exploit proximity in the underlying network topology in locating objects and routing.

Multicasting is built on top of these peer-to-peer overlays. Examples of this strategy include Borg [53] and Scribe [54] which are built on Pastry [55] and Bayeux [56] which is built on Tapestry. Both Pastry and Tapestry provide unicast routing based on prefix-routing, and use a proximity neighbor selection mechanism to take advantage of the underlying physical network. Tapestry and Pastry also provide sensitivity to QoS by constraining the routing distance per overlay hop, resulting in efficient point to point routing between nodes in the overlay mesh. Scribe uses reverse-path forwarding and Bayeux uses forward-path forwarding. The reverse-path construction of Scribe causes many short links in the multicast scheme which provides for lower link stress than Bayeux. Borg uses a hybrid multicast scheme to take advantage of asymmetry in routing in structured point-to-point networks. Borg builds the upper part of a multicast tree using a hybrid of forward-path forwarding and reverse-path forwarding and leverages the reverse-path multicast scheme for its low link stress by building the lower part of the multicast tree using reverse-path forwarding.

Peer-to-peer overlay networks generally are unpredictable and therefore impact the quality of delivery of messages. Strategies such as message preservation priority in queues and expiration times are sometimes used to enable some level of QoS. An

example is to discard messages that have reached expiration times, or at least give priority to those that have not.

4. Hybrid Multicast

Recognizing the problems we have, realizing that the existing multicast protocols are essential but not perfect, new algorithm for Multicast called Hybrid Multicast is proposed in this thesis. The idea behind Hybrid multicast is to combine IP layer and Application layer and make them talk. How do we accomplish such a task? One way to do that is to implement everything at the Application layer and use advantages of the IP layer. The reason why we provide the solution at the application layer is the simplicity of its deployment. If we recall the problem with IP multicast, it is that not everybody has access to such upgraded routers and therefore if we implement something at IP layer it won't be feasible for all users to gain the full benefit from Hybrid Multicast. So we address this problem at the Application layer. All implementation goes to Content distributor (Server) and End-nodes.

Another good thing about implementing everything at the Application Layer is that we do not change the network infrastructure.

4.1 Overview of Hybrid Multicast

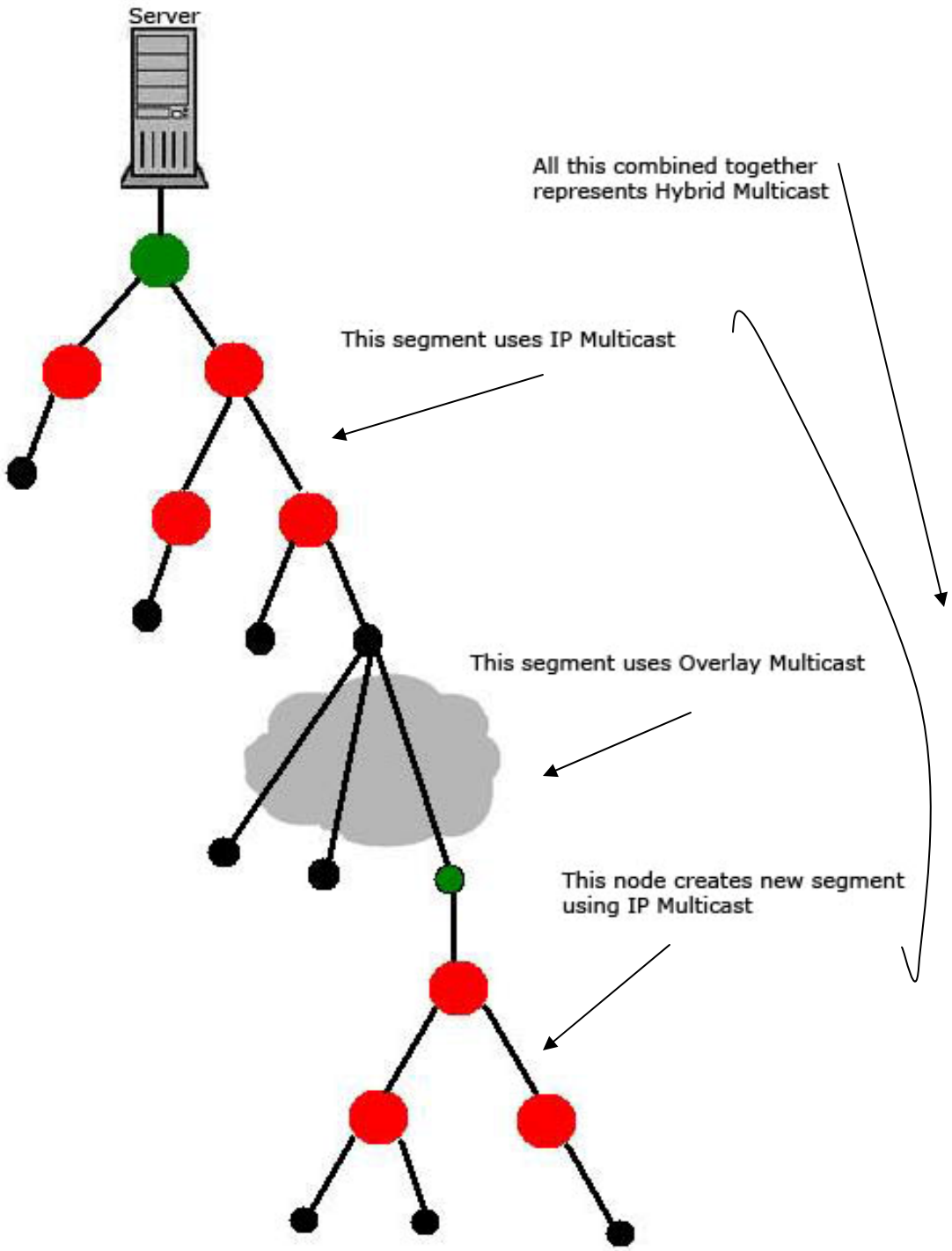
Hybrid multicast (HM) builds a multicast tree which involves the physical link and overlay link. First we start with IP Multicast (not necessarily IPM, could be ALM) and each End-Node on IP Multicast can build Overlay Multicast. Every node

involved in Hybrid Multicast is capable of extending the network and building new multicast trees. The tree building process is centralized. The shortest path calculation is done at a specially designated node called Rendezvous Point (RP). There is only one RP per Hybrid Multicast group.

To allow fast deployment we do not make any changes of the network infrastructure. All we have to do is to install HM software at Content owner and start streaming the data. On End-user side we also have to install Media Player that supports HM. Using HM any Internet Service Provider can stream data to users, without increasing the traffic on network, or having to pay too much for each individual unicast transmission.

The Hybrid Multicast is illustrated in Figure 3-1.

Figure 3-1. Hybrid Multicast



Hybrid Multicast as we see consists of a combination of IP Links and Overlay Links. In the figure above HM starts with IP Multicast, however it could start with ALM. Before we go into details about how HM works, we need to define some key components. We need to implement software for three different platforms, they are:

4.1.1 Content Server: This is the node who is distributing the media content. Content Server is instructed by Rendezvous Point (described later) and the shortest path for each end user is provided. The task of Content Server is not that tough, just building the destination addresses table and sending data to each End-node.

4.1.2 Rendezvous Point: the name was chosen just because it has similar job as what RP does in IP Multicast. The difference here is that RP in IP Multicast constructs IP layer multicast tree, while in HM it constructs hybrid tree. The way these two RPs do their job is quit different, but the task is similar. RP is responsible for building the shortest path from Content Server to End-Node. As we know many multicast implementations use Dijkstra's [57] or Bellman-Ford [58] algorithm. Any of these algorithms is fine for hybrid multicast. In order to construct the acyclic, connected tree with shortest path we need to have information of network topology at RP. Of course RP can NOT know everything about the whole Internet. All we can afford is knowledge at the routers; the routing table of local network nodes. (So we assume that Rendezvous Point only has information that is available at the router that it is attached to) We can afford this knowledge, even if we have to modify something at the router; that would require us to write additional piece of code and change only

one router, which is quite possible. Even if we can not do that, Hybrid Multicast will still work, but as old saint suggests the more knowledge we have, the better.

HM is mainly concerned about the delay. So the weight of link is the delay. The smaller delay, the better chance it to be selected for hybrid multicast tree. The Hybrid Multicast software application sends ‘ping’ message to destination node. We choose to send five ping messages and the average is calculated. We need as better approximation to the real delay as possible.

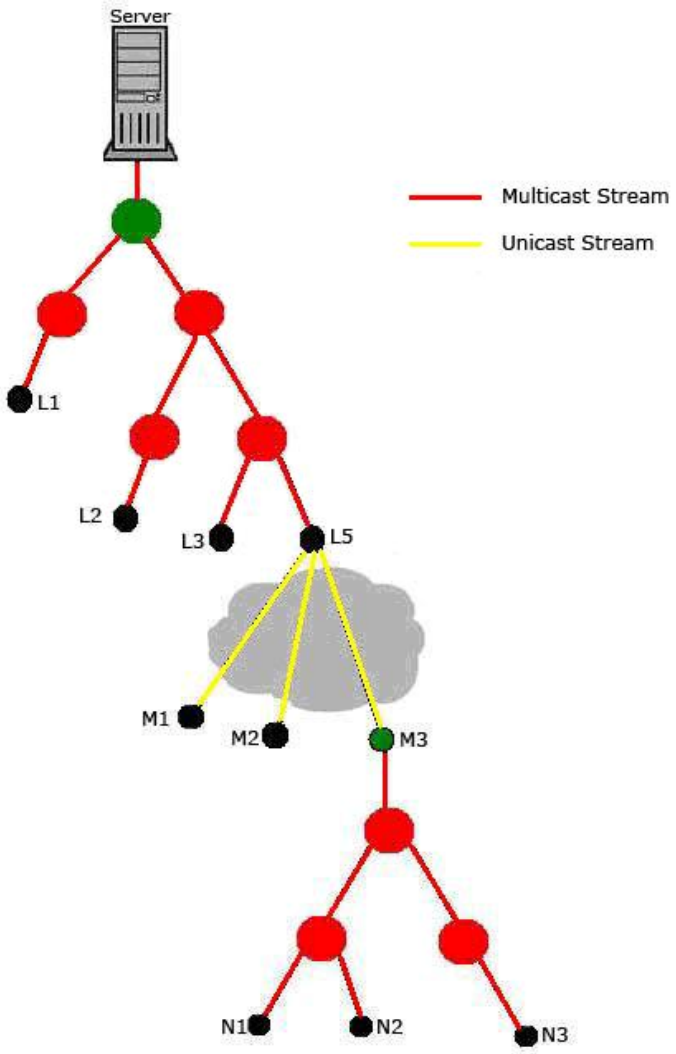
When calculating the delay in Hybrid multicast (i.e. joining IP Multicast to Overlay Multicast) we usually select the delay from the server to the closest intermediate node and the delay from the intermediate node to the destination. These numbers are summed and the total delay is the weight RP would use for the calculation of new shortest path on multicast tree.

4.1.3 Leaf Node – is a computer that is attached to Hybrid Multicast. What we should always keep in mind about Leaf Nodes is that they are capable of extending the network. Leaf Nodes in Hybrid Multicast are as important as Content Server (distributor of media) or Rendezvous Point (responsible for multicast tree). In other words, Leaf Nodes serve as a mediator between Server and End-Node willing to join Multicast group, but not having IP Multicast capabilities.

Let’s say we started IP Multicast, nodes L1, L2, L3, L4 and L5 are Leaf Nodes. When a new node wants to join first we need to identify if that machine is attached to IP

multicast. If yes there's no need of Hybrid Multicast, everything will be done at the IP layer, and if there's only unicast capability we need to ask the Leaf Nodes for help. In figure 3-2 we can see the process of media distribution through Hybrid Multicast.

Figure 3-2. Media Distribution on Hybrid Multicast



Nodes M1, M2 and M3 are not connected to the Server by IP Multicast enabled router, therefore they have to use unicast stream. L5 receives data from Server by IP

Multicast, encapsulated in UDP, and then the multicast packet is encapsulated as unicast packet and sent to M1, M2 and M3. This is done at the Application layer. As shown in Figure 3-2 there are still unicast streams for each unicast-only nodes, but the costs of the content owner will be lower than they would be if unicasst streams had to be sent the entire way starting from the server.

With Hybrid Multicast content owner can profitably provide just one stream to L5 and pay only for multiple unicast streams in the unicast-only local provider's network. The result is that video content that couldn't otherwise be provided is now available, and the transit service provider retains a profitable customer in the content owner.

The question here is why L5 and how we pick this node. We will defer the answer to this question to a later section 4.3 (how to build hybrid path), but just to mention that Hybrid Multicast not only provides multicast capabilities to unicat-only nodes, but also achieves optimal use of Internet resources

4.2 Join and Leave Operations

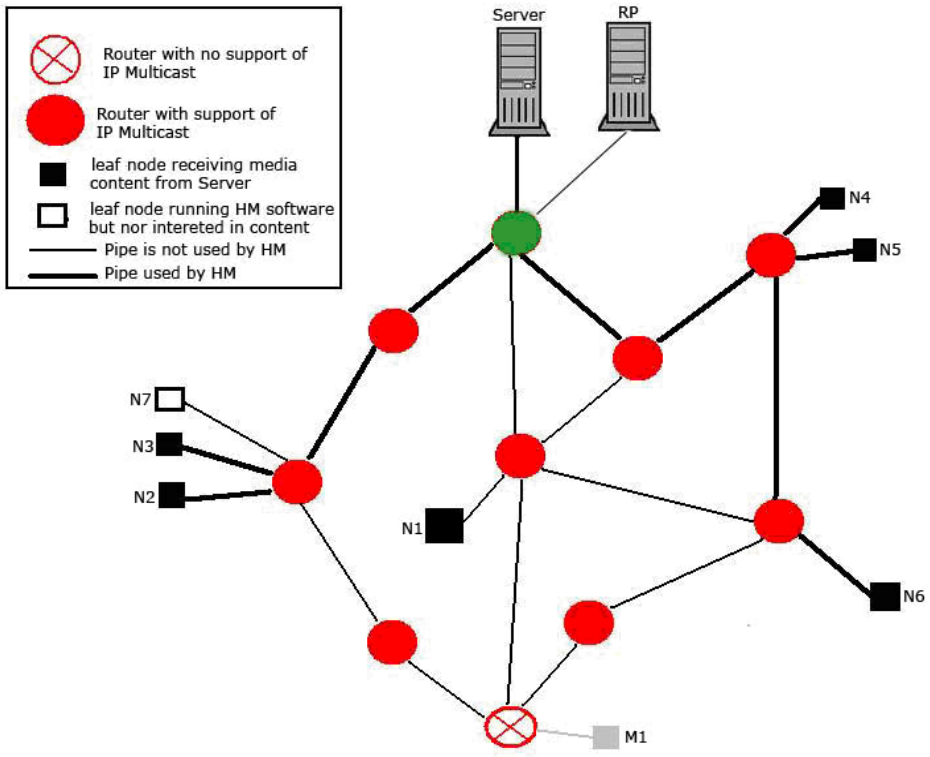
Join and Leave operation are two most important messages sent to RP. A joining host first needs to determine, is it attached to IP Multicast enabled router or not; if not then find out where is Rendezvous Point. We can put RP address in a public directory and everybody can gain access to it. If node is willing to join the multicast group and it has unicast-only capability, it will send join message to Rendezvous Point. RP will calculate all paths from available leaf nodes to the joining host and select the best one.

Regarding the leave operation, leaving multicast group is not as insignificant as it might seem. If host is so called ’Leaf Node’ then it’s fine, we don’t expect any changes of multicast tree, but if it is an intermediate node, than RP has to make changes in its table and the second best path has to be provided for other nodes replacing the current one. Join and Leave operations are discussed in the next section ‘how to build hybrid path’.

4.3 How to Build Hybrid Path

This is the most important part of hybrid multicast. Building hybrid path would be the same problem as combining two layers and making them talk to each other. Here we will refer to a figure, where we have a sample network and we will execute the hybrid multicast algorithm on it. Examples make it easy to understand the process and steps of the algorithm. As we can see in the Figure 3-3 we start with IP Multicast. We have content owner (server) and Rendezvous point running. As usual IP multicast enabled nodes join multicast group and they receive data from the server; everything is done at IP layer and no need of Hybrid Multicast yet.

Figure 3-3. A Sample Network Topology



The picture above shows the topology of a network where we could run the Hybrid tree algorithm. Please note that this is an actual physical topology of the network. As we all know in order to construct multicast tree we need a table that contains Bandwidth and Delay information. Since the algorithm is for hybrid multicast, we would have to store something like hybrid table at RP.

In Figure 3-3 the red points are routers with IP multicast capability and shaded quadrangles (black shaded) are computers that run Hybrid Multicast software. The silver quadrangle is computer machine connected to the server through unicast-only router.

If any user is interested in receiving media from this server, they first have to download hybrid multicast software and install it, then send join request to the server. All join messages actually go to RP. After RP receives the *join* message the following actions will be taken:

RP already knows that this computer is not attached to IP multicast router, so multicasting should be done at Application layer (actually through unicasting)

RP sends list of *Leaf-Members* to New Member; also called *Candidate Nodes* which can be narrowed by using network coordinate service (such as GNP [29])

- New-Member will ping all the candidate Leaf-Nodes and store RTT values for each in two dimensional array. One column stores the ID of leaf node, and column keeps the delay information for this node. This matrix is sent to RP.
- RP will select the best Leaf-Member (shortest RTT) and send message to this particular Leal-Node saying that it has to extend the network tree to New Member.

In our case, RP would send to M1 the following list: N1, N2, N3, N4, N5 and N6. Please note that N7 has not been included; this node is not multicast group member. M1 will ping each of this node and store this information, later it will send it back to RP.

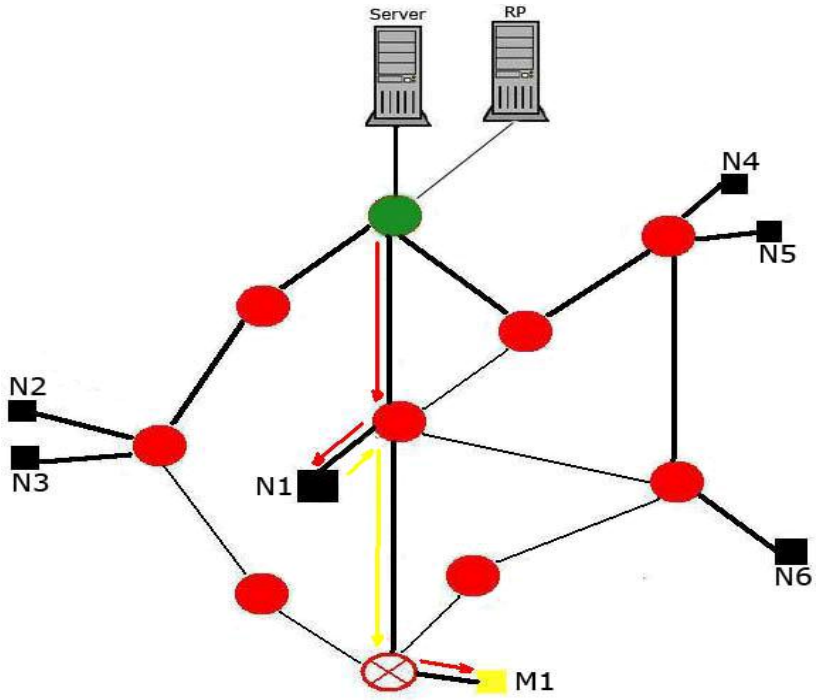
Table 3-1. Storing Delay Information in Two Dimensional Array

ID	Delay
N1	100 ms
N2	120 ms
N3	119 ms
N4	122 ms
N5	122 ms
N6	119 ms

We see that N1 has the smallest delay to M1, so obviously RP would tell N1 to extend network to M1. The way ‘network extension’ works is as follows: N1 receives UDP encapsulated multicast message, the packet goes all the way up to Application

layer, and the hybrid multicast software encapsulates the same packet as unicast packet to N1. The hybrid data flow is shown in figure 3-4.

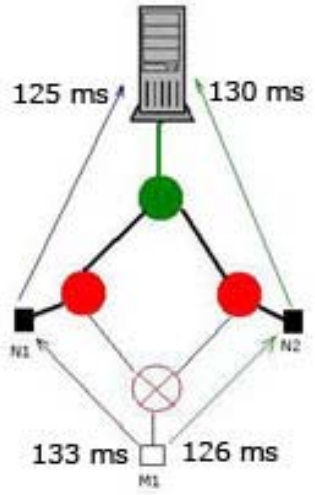
Figure 3-4. Data Flow on Hybrid Path



N1 is a computer that is receiving media stream from Content Server. Now RP has informed N1 to forward all the coming packages to M1. This message would be a unicast message. The only drawback with this solution (in this particular case) is that one packet travels back and forth on the same pipe two times, but otherwise it would be impossible to extend the network to New Member, whose router does not have IP Multicast capability. The red arrow line shows the data flow from Server to Leaf-Node N1 and then to New member M1.

The delay between intermediate node and joining host is considered, but what about delay between Server and intermediate node? Next paragraph answers this question. In figure 3-5 we have a simple scenario, how to pick the right intermediate node.

Figure 3-5. Total Delay Calculation



As we discussed earlier the new member M1 will ping all the *Leaf Nodes*. For simplicity we show only two nodes. The RTT value is obtained by pinging each Leaf Node and the result is stored into two dimensional array. Take a look at Table 2.

Table 3-2. Delay Information Stored in Two Dimensional Array

Leaf Node ID	Delay to M1	Delay to Server
N1	133 ms	125 ms
N2	126 ms	130 ms

The delay between M1 (new member willing to join) and N1 (Leaf Node) is 133ms, and between N2 (another Leaf Node) is 126ms. Obviously N2 seems to be the better one, but the delay between N2 and Server is bigger then the delay between Server and N1. What typically is done in this situation is to calculate of the total delay.

Total Delay N1 = 133 ms + 125 ms = 258 ms

Total Delay N2 = 126 ms + 130 ms = 256 ms

In this case N2 seems to be better option so RP will select N2 for M1.

Note that RTT value from *New Member* to *Leaf Node* is calculated by *New Member*, while RTT time between Server and *Leaf Node* is kept at RP. RP sends heartbeat message to all *Leaf Nodes* and expects response back to make sure they are still up and running. All heartbeat messages are unicast encapsulated.

4.4 Stability Coefficient

Hybrid Multicast is one of the multicast protocols, that provides reliable service. Recent growth of multicast capable networks requires us to address every important issue. One of the most important issues in any multicast protocol is stability. Stability coefficient is little innovation here.

We assume that those nodes that have IP Multicast enabled are more stable, i.e. the chance of failure is small. Why is that? First of all router never leaves the network. Such high-end routers are most frequently owed by universities, research companies and they keep it always on. New routers have more capabilities and speed. Bandwidth is increased and buffer can store more data. All these make us believe that IP Multicast enabled nodes are more stable than those end nodes with unicast-only capability.

Of course the fact that routers never leave network doesn't necessarily mean they will be always working. Router might fail, get congested or link connecting particular router to the rest of the world might get corrupted, so there is still chance of such router being out of order. We consider this situation too. As shown above we have Total Delay calculation formula.

For total delay calculation we add one more coefficient, Stability coefficient for each intermediate node. Each node has its own stability coefficient, the more stable the

node is the smaller coefficient it has. Most multicast protocols select the smallest delay.

Total delay = (delay from server to intermediate node + delay from intermediate node to joining node) * stability coefficient.

Now the question is how to assign number to nodes that represents accurately the stability condition. We use *Message stability detection for reliable multicast* [1] as a reference for this purpose.

In case that there is similar number for several intermediate nodes, that have same delay, same stability coefficient and same distance the tie is broken according to the smallest ID.

4.5 Retransmission in Hybrid Multicast

Regarding the stability and reliability, Hybrid Multicast has retransmission. This is implemented at application layer. IP Multicast doesn't provide any retransmission. We use Negative Acknowledgments (NAK) for this purpose.

The idea behind NAK is simple. We have time out for each packet, before time-out fires we keep packet in buffer. If node on the next hop doesn't receive data and its timer fires, it will signal with NAK message and parent node will resend the data. The packet is kept in the buffer till timeouts. If packet is successfully delivered there is no need of any acknowledgement. Hybrid Multicast is not the only multicast protocol that uses NAK. There are lots of works [27] done that show performance of NAK on multicast.

4.6 Reliability in Hybrid Multicast

The purpose of reliability is to maintain efficient backup routes for reconstructing hybrid multicast tree quickly. In most conventional methods after a node leaves the tree, its children start searching for a new parent. In hybrid multicast we have implemented a so called "proactive" approach to find a new parent over the tree before the current parent fails or leaves.

A proactive approach allows a node to find it new parent immediately so as to switch to the backup route smoothly. In hybrid multicast the structure of the hybrid tree using a redundant degree can decide a new parent without much overhead.

We want hybrid multicast to provide not only new features, but also be as much reliable as possible. We should minimize overhead as much as we can, so the algorithm can perform faster and users won't notice any performance degradation.

Hybrid multicast implements the multicast functionality at end-nodes, in a way different from IP Multicast, which unrealistically requires global deployment of routers with IP multicasting capability. Application Layer Multicast (ALM) like Hybrid Multicast needs only installation of software and requires no change in the current network infrastructure; however usage of network resources is far from being optimal. Overlay trees are not aware of physical condition of network links and often same packet travels on the same link several times. In HM we avoid this situation whenever possible, i.e. we use IP Multicast whenever possible, we connect IP Multicast with overlay links to unicast-only nodes and extend the multicast tree in a hybrid way, n addition HM provides flexibility in routing such as multipath packet transfer and load balancing.

There are several measures to evaluate effectiveness of multicast protocols. They are: quality of data delivery path that is measured by stress, stretch and node degree parameters of overlay multicast tree; robustness of the overlay (for ALM) that is

measured by the recovery time to reconstruct a packet delivery tree after sudden end host failures; and control overhead that represents protocol scalability for large number of receivers.

In most ALM protocols, each end host is a member of the delivery tree, and it leaves freely anytime, which also might fail. This is not a problem in IP multicast, because the non-leaf nodes in the delivery tree are routers and they never leave the network, the multicast tree unless they get congested or fail. In Hybrid Multicast, one of the problems we have to consider is to reconstruct the overlay multicast tree after a node departure. The time to receive the data flow again after a node departure is important for the multicast applications, especially for application that serve live media streaming (e.g. video conferencing).

So what we should do here is to reconstruct the hybrid path so quickly that user won't even recognize what happened (in terms of leave/fail). Therefore it is important to maintain media quality by quickly reconstructing the multicast tree. Little of no attention has been given to this problem in most Overlay Multicast applications, however Hybrid Multicast fully addresses this problem and reduces the overhead time to minimum.

As we discussed if nodes start searching for new parent after departure of their old parent, the media quality will suffer. It usually takes several seconds/minutes to

restore the overlay tree, so let’s just spend this time before node departs, instead of doing this after.

So proactive approach takes into account the node departure before it happens. The basic idea is that each node in the hybrid multicast tree pre-computes a backup route. Recall that in the process of constructing a hybrid path, the new joining host calculates the delay to each intermediate node and sends it to RP. After receiving this data RP adds delay from server to intermediate node and adds stability coefficient, then the best (the lowest) delay is selected and the intermediate node having smallest total delay is notified to extend the multicast tree to the new user. In other words the encapsulated packet with Multicast group address must be encapsulated with address of the new host and resent as unicast packet. Each node has a list of children nodes on the hybrid multicast tree. If any of these nodes is willing to leave, and leave message is sent to RP, then RP will select the second best route from the list it already has calculated for a specific node. The information is kept at RP for all time until the node decides to leave the multicast group. Hybrid Multicast uses knowledge of parent-children information.

This approach might seem good enough, however we can not tell for sure that the state of the network will not change over time. Sometimes the work load increases at routers, links or other network devices that are involved in hybrid multicast. So here we take a look at another approach, which calculates the degree each host has, and ensures backup route proactively whenever a node leaves or joins. Degree represents

the number of outbound links. It is inevitable to consider the degree bound in overlay multicast, which can be easily observed in streaming applications. Each host limits the number of children on the tree it is willing to support. For example, let's assume the bit rate of media is *B* and the outbound bandwidth of an end host is *bi*. The total number of connections it can establish with the downlink world is [*bi* / ***B***]. We denote the total number of connections that can be supported as the maximum degree of End-node. The parent node calculates the residual degree [22] of its children first. Residual degree is represented as unused degree. Let $d_m(x)$ be the maximum degree, $d_u(x)$ be the used degree and $d_r(x)$ be the residual degree of node x. obviously $d_m(x)=d_u(x)+d_r(x)$.

With the degree constraints, when its children do not have enough residual degrees to ensure their backup routes, the parent node employs the residual degrees of the grandchildren nodes and below in calculating until they can finally ensure backup routes. So basically each node keeps looking for backup route up the hierarchy of tree unless it finds it. If parent node is exhausted, child node will try to backup at grandparent, if grandparent is exhausted it will jump up to grand-grandparent and so on. This calculation process generates extra data overheads and is not scalable. Volume of control traffic can be significant for some overlay multicast applications.

We propose a new approach here to avoid the degree limitation and generating heavy overheads. By forcing at least one reserved degree in each host, backup routes can be

always established among the parent and children nodes. It means this approach doesn't generate much overhead to ensure backup routes.

Next we are going to provide an overview of typical ALM protocols, the problem formulation, afterwards the solution is provided with detailed description. We will evaluate the approach in terms of performance in NS2.

4.6.1 Overview of Typical ALM Protocols

Most Application Layer Multicast protocols are concerned with how to construct an efficient multicast tree. ALMI [12] employs a centralized solution. In a centralized scheme, a central controller is used to compute and instruct the construction of the delivery tree based on the information of metrics (e.g. distances, degree bounds) provided by the overlay members. This information is exchanged between nodes. Such a measurement technique often consumes a lot of bandwidth. This type of mechanism exchanges information with some hosts constantly and is called Mesh protocol. There are also Narada [51] and Scattercast [59] known as Mesh-first protocol. The Narada [51] protocol keeps state about all other members that are part of the group. This information is also periodically refreshed. Distribution of such state information about each member to all other members leads to relatively high control overhead. The Scattercast [59] protocol builds a routing table using a protocol called Gossamer for neighbor discovery in environment with multicast proxies. As most

mesh protocols require each member to estimate distance to all or a large number of the members, they are not suitable for large scale applications.

In contrast, Overcast and Peercast are distributed tree based protocols for larger groups. This constructs a shared data delivery tree first. Packets are transported from the source node to its children and from the children to the grandchildren and below in order. In some methods, each member discovers a few other members of the multicast group that are not its neighbors on the overlay tree and establishes and maintains additional control links to these members after tree construction. The Yoid protocol incorporates loop detection and avoidance mechanisms when members change parents in the tree. The Overcast protocol targets creating high bandwidth channels from one source to receivers. It does not explicitly consider the latency, but minimizing tree depth reduces buffering delays. The Peercast protocol considers join and leave algorithm. It uses the round trip time method in join and the grandfather method in leave. A node in tree based protocols does not make as many connections as in mesh protocols. Tree construction of our proposal is based on the Peercast algorithm. Additionally our proposal considers robustness against node leaves and failures.

OMNI defines a local transformation for the overlay tree to minimize the average latency of the entire hosts with degree constraints. Local transformation occurs between nearby nodes on the overlay tree periodically, and each host uses probabilistic transformation to optimize the overlay tree as a whole. In hybrid

multicast we do not consider dynamic tree reconstruction. However we use the round trip time as a metric in the tree construction, thus our proposal constructs a low delivery latency tree to some extent.

ZIGZAG and NICE [40] use a hierarchical cluster-based approach to construct overlay trees. Both of them use cluster leaders to manage the clustered overlay structure. The ZIGZAG protocol avoids network bottlenecks and keeps end-to-end delay lower. The hierarchy of the NICE protocol is for scalability to large groups.

There is a method [28] using both delay and bandwidth as a metric, which places more emphasis on bandwidth and less on delay. The scheme selects some low delay nodes first, and selects the node in those so that the bandwidth can be used most efficient. Our proposal does not use bandwidth as a metric in tree construction but our main purpose is to construct backup routes proactively with low overhead.

The problem caused by node failures in overlay multicast has been recognized in more recent work. Peercast uses a reactive approach to deal with node leaves or failures in overlay multicast. It finds appropriate places in the subtree of the grandparent or the root for the affected nodes after failure happens. The time to find an appropriate place may be long and those affected nodes may even compete with each other to connect to other nodes. Our proposal differs in that each node has its backup route before node departures, so the time to find its appropriate place after

node departure can be reduced. We will describe the difference between a reactive approach and a proactive approach in details in next section.

4.6.2 Reactive Approach

Most of these ALM methods employ a reactive approach, in which tree recovery is initiated after node departure. In this reactive approach, a node which leaves the overlay tree sends a message to inform other nodes to be affected by its leaving such as its parent and children. Affected nodes cannot receive a data temporally until they connect to a new parent node. When a node suddenly fails, it cannot send a message to affected nodes, and they will not notice the failure for a while. Heartbeat mechanism helps the affected node to notice the failure. The parent and children nodes send a heartbeat packet to each other periodically.

When the children nodes fail to receive heartbeat packets from the parent node over a period of time, the children nodes consider the parent node as a failure. However, the children nodes need a timeout period to recognize the failure. They cannot receive data flow during that time. Peercast proposes several recovery processes after a node departure, as listed below.

Root

When a node leaves the tree or fails, each of its children tries connecting to the root. The sub tree rooted at each of its children is maintained. Only children of the departed node rejoin the root. The root will try to accommodate them. The root accepts them as long as its degree does not become its max. If the degree of the root is exhausted, the root will redirect some or all of them to its descendant. This redirection algorithm is also used in other recovery processes.

Root – All

When a node leaves the tree or fails, all its descendants connect to root.

Grandfather

Like the Root, when a node leaves, the children of the departure node contact the notified grandfather. When a node fails, the children contact the root node because the children cannot receive a message about their grandfather from their parent.

Grandfather – All

When a node leaves the tree or fails, all its descendants contact the grandfather.

In these methods, it has been shown that the grandfather approach is most efficient. We therefore choose Peercast algorithm with the grandfather process as a comparison to our proposal. The main task here is reconstructing the tree by finding a new parent for each affected child as fast as possible when node departure happens However, especially in the node failure phase, it takes a long time to find a new parent because each affected node connects to its new parent by contacting the root in the tree, and the root might be quite far from the affected node. Furthermore, when the degree of upper layer nodes of the tree is exhausted, the redirection operation has to be repeated and might reach the node located at the lowest layer. In addition to taking a long time, redirection generates extra packets. If the number of children of a departed node is large, obviously the grandfather will not be able to accept all the children, so redirections will happen. Therefore, it is inevitable that it takes a lot of time to find a new parent in the reactive approach.

In a proactive approach, each host has a backup route to recover from the parent departure. Once a node departure happens, affected nodes connect to their backup route node, thus affected nodes can receive data flow after lower interruption time than that of the reactive approach.

PRM proposed a proactive approach with randomized forwarding in reconstructing an overlay tree when nodes departure happens. In PRM, each overlay node chooses a constant number of other overlay nodes at random and forwards data to each of them with a low probability. Randomized forwarding seems to be effective in some

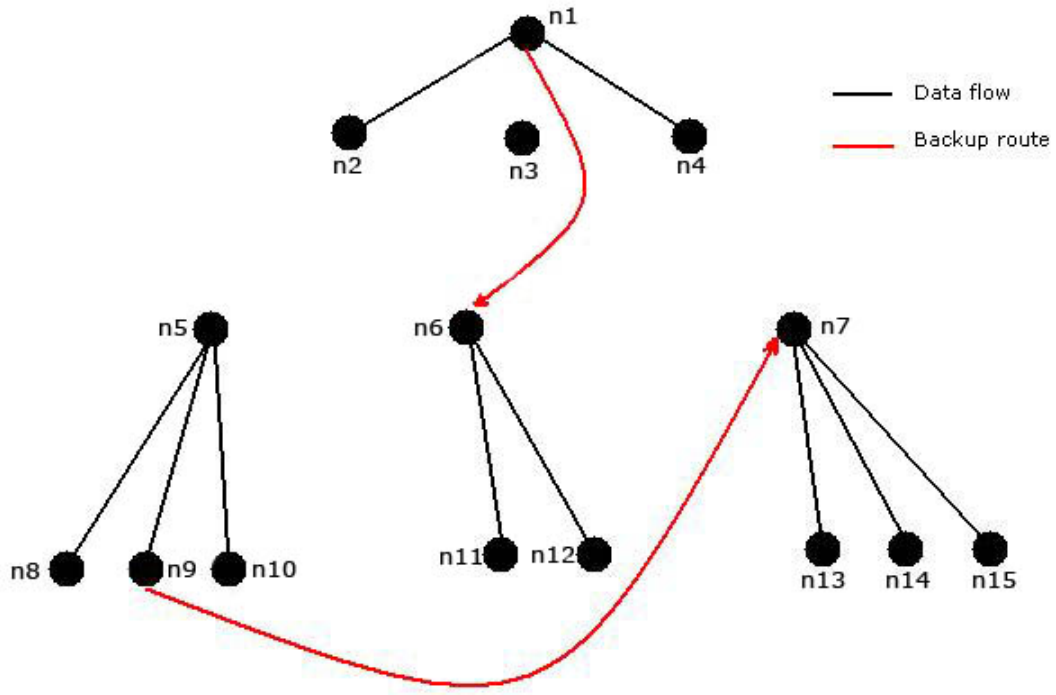
situations, but this scheme may generate some overhead traffic to send packets at random constantly. In proactive approach, each non-leaf host calculates a backup parent for its children. A backup route is ensured by using residual degree of nodes in the overlay tree. Each host uses (1) to figure out if all its children can form a backup route.

$$\sum_{j=0}^{n-1} d(Cj) >= n-1 \qquad (1)$$

A node in multicast session has n children $\{C_0, C_1, \ldots, C_{n-1}\}$. $d(C_j)$ is the residual degree of the child C_j.

First, a parent node calculates residual degrees of the children. If the total residual degree of the children is not less than n-1, all its children can form their backup routes. If not, the children cannot. In this case, the parent node calculates the total residual degree including the residual degree of descendants of the children. Second, the parent node selects the child that has the smallest latency from the grandparent to it. The selected child holds the backup route to the grandparent. The subtree of the child which holds the backup route supplies a backup route to the other children. Then, the descendants of the child and the child measure the latencies to the other children, and the smallest edge is selected. This operation is repeated until all children hold their backup routes.

Figure 4-1. Finding Backup Route



In the figure 4-1, the outline of the proactive algorithm is shown. We can see graphically how to form backup route. The parent node is n3 and children are n5, n6, n7. The maximum degree of each host is 3. the sum of residual degrees of them is less than (n-1), where n=3. So the total residual degree is less than 2. They can not form a backup route among them. Then node n3 finds the descendant of its children to make the total residual degree larger than or equal to 2. When the total residual degree of the children and the grandchildren become larger or equal to 2, the children can form their backbone route. In Figure 4-1 node 6 has the smallest latency to grandparent and holds the backup route to node n1. Node n6, n11 and n12 measure their latencies to

node n5 and n7. Based on the measurement, node n5 holds the backup route to node n6 and node n7 holds backup route to node n9. In this case, the backup routes of the children are formed by using the residual degrees of the children and grandchildren. However, searching the residual degrees doesn't always finish in the children and grandchildren. When this operation continues in the lower layer, it seems to generate many packets.

As mentioned above, the reactive approach takes a lot of time to recover from node departures, and the previous proactive approaches generate extra packets. Therefore, for Hybrid Multicast we come up with a proactive approach which suppresses extra packets as described in the next section.

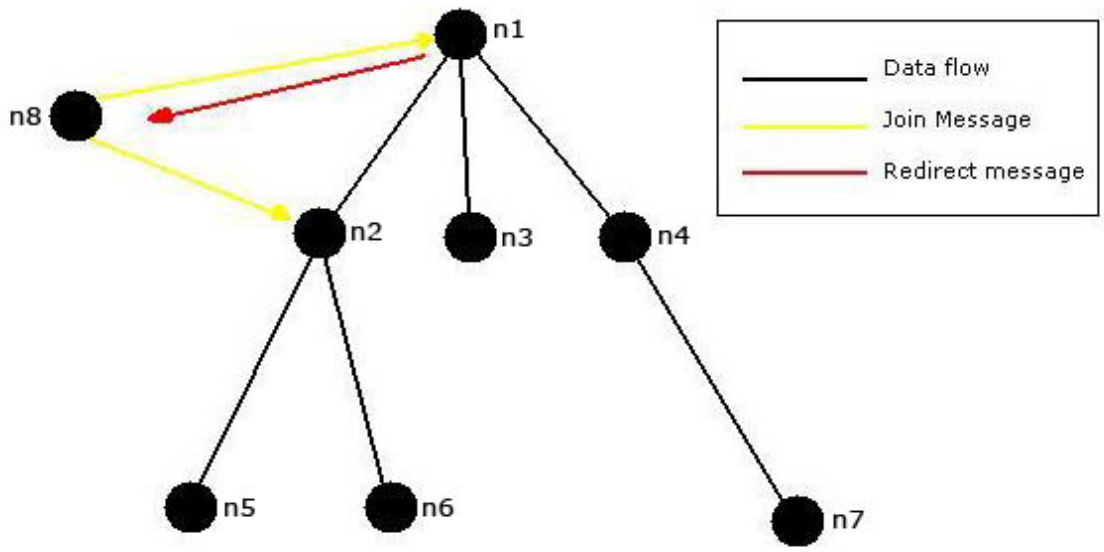
4.6.3 Proactive Route Maintenance

For Hybrid Multicast we have considered previous proactive approach. Here we explore a new approach that speeds up performance and reduces overhead time. Later in the simulations we will show the comparison between ALM and HM.

We construct overlay link between IPM and ALM (or ALM to ALM) without each host exhausting its degree. Each host constantly has residual degrees not less than 1. The children of each node can ensure their backup route between the grandparent and them by using their residual degree. This simplifies backup route calculation and contributes to overhead reduction.

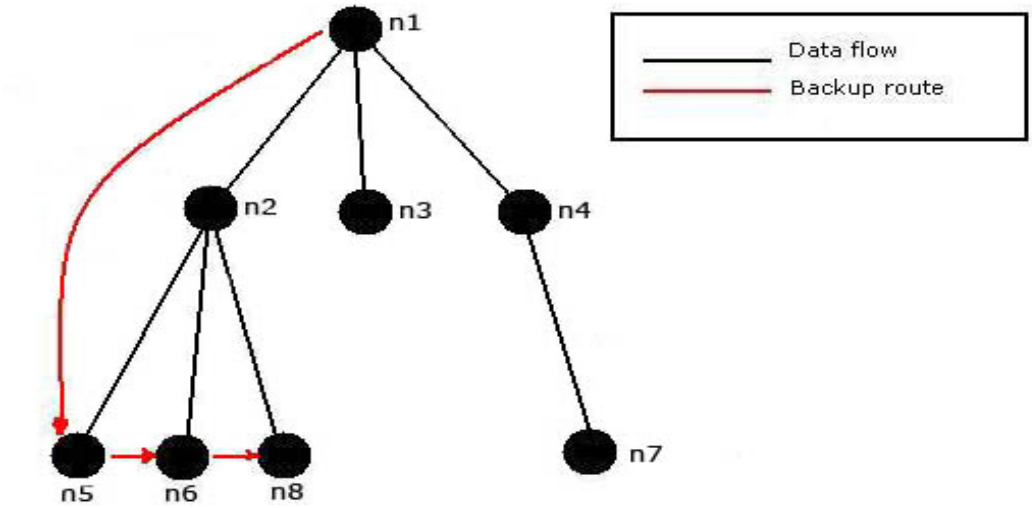
We can see the process of node joining the hybrid tree in Figure 4-2. It is assumed that maximum degree of each node is equal to x (x is a constant number in Hybrid Multicast which equals to 5). We then limit the active degree of each node to x-1 and reserve 1 degree for backup route maintenance. In previous work, when new node 8 requests to connect to node 1, node 1 accepts node 8 to join as its child, because its degree is not exhausted. However, in HM, node 1 refuses the request because the residual degree of node 1 is only 1. Node 8 sends a join request to node 2 after receiving a redirect message from node 1. As a result node 8 becomes a child of node 2.

Figure 4-2. New Node Participation



Next, we show how to decide the backup route of each node in Figure 3. When node 8 joins the hybrid tree and becomes a child of node 2, node 2 updates its children list. Node 2 sends the children list to node 1. After that node 1 measures a round trip time between node 1 and each node written on the list, and ranks the nodes in ascending order. Lastly node 1 informs them of their backup route. A node having the smallest round trip time holds a backup route to the grandparent. The second node has a backup route to the smallest RTT node, and the third node has a backup route to the second node. A node other than the smallest RTT node has the backup route to the next smaller RTT node than itself. In Figure 4-3, if the ascending order of the nodes in round trip time is node 5, 6, 8, the smallest RTT node 5 has the backup route to node 1. The second node 6 has the backup route to node 5. The largest RTT node 8 has the backup route to the second node 6. In a specific case, if the children list of node 2 includes node 8 only (i.e. no other children exist), node 2 immediately informs node 8 that node 1 is a backup parent of node 8.

Figure 4-3. Finding Backup Route



The methods described above work well for most overlay multicast applications, but hybrid multicast is a little different and has some specific requirements. First we need to realize that at some portion of the hybrid tree there might be IP Multicast, and we can not apply the above described algorithm to IP level. So this algorithm of root-grandparent-child idea works at point where IP multicast connects to overlay link or overlay links are connected to each other. Let's take a look at a sample network topology and we let one node die.

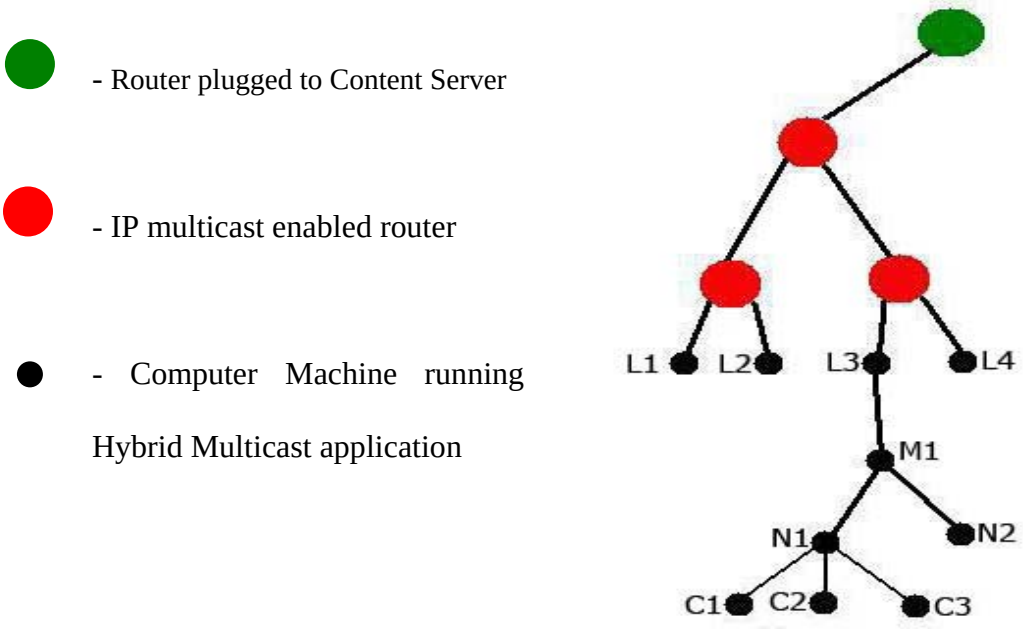
4.7 Delay Reduction on Hybrid Links

First we have IP multicast tree, where multicast routers are involved. At the end of this tree we have "leaf nodes". The basic idea with hybrid multicast is to extend these leafs by running application layer tree. Whenever a new member joins the multicast group it becomes a Leaf-Node. Note that Leaf-Node and Child Node are not the same, we'll see the difference. Leaf-Node is the one who doesn't have children (or child), and Child node is the one that has parent. Both leaf-node and a child node can extend the multicast group and possibly become Parent-Node. Child node might reject join request if it is serving enough children nodes and can not take more workload. As we see in Figure 4-3 the method of solving delay problem in multicast is to divide the tree into different layers. Nodes, standing at different degree are represented with different symbols. Any intermediate node (Parent Node) should be ready to provide help whenever it's needed.

The topology is shown in Figure 4-4. Lets see what would happen if M1 had left. M1 is in the middle of *leaf nodes* and N1 and N2, which means M1 has parent and children. In case of sudden failure of M1, grandparents have to take over grandchildren nodes (N1, N2). If M1 leaves N1 and N2 will switch to L3 immediately. To make this idea work, in terms of implementation we have to classify nodes. So we have:

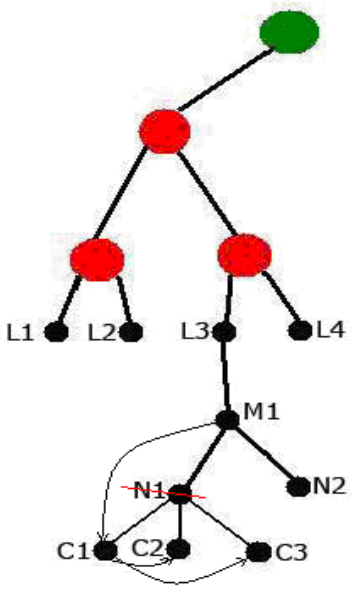
- Leaf Node
- Parent Node
- Child Node

Figure 4-4. (a) Delay Reduction



Child Node doesn't have to care about providing help. Parent node needs to provide grandparent information to Children.

Figure 4-4 (b). Parent/Intermediate Node Failure



4.8 Reconstruction of Redundant Tree

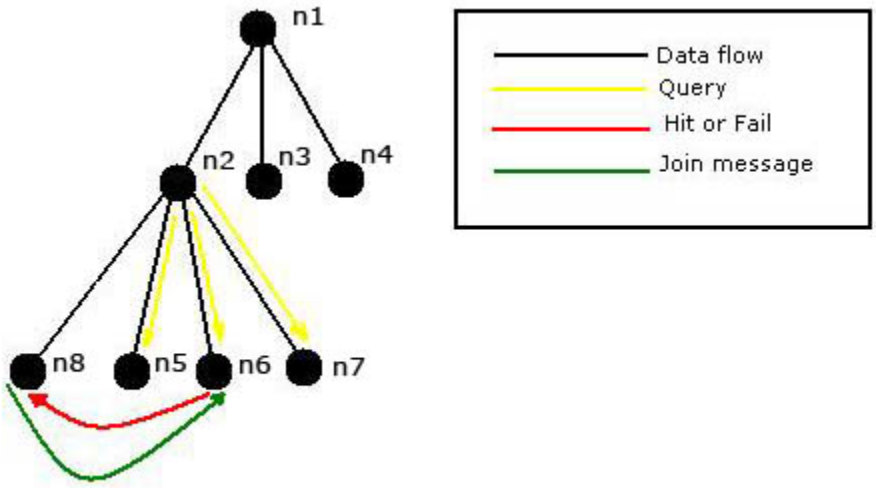
This backup route calculation is carried out whenever a node joins, leaves and fails. When a node leaves the overlay tree, the backup route is immediately applied and the new backup route calculation is initiated. Note that the backup route calculation is required only at the children layer of the departure node. It never goes down to calculate in the lower layers dissimilar to the previous approach. In some rare cases, a node cannot use its backup route. When the current parent and backup parent node

leave or fail at the same time, the node cannot connect to a new node immediately. Another case is that a node is not informed of its backup parent node. This happens when the parent node leaves the tree without noticing the node of its backup parent node before the backup route calculation is finished. In “Proactive approach to reconstruct overlay multicast trees” handling these cases is shown. In this case, it uses the ancestor-list, which contains node information from grandparent to root. Our approach also uses the same method in such cases. In the method, when a node connects its backup parent node and the backup parent node does not reply, it uses the ancestor list. First, it ordinarily joins the grandparent and it follows the redirection algorithm whether the grandparent accepts the node or not. When the grandparent does not exist because the grandparent has left or failed at the same time as the parent has, the node tries to connect to a node in higher layers of the ancestor list.

Backup routes created in the redundant overlay tree are certainly efficient as long as each host does not exhaust its degree. However it is possible that a host exhausts its degree by accepting a node rejoining in the backup route procedure. When this happens, a tree reconstruction procedure is invoked by the host itself in order to keep the route redundancy. This procedure is carried out by asking the children of a backup route node except the newly connected node whether their degree is exhausted. At the time the newly connected node finds that a certain node has residual degree, the newly connected node moves to the node that has the most residual degree. We show the procedure in Figure 4-5. Node 2 uses up its degree because node 8 joined node 2 as its backup route. Node 2 sends a query to other children, which are nodes 5, 6 and

7, and they reply hit or fail messages to node 8. The hit message means it can accept join. The fail message means it cannot accept. Node 8 moves to the node which has sent the hit message first. In Figure 4-5, node 6 sends a hit message to node 8, and node 8 joins node 6. If all messages of the children are failed, the newly connected node joins the node which it has received a message first from. It receives a redirection message from the first node.

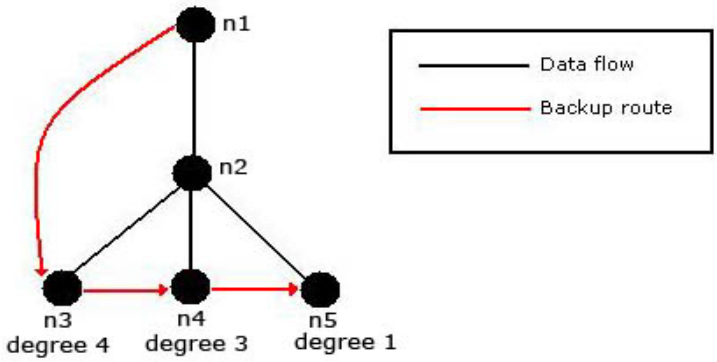
Figure 4-5. Reconstruction of Redundant Tree



One question in our proposal is that there are nodes whose maximum degrees are zero or one. Existence of nodes with zero degree (receiving only) is a common problem in ALM. Nothing could be done but they are treated as a leaf node in the overlay tree. This is similar to the case described above with different layers. Handling of the nodes which have one maximum degree is a specific problem in Hybrid Multicast,

because we construct the redundant hybrid tree by forcing reserved one degree in each node. A node of one degree can not have a child node. In the case that the maximum degrees of all children of a node are one, our proposal cannot construct a sub tree rooted at the children, so the tree can not be constructed effectively. In the worst case that the maximum degrees of all children of the root node are one, our proposal can not construct the tree any more. To avoid this case, we allow the nodes of one maximum degree to have a child although their degree is one. Another problem is that they cannot provide backup routes because of exhausting their one maximum degree or zero maximum degree. We then decide that each node can have only one node whose maximum degree is one or zero, and place the node at the end of the backup spanning tree so that the node need not provide a backup route. We show this case in Figure 4-6. Node 2 is a parent of three children, which are node 3, 4, and 5. The maximum degree of node 5 is one. We place node 5 at the end of the spanning tree of the backup routes, and node 5 needs not provide a backup route to other nodes. Finally, all the children nodes can get their backup routes.

Figure 4-6. Treating Leaf Nodes Having No Children



The process of building redundant trees is good only for ALM. Our protocol (Hybrid Multicast) however involves not only Overlay networks, but also IP Multicast; therefore we a need solution for those nodes, which receive data through IP Multicast. We all know that IP Multicast provides no redundant trees no retransmission of lost packets.

In Hybrid Multicast we have the whole infrastructure of Hybrid Network (combination of different types of networks). We should definitely take advantage of this. IP layer has less knowledge; IP Multicast doesn't have any retransmission or packet recovery. The problem is that IP Multicast packet is encapsulated with IP Multicast group address, and if one of the end nodes doesn't receive packet we either would have to retransmit packet with unicast, or some other members have to help this particular node.

The way we deal with packet recovery is to push everything to the application layer. The main problem again is changes at router, which we can not afford. So the unicast comes for help again.

One thing we should realize here is that we address IP multicast packet retransmission at local segment. If we recall the idea of deploying IP Multicast, we'll see that HM consists of number of networks; Overlay network, Token Ring, IP Multicast enabled LAN, etc… if there is IP Multicast enabled LAN, then all

retransmissions or packet recoveries would happen at this segment only. Server is some kind of special case here. (IP Multicast) Server receives data as unicast, then encapsulates packets as multicast and distributes among local group members. Packet retransmission on Overlay is not very interesting since we already have such algorithms; we are more concerned about IP Multicast. In HM even for IP multicast, packet retransmission will take place at the application layer.

As we mentioned earlier we have so called Hybrid Multicast tree and Rendezvous Point. RP is more of a virtual machine and is responsible for the negotiation part between IP and Application layers. We want RP to have all information about the Multicast group. Whenever a computer joins the IP Multicast group, all other members are notified. This can be easily done and the message contains only the ID of the new member and its IP address. After that every member will ping the newly joined node and in case of connection failure this node could become a potential backup node. Whenever a new node joins the IP Multicast group server will update the table at Rendezvous Point (table contains just ID and address of multicast group members). RP will send out this information to all other members, each member of the multicast group will ping the newly joined node and if the delay between these nodes is smaller than the current one then this node will become prioritized. Of course we consider properties of a tree. We want nodes to find backup route at upper or same level of the tree hierarchy. Certainly nodes at the same level are closer than those higher (This is not necessarily so), so for help we will first ask those nodes that are closer, and if we don’t find any help, we contact grandparent.

4.9 Backup Routes at IP Multicast Segment

One good thing about IP Multicast (IPM) is the stability of the network architecture. Routers are deployed and they typically remain unchanged for a long period of time. In Hybrid Multicast we have a whole infrastructure of different types of multicast protocols, and combination of different types of networks. We should definitely take advantage of this. We have described algorithms that deal with tree reconstruction at the Application layer, but IP layer is different. We have less knowledge. IP Multicast currently does not have any retransmission or packet recovery protocols. The problem is that IP Multicast packet has Multicast group address, which includes list of IMP group members and if one of the end nodes doesn't receive a packet we either would have to retransmit the packet with unicast, or some other members have to help this particular node.

The way we deal with packet recovery is to push everything to the application layer. The main concern again is network infrastructure modification at the router, which we can not afford. So the unicast comes for help again.

Earlier in this thesis we provided help for overlay trees. When a node fails or leaves, each node has a backup route pre-computed, IP Multicast is different though. Our claim is that HM is a very reliable protocol, so we need to take the same care of IP Multicasting as well.

While building hybrid tree, we have two alternatives, either IP Multicast or Overlay Multicast, depending on availability. One thing we all agree here is that IP Multicasting is much better than overlay, so we use it whenever possible.

What we want here is to provide reliable backup route for IP Multicast nodes as well. As we discussed IP Multicast protocol provides no help at all for dropped packets. If a link fails IPM protocol will reconstruct the tree with the routing algorithm. That takes some time and the overhead might be significant. In the next section we provide a method in which IP Multicast segment has reliable backup.

Again, we go back to the joining process of Hybrid Multicast tree. Each node needs to find out, if IP Multicasting is available, if yes it takes advantage of that, if not it simply uses an overlay link to connect to the server.

We solve the problem by treating each IP Multicast node as a unicast node. In this case we do have IPM capability, so each node joins the local server and receives the data, however on the background, it builds an overlay edge to the hybrid multicast tree. In other words we pretend there is no IP Multicast. In case there is a problem with IP Multicast, we would switch to overlay multicast. There are two scenarios for IP Multicast link failure. Either IP M protocol reconstructs the tree back, or the network loses IP Multicast capability.

So let’s see what happens if we build hybrid edges at IP Multicast enabled segment as if not having that capability.

At IP Multicast tree link or router might fail, so some nodes may stay out of multicast group. If we take no action, what would typically happen is that disconnected nodes would try to connect to the hybrid tree via Overlay link. So why should we wait before that happens? We will build overlay link at the global hybrid tree as usual, if something happens and IP Multicast fails, this particular node will switch to overlay immediately; whenever IP Multicast tree is fixed, the node will switch back to IPM again. All we have to do is to manage the switching part at the application layer which is quite possible. If IPM tree is not reconstructed, the node remains on the overlay link as long as it takes for IPM to be up again.

The general idea behind this method is pretty predictable. We want to find backup node before something bad happens. We need to test this solution with all possible scenarios. If link fails we have several nodes disconnected from the network, and hypothetically the backup node might be in trouble as well. With hybrid multicast this problem can be easily solved, we do not even have to implement anything new.

While building the hybrid tree, we maintain the candidate nodes, the best node based on delay will be selected as the parent node. It is really hard to predict who would be the potential parent of a node, but each node stores a table for candidates. The table contains the address, ID and delay information, the table is sorted based on delay and

whenever switching to a backup route, the disconnected node will send a help message to the first node in the table, if after time-out there is no help that means this node was disconnected as well, then try the second best, then the third best and so on. If none of the nodes could be reached then the help message is sent to the local root node (local IP Multicast server) and if even that does not help, than the help message is sent to the content distributor (main server).

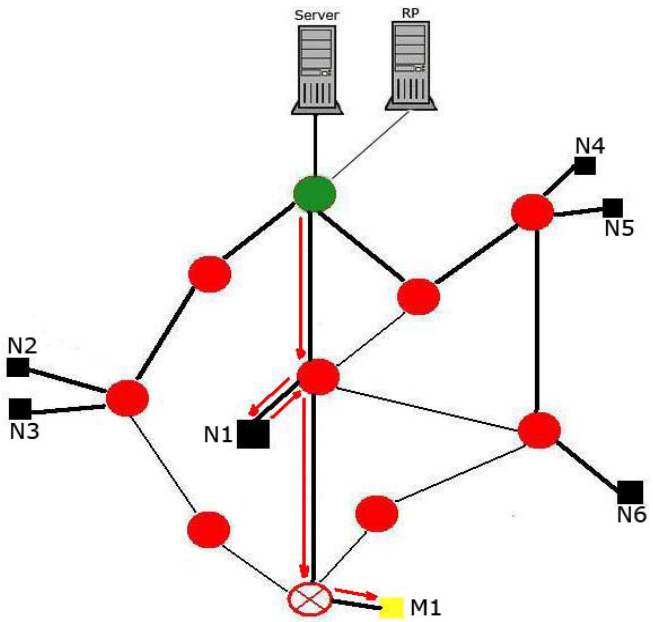
Let's run a sample scenario and see what would happen if we let each of the links fail. We have IP Multicast segment, and here we will build a hybrid tree. We consider IP Multicast, but on top of it we build a hybrid tree pretending there is no IP Multicast. Nodes are numbered in the order they join the multicast group (N1 joined first, N2 second, so on…). Once Server distributes the data through IPM it establishes unicast connection as well. The good thing about unicast is that the packet gets delivered through any available route, while IPM has predetermined route. So there is an Overlay link between server and N1, then N2 joins, depending on which node is closer, Server or N1, N2 will consider it as a parent, then N3 joins, if N2 is the closest to N3 then N2 becomes a parent of N3, and so on… in the end, N6 joins let's say through N5.

The process of building a hybrid tree considers all possible candidates, so N5 is the best parent for N6, N4 is the second, N1 is third. Now if the link between server and N1 fails, N6 will send a help message to N5, if it doesn't respond then to N4 and so

on unless is finds a helping node. If there is no node available at the local IP segment, the help message would go to the parent of the local Server.

The help message is small and each node after sending it waits a certain amount of time, e.g., around 10 ms. Depending on the number of neighbors, the complexity of finding a backup node could be up to *O(n)*, if we have *n* neighbors.

Figure 5-1. Hybrid Multicast with Redundant Links



One issue with IP Multicast segment is to teach Hybrid Multicast how to select IP Multicast when available and overlay when necessary (pruning process).

The nature of Hybrid Multicast allows us to approach this problem easily. Let's imagine we have a HM tree, and somewhere on the network there is a node with IP

Multicast capability that wants to join, however there is no IP multicasting between the existing HM tree and the current node. Obviously, this node will join HM group via overlay link. If it happens that another member of the same segment wants to join the HM group, the main RP will select possible candidates and provide a list to the new member.

The node, willing to join the HM group will decide based on delay, which option is best. There is a big probability, that IP Multicast works best; if yes the node will send an IP Multicast join message to another member from the same segment that joined the HM group earlier. In case of IP Multicast failure, the node will immediately switch to the overlay link that is constructed in advance. The IP Multicast protocol will try to reconstruct the path and establish a connection again. If that is done successfully, we no longer need the overlay link, so a stop message can be sent to the overlay parent. If the so called ‘leader node’ decides to leave the group, all other members at the local segment, will switch to the overlay temporarily, and one of the members, that has the minimum delay to the HM tree will establish an overlay link. All other members of the local segment will discover that there is IP Multicast availability at the local segment. IP Multicast will construct the local IP level tree and the new leader will start distributing data to other members.

We know that overlay multicast is pretty much unicasting. Each member of the overlay tree has a list of children nodes and sends unicast messages coming from the root node. What a stop message does in HM is that it removes node (sending stop

message) from the children list. So basically nodes take care of themselves in HM based on information they receive from the RP.

4.10 Passive Members of Hybrid Multicast

Another innovation we have in Hybrid Multicast is an idea about Passive-to-Active member. We have described how to build hybrid path on the network. As we all already know everything in HM is implemented at the application layer, so we have to install the software on our computers. After installing, people often leave the application running, and sometimes they do not even request to join the multicast group. We now have one special case for Hybrid Multicast users. Whenever a user starts the Hybrid Multicast software and does not request to join the multicast group, we say this is a passive member.

Now what we should do about passive members is that we make them active whenever needed, especially for those nodes attached to IP Multicast routers. Take a look at Figure 6-1.

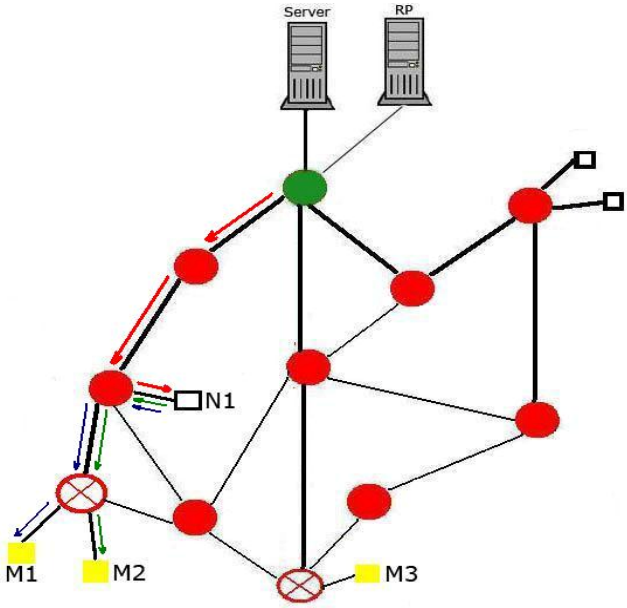
If few members want to receive media stream, Content Server can send either unicast messages or include it as a multicast group member. Figure 6-1 shows that there are two users - M1 and M2 interested in content. Sending unicast messages might consume more bandwidth. We notice that there is a computer N1 running the software, but this user has not expressed an interest in joining the multicast group. So why not make this user active? If we do that N1 will receive the data and then forward everything to M1 and M2 as unicast message (shown in Fig. 6-1). Otherwise

there would be two unicast messages going throughout the network. This would help us save some bandwidth along the path between Server and N1. All users, who are using the Hybrid Multicast software, must be aware that Rendezvous Point might use their resources as intermediate node. Therefore if you are using Hybrid Multicast be ready to help others, while others will be helping you. We call this process “One for all, all for one”. How fair the idea is is a subject of discussion and whether such method should be used or not is not discussed in this work.

We should also note that in the case of Active/Passive member we don’t need any additional software implementation. Few lines of codes could enable an inactive member and make it active. The user will never know what’s going on behind the scene. In fact, they don’t even need to know anything. This might raise some concerns, but again we leave it as a subject of discussion.

The process shown in Figure 6-1 is just one single example of such condition; we should generalize it more and consider all scenarios. However first let’s take a look at what should happen with passive member N1 that is attached to an IP Multicast enabled router.

Figure 6-1. Enabling Passive Member in Hybrid Multicast



The picture above shows the transmission process from Server to N1 that is attached to a router with IP multicast capability, and then N1 sends two unicast messages, one to M1 and the other one to M2. Two nodes here are shown only for simplicity. We could send data to M1 and M1 would forward packages to M2. That would be pure application layer (overlay) multicast. Generally that's what happens. M1 sends a join message to Server and then when M2 requests to join, M1 will be selected as a close node to extend the multicast. This two are connected to server with overlay links, and we already know overlay is not optimal.

What we typically do about passive members in Hybrid Multicast is that we treat them as active members. When M1 sends a join message N1 will be included as

active member in multicast. If it happens that N1 is selected as the best intermediate node, having smallest delay, RP notifies N1 to extend the network to M1. N1 is not necessarily involved in multicast, so before N1 actually extends the multicast tree, it sends a join message and becomes a member of the multicast group without user knowing that. Every time user runs the hybrid multicast software the heartbeat message is sent to the Rendezvous Point letting it know that this node is available to hybrid multicast and its resources could be uses as needed.

5. Comparison of HM with Other Multicast Protocols

There are some protocols that could be considered as potential competitors of Hybrid Multicast. One of the major competitors is IP Island Multicast. We are going to conceptually compare Hybrid Multicast with Island Multicast (IM) and see what the major differences are.

5.1 Hybrid Multicast vs. Island Multicast

Island Multicast is a subset of Hybrid Multicast. Hybrid Multicast is more general and addresses lots of issues which are not discussed in Island Multicast. The basic idea behind Island multicast is to bridge IP Multicast ‘islands’ to each other through overlay tunnel. IM uses application-level multicast protocol to build island overlay. So the islands are local IP Multicast enabled networks and such islands are interconnected. IM has so called leader node which is responsible for distributing data among the local group members and serves as a bridge node as well. The leader election process is a little bit awkward; the best candidate is the one that has the lexically smallest IP address. We are going to describe the process of Island Multicast leader election.

When a host discovers that there is no leader, or it has failed or is leaving, it would assume itself to be the new leader by sending, after a random timeout, a heartbeat message to all group members. If the host receives a heartbeat message, it suppresses

its heartbeat message. In this way, the first node that sends the heartbeat message becomes the new leader. The host with the lexically smallest IP address is chosen as the final leader. The new leader then becomes the bridge node to all neighbor islands and joins this group with all other islands.

Although Island Multicast has a bridge improvement algorithm, that tries to reduce the number of hops, initially the leader node is the bridge node. The process of ‘improving bridge’ takes place after selecting a leader, while Hybrid Multicast selects the so called ‘bridge’ node from the beginning. When a node sends a join message, Hybrid Multicast RP will consider the delay information and build a hybrid path. Island Multicast assumes that all nodes have access to a network coordinates, and based on that Island Multicast can reduce the number of hops. First of all reducing the number of hops might not be optimal, that works only in cases when all nodes have the same bandwidth and same delay, which doesn’t always apply to physical links. In Hybrid Multicast we do not assume that we have either network coordinates or physical topology of the network. If we could afford that, Hybrid Multicast would build a hybrid tree with the single-source-shortest-path algorithm, which is more accurate; however we can’t afford the knowledge of the whole internet topology.

Another important feature Hybrid Multicast has and Island Multicast doesn’t is backup routing. IM builds overlay link between IP Multicast islands and if that link fails there is no alternative path. IM has also no algorithm that protects each island

when bridge nodes leave. Even in case of leader failure the selection process gets executed afterwards which of course is undesired delay.

The way we construct Hybrid Multicast is gathering delay information. Each node has this information to each candidate node; the one that has smallest delay is selected as best candidate. If parent node fails/leaves child node switches to second best candidate, thus all redirections are calculated in advance so all overheads are minimized.

Moreover Hybrid Multicast has backup routes not only for overlay trees, but also IP Multicast. It's true we can not build backup route at IP layer, but HM still provides backup for IP Multicast nodes. This issue is not addressed by Island Multicast at all. In general Hybrid Multicast contains IP Island, with other words IM is subset of HM. IM simply connects IP enabled multicast islands, but in internet there are lots of networks that don't have IP Multicast capability and we have to provide some kind of multicast support. In Hybrid Multicast we can not always afford knowledge of physical network architecture, while IM requires us to have such knowledge.

Another important feature Hybrid Multicast has but Island Multicast doesn't is backup routing. IM builds an overlay link between IP Multicast islands and if that link fails there is no alternative path. IM although has algorithm that protects each island when the bridge nodes leave, the re-selection process gets executed afterwards, which introduces undesired delay. In Hybrid Multicast, all redirections are calculated

in advance so the delay is minimized. Moreover Hybrid Multicast has backup routes not only for the overlay trees, but also for IP Multicast. This issue is not addressed by Island Multicast at all.

HM combines not only IP Multicast enabled networks, but also all other networks that do not have IP Multicast capabilities. In Hybrid Multicast we can not afford the knowledge of the physical network architecture, while IM assumes we have such knowledge.

Another weak point of Island Multicast is the usage of the Minimum Spanning Tree (MST) algorithm instead of the single-source-shortest-path (SSSP) algorithm. Mathematically it can be proven that for routing protocols, SSSP is preferred over MST. The way IM joins islands is to run the MST algorithm among the leader nodes, while hybrid multicast assumes that the whole network is one unit and each node is treated as a member of the hybrid tree, not a member of separate islands. If we treat each node as an island, IM would have to rerun MST algorithm every time node leaves. This is a huge drawback for dynamic network environment.

5.2 Hybrid Multicast vs. Other Hybrid Approaches

Island Multicast however is not the only competitor. There are some other hybrid approaches and we are going to look into some technical details of those approaches. There are some fundamental differences between HM and other approaches. They are

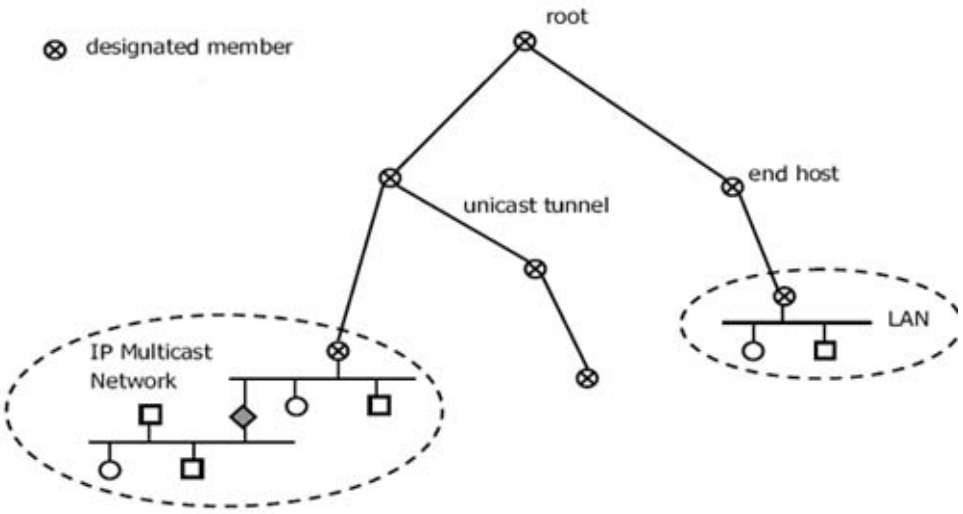
called in many different ways, like Proxied Overlay Multicast, Unicast-Multicast, Gateway-Multicast, etc…

First we are going to talk about similarities, and for simplicity we call it hybrid approach. The goal for hybrid approach is pretty much same as for HM, devise multicast framework that addresses limitations of native multicast. By leveraging application layer and native techniques we can address the problem. As you see the problem set for hybrid approach is similar to hybrid multicast, and they both share the name ‘hybrid’. Combination of different types of layers is the hybrid approach. So far hybrid approach has been similar to hybrid multicast, but the way hybrid approach resolves the communication between application and IP layers is different.

The basic idea of hybrid approach is to construct a backbone overlay by deploying some special intermediate nodes, and as we can see here that is a big difference between hybrid multicast and hybrid approach. HM has all its implementation at the application layer, and all we need for HM is a software application. Sometimes it is infeasible to deploy intermediate node on the Internet. If we could do that we would place IP Multicast enabled router and the problem would be resolved. However hybrid multicast is needed because we can not afford such replacements and we want to use IP Multicasting whenever it is possible.

Let’s take a look at some technical details of hybrid approach. Hybrid approach has proxied overlay multicast. Proxies create multicast trees among themselves and end hosts communicate with proxies via unicast or native multicast.

Figure 6-2. Hybrid Approach



As we can see in Figure 6-2, Hybrid Approach is pretty close to Island Multicast. We have a number of islands and at each island we have a designated node (computer machine, network device, etc…). Hybrid multicast is different in terms of building the hybrid path. If we recall the process of building tree in hybrid multicast, we will see that HM has a tree that might involve either IP layer, or application layer, depending on the availability. HM always prefers IP Multicast, but if there is no support of IPM it uses ALM.

The goals and problem set for both Hybrid Multicast and hybrid approach are the same, but the way HM resolves the problem is different.

A hybrid approach enables interoperability of different multicast protocols as well as HM, depending on what is available in a given region of the network; however hybrid approach requires at each region the existence of one or more hybrid gateway (GW) nodes. The GW converts incoming ALM data to NM transport, or vice versa.

This solution is ok, but if we assume that the internet contains lots of networks, and the world is pretty big, we can not always allow the deployment of gateway nodes. It might be possible within some region, but the internet has no boundaries and user in Australia might request to join a multicast group of which the content distributor is physically located somewhere in Europe.

The good thing about Hybrid Multicast is that it doesn't require any deployment of any kinds of network devices, or designated nodes. All that's needed is a software application which is pretty easy to download. Hybrid multicast doesn't treat networks as islands, the whole multicast group is one unit and all nodes are members of hybrid multicast tree. Whenever hybrid multicast extends its edges, delay information is considered and the path between the new member and intermediate node has minimum delay, while designated nodes are fixed in the network and the link between island gateway nodes might not be the most optimal one.

6. Simulation and Performance Evaluation

We evaluate the performance of Hybrid Multicast (HM) using simulation. For the purpose of simulation, we used NS2 [34] and Georgia Tech's internet topology generator [33]. We are mainly interested in reliability, resilience and delay reduction in HM. HM in comparison with ALM or Island Multicast performs better. We do not compare HM with other approaches that use gateway and network coordinate service because these services are not always available. Hybrid Multicast can be used everywhere with the internet connection, and deployment of additional hardware is not necessary. We conducted simulations to evaluate and compare performance of Hybrid Multicast against Application Layer multicast. The quality of the multicast protocol is judged based on the following metrics:

Tree Cost: the cost of a tree is the sum of delays on the tree link latencies. It is convenient though somewhat simplified metric to capture total network resource consumption of a tree. The ratio of a tree's cost to that of a corresponding shortest path source tree SPST is the tree's *cost ratio*.

Delay penalty measures how much the overlay stretches end-to-end latency. *Relative Delay Penalty* (RDP) is the ratio of the latency between a node pair on the overlay to the latency between them on the physical network. *ARDP* is the average RDP over all node pairs. The smaller the delay penalty, the closer node-pair latencies on the overlay are to latencies on the physical network

6.1 Performance Comparison

The network configuration under NS2 is as follows. Results presented are based on simulations on a network consisting of 1000 nodes, representing routers and 3500 links. We tried to make setup of the simulation realistic therefore the network has been segmented so that multicast enabled routers are clustered close to each other. Using Georgia Tech network topology generator we create two networks with 200 nodes, having IP Multicast capability, each of them has attached agent to it. Later we create another network with 600 nodes, with unicast-only capability. We limit the number of children nodes; maximum node degree constraint is set to ten when running HM. These three networks are interconnected with 10 designated members, each inter domain router has redundant links to other networks. Data points on the graph represent the average over 100 runs.

Figure 7-1 shows the simulation results from a Hybrid Multicast run, to construct a hybrid tree. We have 300 members randomly chosen from the existing 1000 nodes. One of them is Server, generating the data for the group members. Rendezvous point is a separate agent, designated to manage tree building algorithm. All nodes on the network have Hybrid Multicast Rendezvous Point (HMRP) address and join requests go straight to HMRP. Members start joining the hybrid multicast group one by one on first minute of simulated time. The tree cost increases initially since we have lots of joining requests coming in. 100 of them are using IP-Multicast to receive the data on the hybrid tree, and 200 respectively receive the multicast data via overlay links.

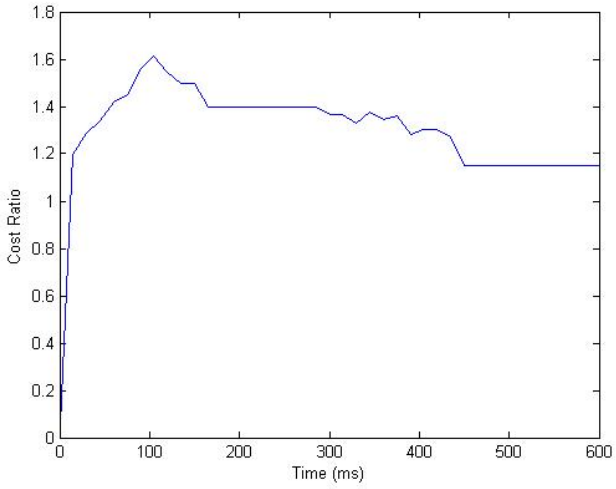


Figure 7-1. HM Tree Cost Ratio

Figure 7-1 shows the result from a typical run to construct an end-host hybrid multicast tree with 300 members in total. Members join the group one by one in the first minute of simulated time that explains the sharp increase of the tree cost ratio (on *y* axis). On the following second and third minutes the tree stabilizes, the data flow goes on hybrid links and no join requests are sent to RP. The tree cost ratio is relatively low at peak (about 1.6 times of SPST), we attribute this to the increased number of group members, which help newcomers find the closest parent. After 5 minutes we start selecting end-hosts and let them leave the group. In the simulated scenario 100 members leave the hybrid multicast group, departing members are chosen randomly. After hundred departures it takes less than minute for remaining members to settle upon another stable hybrid tree and the tree cost becomes less. We also monitor the behavior of new parent selection process. The simulation lasted for 10 minutes.

Most of our comparisons are done between HM and ALM. We implemented a competitive application layer multicast, to make sure the comparisons are fair. Our ALM implementation is very close to ALMI [60]. For the multicast tree generation we use the Djikstra's algorithm, which is also called single source shortest path (SSSP) algorithm. Rendezvous Point (RP) of ALM has the information of all the overlay nodes on the network; the delay information is measured via 'ping' messages. Our ALM implementation is center based and the multicast tree construction is done only at RP. For the recovery time, later in Section 6.2 we use the reactive approach, referred to as *grandparent-all - root-all*. The maximum number of children nodes in ALM, just like in HM is 10. When the node leaves/fails all children nodes are redirected to the grandparent; if grandparent can only take only take care of the failed nodes as the number of children nodes has reached the maximum degree, the remaining children nodes send join request to RP. RP removes any node from the table that does not send a heartbeat message; the heartbeat message includes only ID of the sending node. When the join request is received, RP starts executing the SSSP algorithm until the joining node is reached. As soon as the path from the server to the joining node is found the node is connected to the overlay tree and starts receiving multicast data. In the case of node leaving, each node sends STOP message to RP, RP removes this particular node from the table and runs the SSSP algorithm again for the remaining children nodes.

6.1.1. Tree Cost Ratio

Next, we compare Hybrid Multicast with the Application Layer Multicast. The only difference between these two scenarios is that HM uses resources of IP Multicast whenever available, while ALM generates multiple unicast streams all the time. We experimented this scenario with 500 members. Initially we let 50 nodes join the multicast group, and one of them is HM source, which distributes the data among all other group members. Rendezvous Point is running on a separate agent and we calculate the tree cost of a resulting tree.

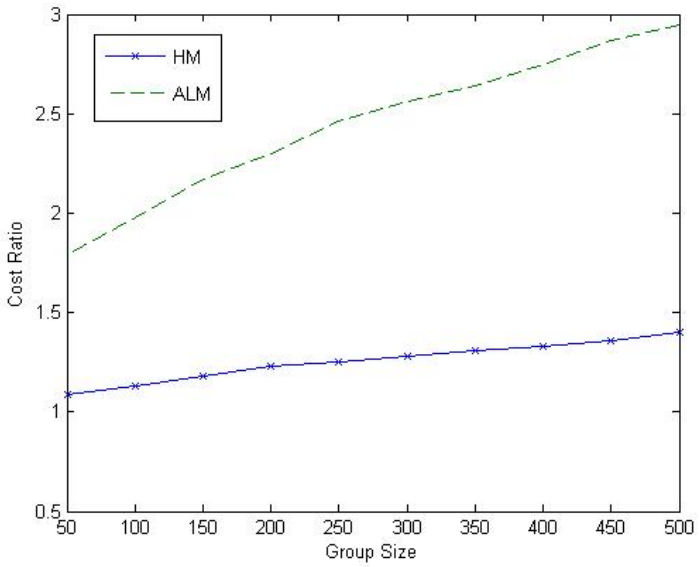


Figure 7-2. Cost Ratio of HM and ALM

As expected ALM tree cost ratio is larger than HM, which increases with the larger group size (Fig. 7-2). HM's cost ratio is greater than 1 for HM and increases very

slowly with increased group size. We attribute this to effective use of IP Multicast resources. In the simulation we implemented tree improvement algorithm. Suppose an end-host joins hybrid multicast with IP-multicast capability, this node becomes designated node for transitioning unicast messages into the multicast, or vice versa. When other nodes with IP-multicast capability request join, they will receive the data at IP layer through earliest joined node. The benefit we gain here is the efficiency of IP links, and reduced overhead for packet processing. Packets now are processed at the IP layer instead of the Application layer. Apparent from the figure is that the tree cost of HM increases slowly even when the group size becomes large. For the tree cost ratio calculation we account the cost of all intermediate links on the hybrid tree. For our simulation results we collect the data for Hybrid Multicast (HM) and Application Layer Multicast (ALM).

The difference in their tree cost lies in the overhead attributed to routing on overlay topology. As we see in Fig. 7-2, HM value if fairly low than pure Application Layer Multicast, as we said this is due to the fact that ALM sends many duplicate packets in the internal network. Increased group size increases the network load therefore the tree cost becomes significantly larger. If source is far from the receiver the packets incur long delay along the path, we call this wide distribution. HM has better approach of solving this problem than ALM. If network segment has IP Multicast capability, and one of the local segment members joins the hybrid multicast tree, all the other nodes from the same segment can receive the data at IP layer. This incurs smaller delay and is more efficient than having multiple unicast transmissions.

6.1.2 Tree Delay

Multicasting protocols, such as HM, ALM incur penalty on end-to-end delay. We collected results for the large absolute delay, which is reflected in *Relative Delay Penalty* (RDP). Figure 7-3 shows the RDP from our simulations, as we increase the group size.

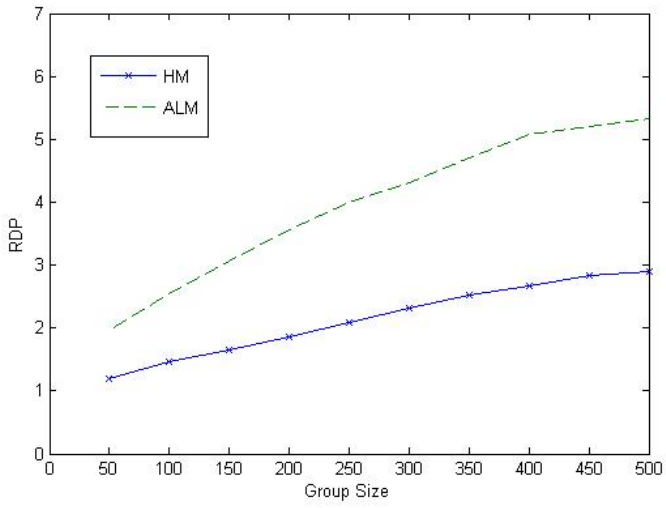


Figure 7-3. Tree Delay of HM and ALM vs. Group Size

With the increase of the group size, RDP increases in the graph. We calculate delay penalty based on the delay from the source to destination, including the last hop. For this simulation we used 500 nodes, initially we start with 50 nodes only, and within time interval 50 other members join the group unless it reaches 500 members. 200 of them are IP Multicast enabled, and 300 unicast-only. Network has 10 inter domain routers that connect these networks with redundant links. Totally the simulation lasts for 10 minutes; data points on the graph represent the average value of 100 runs. The

delay value of the intermediate links is often bigger for ALM than for HM, since ALM sends data over overlay links, which includes underlying physical links. The actual topology for ALM is always hidden. For HM intermediate hops can be end-hosts and routers, depending on the availability.

Increased tree cost and end-to-end delay is expected for all the end-host based multicast delivery trees. Delay results presented in the graph here include the delays between server and all members in a multicast group. In reality obviously not all members of multicast group will be senders, it's just root node. The worst case scenario would be the sender and the receiver having distance equal to the network diameter (two furthers members). We want to evaluate these multicasting protocols under fair conditions, so root is not located in the middle of the network, it is randomly deployed.

6.2 Recovery Time

We implement proactive approach in Hybrid Multicast for the alternative parent selection. We carried out the simulation results by NS2 (ns-2.30). Our simulation topology is different from what we had in previous two sections (*Tree Cost Ratio* and *Relative Delay Penalty* measurements). Our topology has 50 routers, five of them are domain to domain and all other nodes have attached end-hosts to it. The topology is generated using Georgia Tech Internet Topology Generator, available under NS2 (/ns -2.31/gt-itm).The bandwidth between domain routers is 100 Mbps, and 100 ms delay.

The delay between routers within a domain varies between 10 to 50 ms, and bandwidth is set to 10 Mbps. The nodes (end-hosts) are connected to 45 available routers, except the five inter domain routers. The total number of nodes varies from 20 to 100; the link latency varies from 10ms to 100ms. Initially we let all nodes join the hybrid multicast tree, just like the previous simulation environment. We have one node selected as a root, and one as a Rendezvous Point, after that we randomly pick nodes and let them leave/fail.

First we use the average recovery time as the performance measure. The recovery time is the time for an affected node to find a new parent. Each node in our simulation has a heartbeat message, that's how RP knows the node is still on the network; also if children don't receive the multicast data from their parent, they assume it has failed and start looking for a new parent. Figures 7-4 and 7-5 show the average recovery time for leave and failure in the simulations. Figure 7-6 shows the maximum recovery time for 100 nodes. The comparison is done between Hybrid Multicast and Application Layer Multicast. We don't have IP multicast enabled routers in this simulation. One reason we disable IP Multicast capability is to evaluate how much better Hybrid Multicast can perform over the ALM when all conditions are the same. The main advantage of HM is the usage of IP Multicast resources. If such resources are not available HM is expected to have similar performance as ALM. We can observe from the plotted results that although network configuration is the same for both scenarios, HM can perform still better than ALM.

In Fig. 7-4 the average recovery time is shown. We randomly select a node and let it leave the hybrid multicast group, this process is repeated 100 times and the average time in milliseconds (ms) is recorded. After that we increase the number of group members and repeat the node leaving procedure 100 times.

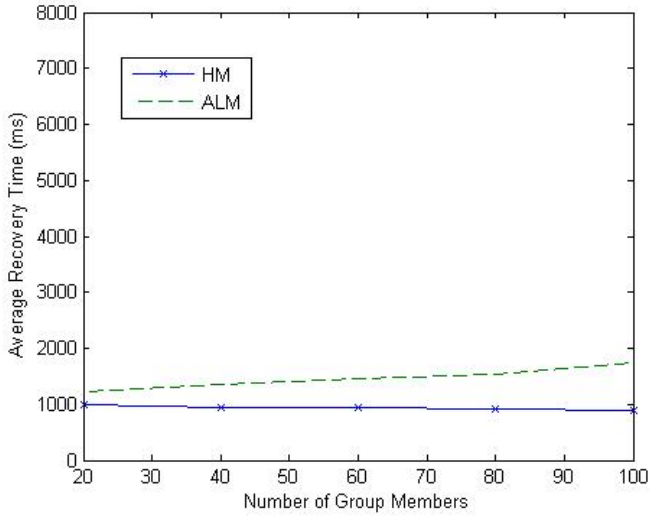


Figure 7-4. Average Recovery Time with Varying Number of Group Members in Leave

As we can observe in the figure above our proactive approach in HM is better than the ALM reactive approach used in most multicasting protocols. Reactive approach is based on a simple idea of redirecting all failed nodes either to grandparent, or root. Since we have maximum degree constraint, sometimes ALM redirects nodes to root. When node leaves the group, it sends a notification to PR so the reconstruction of tree is done. HM has already stored the alternative parents list and takes less time to select a new parent. The average recovery time for ALM increases with the number of group members; on the contrary for HM the recovery time decreases. One more

advantage of HM is that the more group members we have the easier it is to find a backup.

In Figure 7-5 we plot the simulation results in the case of node failure. The setup is similar to Fig. 7-4, the only difference is that we let nodes fail (instead of leave), and then record how long it takes for the children nodes (affected by failure) to find the new parent. It is important that when non-leaf node fails, all children nodes, depending on the failed one have to be brought back to the multicast group. We record how long it takes to recover all children nodes, affected by a parent failure.

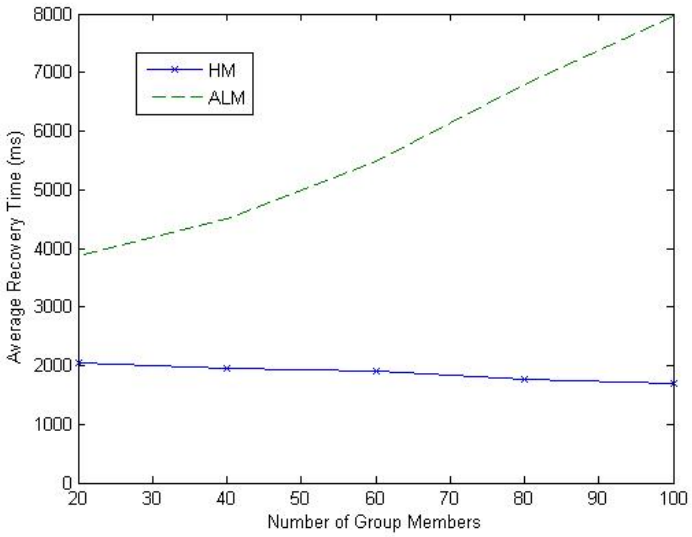


Figure 7-5. Average Recovery Time with Varying Number of Group Members in Failure

As we mentioned earlier, increased number of group members increases the average recovery time for ALM. HM has slightly larger recovery time compared to '*leave*'

scenario. The proactive approach we have implemented for HM enables affected nodes to immediately find a backup parent. On the contrary in ALM, which uses reactive approach, RP might be flooded with lots of reconnecting requests, especially if the departed node had lots of children. All failed nodes need to be reconnected to the multicast group, which increases average recovery time. Large number of nodes in the overlay tree is disadvantage for ALM while it is advantage for HM; the more nodes we have on the network the easier it is to find a backup.

In the Figure 7-6 we can see the trend notably. The maximum recovery time of the ALM is much higher than HM. The time of leave is smaller than failure; this is because departing node notifies RP in advance. In the case of failures, children nodes find out about failure only after timeout. Especially when the failed node is close to the root, lots of redirections occur on the network. Our proposal scheme has much smaller delay, the difference lies in the method of the alternative parent selection in advance. HM keeps a list of candidate nodes and if there is a failure, it immediately switches to first available backup route and then the join request is sent to HMRP. The influence of failure is the time of notice when heartbeat message is not received by HMRP.

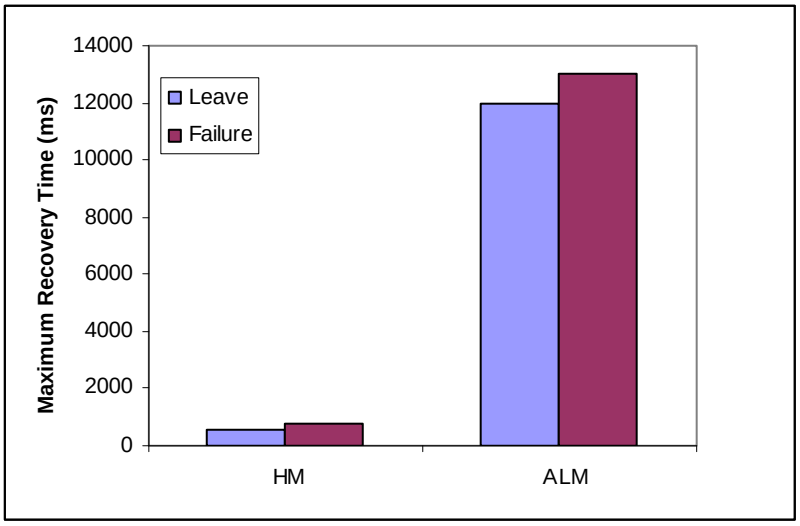


Figure 7-6. Maximum Time for Recovery

Our proposal achieves better performance and the fast recovery in both leave and failure cases consistently. On the other hand reactive approach does not show good enough performance in failure case, especially in maximum recovery time. We can conclude here that reactive approach is undesired and unsuitable for the real time applications.

We also compare HM with Island Multicast (IM). For this simulation we have a different network setup. IM requires us to have interconnected domains, each having IP Multicast capability. We create four network domains, each having 20 routers and 20 nodes (end-host); one agent is attached per router. The network domains are interconnected with four intermediate routers with unicast-only capability. We select root for each domain and build IP Multicast group. Randomly selected node from another domain joins this group, the overlay link is built between different domains, and data is distributed at the IP layer for the local group members. With the increase

of the group size we randomly select node and let it fail, then we record how much time it takes to recover the failed node(s). The results are plotted in Figure 7-7.

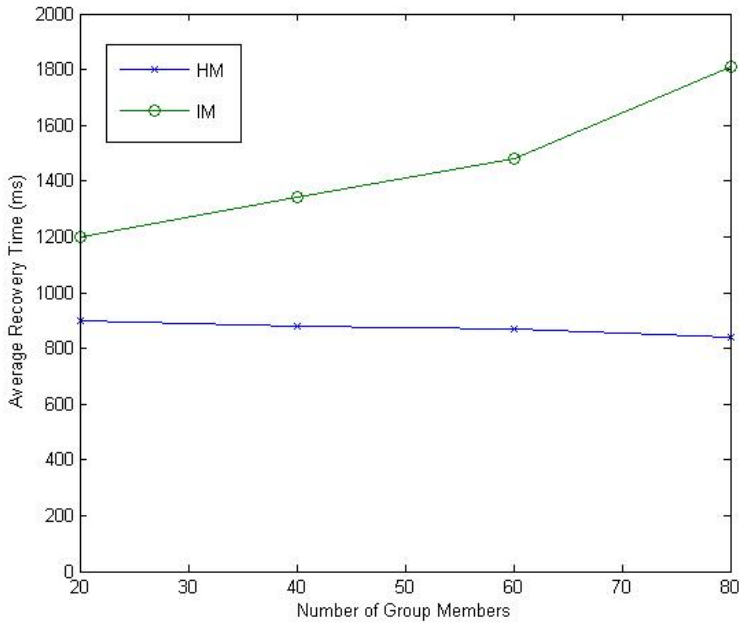


Figure 7-7. HM and IM Average Recovery Time in Failure

IM is different from the ALM and requirements for IM are different as well. Interconnected islands use the IP Multicast within the domain, so if any of the nodes fails the recovery is done by the IP Multicast protocol. As we mentioned earlier in Section 4.9 (Backup Routes at IP Multicast Segment) in HM, if IP Multicast is available, we use it, and at the same time we build overlay path to server at application layer. In the case of failure in HM, the node reconnects to backup parent immediately, while IP Multicast protocol tries to bring this node back to the network. In IM, also the failure is recovered via IP Multicast protocol, however sometimes the leader nodes fail, so the new leader election algorithm is executed. As original design

of IM suggests, the node with the smallest ID in the island will become a new leader. Initially all nodes declare to be leaders, during the leader election process, each node sends its ID to the neighboring nodes, and in the end of the process everybody finds out who has the smallest ID. The new leader sends join request to closest neighboring island, the overlay link is built and all other nodes, within the island connect to the newly elected leader at IP layer. The islands itself are connected with Minimum Spanning Tree algorithm (MST).

HM has exactly the same implementation, as explained in Section 4.9. Although IP Multicast is available, we build overlay links too, if failure is detected, node switches to backup route unless IP Multicast protocol has reconstructed the tree. The major difference between these two protocols is the way they prepare for failures. In IM failure of regular member is not a big deal, however if leader node fails we have significantly bigger delay. We should also mention that the IM protocol has better performance than ALM, however IP Multicast enabled islands are not always available, so IM is not even an option sometimes.

In Section 6.2 we present method of the proactive route maintenance for HM, which enables us fast recovery from node departures and failures. In comparison with other multicasting protocols, such as ALM and IM, our proposal performed much better and delay penalty caused by failures was reduced significantly. Main advantage of our proposal, is the backup parent finding in advance, so if failure occurs in the multicast tree, each node is protected from a long delay penalty. The overhead of

searching a backup parent depends on the group size, the more members we have, the easier it is to find the backup.

According to the simulations we can conclude, that HM is a multicasting protocol that addresses number of issues, which are not addressed in other multicasting protocols therefore we have better performance. In some multicasting applications, in order to provide multicast service, a dedicated server needs to be placed all over the Internet and run end-host multicast protocol among these servers and connect end-users with the unicast messages. The design issues include the deployment of dedicated servers and making routing decisions with the knowledge about server locations, bandwidth and delay. Multicast server overlays certainly provide better multicast service than pure end-host multicast applications however the deployment of such servers is not always feasible. HM does not require hardware deployment, it is purely application layer protocol, which efficiently uses IP layer resources and provides reliable service to end-users.

7. Concluding Remarks

We present a method for multicasting called Hybrid Multicast. The behavior of the algorithm is close to the hybrid approach of combining different types of multicast protocols and implements the negotiation part.

Hybrid Multicast is a transition strategy of multicast and unicast. It builds dynamic links from the host and can support any host application; this is done by encapsulating multicast packets as unicast, or vice versa. HM uses the UDP protocol for distributing media content. The hybrid version of multicasting, introduced in this thesis is mainly dedicated to the real time video distribution, such as video conferencing. This is an effective way to provide video content over the Internet until all ISPs support multicast and update their routers. Hybrid Multicast can support any platform; any multicast protocol can be integrated into the HM; in the end we have data being processed either at IP layer or Application layer, depending on the availability of IP Multicast.

The strength of hybrid multicast is that it does not use any designated gateway machine for transition (multicast to unicast) and uses available resources of IP Multicasting which definitely performs better than any ALM. Proactive approach of route maintenance makes HM more reliable. Most of the multicasting applications use the reactive approach. The problem is taken care after it has occurred, while HM prepares for the failure in advance and constructs the backup routes in advance. IP

Multicast protocol has no retransmission methods, we add NAK to hybrid multicast and treat IP Multicast segment as overlay to solve the problem of packets loss.

In summary, simulation results have been shown that confirm that our proposal works better than other existing multicast protocols. It generates less overhead than ALM for data delivery, and reconstructs the broken path in an efficient way.

In the future, our work should be tested in real networks in different network environments and with large number of multicast group members. Thus problems of Hybrid Multicast can be identified easier. Real network is always the best way for proposal evaluation, by doing that we will obtain more extensive results. For the future work, we will also consider security issues for the Hybrid Multicast; the packets on the multicast tree will be encrypted with the latest encryption algorithms (e.g. RSA, IDEA, DES, etc.) protect the data during transmissions.

References

1. Katherine Guo, Injong Rhee; “Message Stability Detection for Reliable Multicast”; INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies.; 2000

2. Tetsuya Kusumoto, Yohei Kunichika, Jiro Katto, Sakae Okubo; “Tree-Based Application Layer Multicast using Proactive Route Maintenance and its Implementation”; International Multimedia Conference, Proceedings of the ACM workshop on Advances in peer-to-peer multimedia streaming; 2005

3. Cheuk, K.-W.R, Chan, S.-H.G, Lee, J.Y.-B.; “Island multicast: the combination of IP multicast with application-level multicast”; Communications, IEEE International Conference on Volume 3, Issue, Page(s): 1441 - 1445 Vol.3; 2004

4. “NACK-Oriented Reliable Multicast Protocol (NORM)”, http://www.ietf.org; 2002

5. Internet Protocol (IP) Multicast, http://www.linuxjunkies.org/html/Multicast-HOWTO.html; http://www.nanog.org/mtg-9806/ppt/davemeyer/; http://www.cisco.com/en/US/products/ps6552/products_ios_technology_home.html;

6. Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, Samir Khuller;”Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications”; In Proceedings of INFOCOM, San Francisco, CA; 2002

7. S. Shi and J. Turner; “Routing in overlay multicast networks” Department of Computer Science, Washington University in St. Louis; IEEE INFOCOM 2002

8. Thilmee M. Baduge, Hirozumi Yamaguchi and Teruo Higashino;"An Efficient Overlay Multicast Protocol for Heterogeneous Users"; Transactions of Information Processing Society of Japan; Vol.46, No.11(20051115) pp. 2614-2622; 2005

9. Nobuo Funabiki, Jun Kawashima, Shoji Yoshida, Kiyohiko Okayama, Toru Nakanishi and Teruo Higashino; "P2PMM router: a Two-Stage Heuristic Algorithm to Peer-to-Peer Multicast Routing Problems in Multihome Networks", IEICE Transactions on Fundamentals, Vol.E87-A, No.5, pp.1070-1076; 2004

10. W. Zeng, Y. Zhu, H. Lu, and H. Jiang, “A novel path-diversity overlay retransmission architecture for reliable multicast,” IEEE Inter. Confer. Multimedia and Expo; 2006

11. M. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan; “The minimum latency problem,” in Proc. ACM Symposium on Theory of Computing; 1994

12. D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel; “ALMI: An Application Level Multicast Infrastructure”; Proc. USENIX USITS 2001, pp.49–60; 2001

13. Kevin Almeroth, Jeremy Mineweaser, Mark Pullen, Keith Ross, Kurt Tutschku, Wenjun Zeng; “Scalable Adaptive Multicast SAM RG in IRTF”; Panel on Scalable Adaptive Multicast SAM RG in IRTF; 2007

14. J. Liebeherr, M. Nahas, and W. Si; “Application-layer multicasting with delaunay triangulation overlays”; Selected Areas in Communications, IEEE Journal on Volume 20, Issue 8, Page(s): 1472 - 1488; 2002

15. Y.-H. Chu, S. G. Rao, and H. Zhang; “A Case for End System Multicast”; Proc. ACM Sigmetrics; 2000

16. W. Zeng, Y. Zhu, H. Lu, and H. Jiang; “A novel path-diversity overlay retransmission architecture for reliable multicast”; IEEE International Conference Multimedia and Expo; 2006

17. J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O’Toole; “Overcast: Reliable Multicasting with an Overlay Network”; In Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI); 2000

18. N.F. Maxemchuk et al.; "A Cooperative Packet Recovery Protocol for Multicast Video"; Int'l Conf. on Network Protocols; 1997

19. S. Pejhan et al; "Error Control Using Retransmission Schemes in Multicast Transport Protocols for Real-Time Media"; IEEE/ACM Transactions on Networking, vol. 4, No. 3, pp. 413-427; 1996

20. “Network-based service for recipient-initiated automatic repair of IP multicast sessions”; Technical Report, Patent #:6567929 , AT&T Labs Research; 2003

21. Y.Chu, S. G. Rao, S. Ses, H.Zhang; “Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture”; In Proc. ACM SIGCOMM 2001, San Diago, CA; 2001

22. M. Yang, Z. Fei; “A Proactive Approach to Reconstructing Overlay Multicast Trees” in proceedings of INFOCOM; 2004.

23. S. Deering; “Host Extension for IP Multicasting”; IETF, RFC 1112; 1989

24. Dennis M. Moen; “Overview of Overlay Multicast Protocols”; http://bacon.gmu.edu/XOM/pdfs/MulticastOverview.pdf C3I Center Technical Report; George Mason University.

25. Konrad Lorincz; “HyperCast: A Super-Scalable Many-to-Many Multicast Protocol for Distributed Internet Applications”; School of Engineering and Applied Science, University of Virginia; 2001 http://www.comm.utoronto.ca/hypercast;

26. A. Rowstron and P. Druschel; “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”; IFIP/ACM Middleware 2001 (Heidelberg, Germany); 2001

27. PGMCC single rate multicast congestion control; Internet Engineering Task Force INTERNET-DRAFT http://tools.ietf.org/html/draft-ietf-rmt-bb-pgmcc 2005

28. Lei Zhang, Jogesh K. Mupala, Samuel T. Chanson; “MAPS - A Generalized Scheme for Quality of Service Routing under Delay-Bandwidth Constraints”; Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on 20-22 Oct. 2003 Page(s):491 - 496

29. T. S. E. Ng and H. Zhang; “Predicting internet network distance with coordinates-based approaches”; In Proceedings of IEEE INFOCOM; 2002

30. Y. Zhu, W. Zeng and H. Lu, “Exploiting Overlay Path-diversity for Scalable Reliable Multicast,” IEEE International Conference on Multimedia and Expo, 2007

31. P. Francis; “Yoid: Extending the Internet Multicast Architectuire”, http://www.icir.org/yoid/; http://www.isi.edu/div7/yoid/docs/ycHtmlL/htmlRoot.html

32. The Network Emulator Nist Net, http://snad.ncsl.nist.gov/itg/nistnet/;

33. Georgia Tech Internet Topology Generator, http://www.isi.edu/nsnam/ns/ns-topogen.html;

34. The Network Simulator ns-2, http://www.isi.edu/nsnam/ns;

35. Waitzman, D., C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol", IETF RFC 1075; 1988.

36. Estrin, d., D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", IETF RFC 2362; 1998

37. Moen, Dennis, and J.M. Pullen, “Enabling Real-Time Distributed Virtual Simulation over the Internet Using Host-based Overlay Multicast”, Proceedings of the Seventh IEEE Workshop on Distributed Simulation and Real-Time Applications, pp. 30-36; 2003

38. Deering, Stephen E., and David R. Cheriton, “Multicast Routing in Datagram Networks and Extended LANS,” ACM Transactions On Computer Systems, Vol. 18, No 2, pp. 85-110; 1990

39. Eriksson, H., "MBONE: The Multicast Backbone", Communications of the ACM, Vol. 37, No. 8, pp. 54-60; 1994

40. Chu, Yang-Hua, San jay G. Rao, Srinivasan Seshanand, and Hui Zhang, “Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture,” ACM SIGCOMM Computer Communication Review, pp. 55-67; 2001

41. Junginger, Markus Oliver and Yugyung Lee, “A Self-organizing Publish/Subscribe Middleware for Dynamic Peer-to-Peer Networks,” IEEE Network Magazine, Vol. 18, No. 1, pp. 38-43; January/February 2004

42. Wang, Wenjie, David Helder, Sugih Jamin, and Lixia Zhang. “Overlay Optimizations for End-host Multicast”, NGC02, ACM, 2002, pp. 154-161.

43. Berstekas, Dimitr P., "Network Optimization: Continuous and Discrete Models", Athena Scientific, Belmont, Massachusetts, 1998

44. Yan, Shuqian, Michalis Faloutsos, and Anindo Banerjea, “QoS-Aware Multicast Routing For The Internet The Designing And Evaluation Of QoSMIC,” IEEE/ACM Transactions on Networking, Vol. 10, Issue 1, pp. 54-66; 2002

45. Liebeherr, J., M. Nahas, and Si Weisheng, "Application-Layer Multicasting with Delaunay Triangulation Overlays", IEEE Journal on Selected Areas in Communications, Vol. 20, Issue 8, pp. 1472-1488; 2002

46. Cui, Yi, Baochun Li and Klara Nahrestedt, “oStream: Asynchronous Streaming Multicast in Application–Layer overlay Networks ,” IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, pp.91-106; 2004,

47. Francis, Paul, "Yoid Tree Management Protocol (YTMP) Specification", ACIR Center for Internet Research, Berkeley, CA, April 2000.

48. Kwon, Minseok, and Sonia Fahmy “Topology-Aware Overlay Networks for Group Communication”, ACM NOSSDAV, pp. 127-136; 2002

49. Kwon, Minseok, and Sonia Fahmy “Topology-Aware Overlay Networks for Group Communication”, ACM NOSSDAV, pp. 127-136; 2002

50. Jannotti, John, David K Giffors, Kirk L. Johnson, M. Frans Kasschoek, James W. O’Toole, Jr. “Overcast: Reliable Multicasting with an Overlay Network”, Cisco Systems; OSDI 2000, San Diego, CA, 2000

51. Chu, Uang-hus, Sanjay G. Rao, and Hui Zhang, “A Case for End System Multicast,” ACM SIGMETRICS, pp. 1-12; 2000

52. Wang, Wenjie, David Helder, Sugih Jamin, and Lixia Zhang. “Overlay Optimizations for End-host Multicast”, NGC02, ACM, pp. 154-161; 2002

53. Zhang, Rongmei, and Y. Charlie Hu, “Borg: A Hybrid Protocol for Scalable Application-Level Multicast in Peer-to-Peer Networks,” ACM NOSSDAV, pp. 172-179; 2003,

54. Castro, Miguel Peter Druschel, Anne-Marie Kermarrec, and Antony I. T. Rowstron “Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure,” IEEE Journal On Selected Areas In Communications, Vol. 20, No. 8, pp. 1489-1499; 2002

55. Rowstron, Antony, and Peter Druschel. "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems", Proc. IFIP/ACM Middleware 2001, Heidelberg, Germany; 2001

56. Zhuang, S. Q., B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. Kubiatowicz, “Bayeux: An Architecture for Scalable and Fault-tolerant Wide-Area Data Dissemination”, Proceedings of the Eleventh International Workshop on Network and Operating System Support for Digital Audio and Video; 2001

57. Single Source Shortest Path (SSSP) http://en.wikipedia.org/wiki/Dijkstra's_algorithm

58. Bellman-Ford algorithm http://en.wikipedia.org/wiki/Bellman-Ford_algorithm

59. Yatin Chawathe, “Scattercast: An Adaptable Broadcast Distribution Framework”, Multimedia Systems, ACM, pp. 104 – 118; 2003

60. D. Pendarakis, S. Shi, D. Verma, M. Waldvogel “ALMI: An Application Level Multicast Infrastructure”; 3rd USENIX Symposium on Internet Technologies, San Francisco, March 2004