# Session Management in Multicast

Tianyu Wang

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy (Ph.D.) at
Concordia University
Montreal, Quebec, Canada

September 2008

# CONCORDIA UNIVERSITY

## School of Graduate Studies

This is to certify that the thesis prepared

By: Tianyu Wang

Entitled: Session Management in Multicast

and submitted in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy (Ph. D.) in Computer Science**

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. R. Zmeureanu

_____ External Examiner
Dr. Kevin C. Almeroth

_____ External to Program
Dr. F. Khendek

_____ Examiner
Dr. G. Butler

_____ Examiner
Dr. J. Opatrny

_____ Supervisor
Dr. J.W. Atwood

Approved by _____
Dr. N. Shiri, Graduate Program Director

_____ 20____ _____
Dr. Robin A. L. Drew, Dean
Faculty of Engineering and Computer Science

# Abstract

Session Management in Multicast

Tianyu Wang (Ph. D.)

Concordia University, 2008

As a new network technique to efficiently distribute information from a small number of senders to large numbers of receivers, multicast encounters many problems in scalability, membership management, security, etc. These problems hinder the deployment of multicast technology in commercial applications. To overcome these problems, a more general solution for multicast technology is needed. In this paper, after studying current multicast technologies, we summarized the technical requirements for multicast, including data delivery, scalability, security, group management, reliability, and deployment. In order to understand and meet the requirements, we define a life cycle model that most multicast sessions should follow. According to the requirements and the life cycle model, we propose and design a general solution that can control each phase of a session and satisfy most requirements for multicast technology. This general solution has three parts: hierarchical topology auto-configuration algorithm, Session Management Mechanism, and techniques supporting different multicast protocols. To verify the feasibility of our solution and compare its performance with other multicast techniques, we simulate our solution and compare it with PIM-SM and ESM.

# Acknowledgements

I would like to express my gratitude to my supervisor, Dr. J. W. Atwood, Concordia University, Montreal, for his insight and suggestions guiding me to complete my thesis.

My very special thanks to my wonderful wife, Qian Wang, for her unremitting love and support throughout my project. I am also grateful to my mother, Huang Qi, mother-in-law, Lin Juhui for their understanding. I would like to take this opportunity to thank my brother, Wang Tianwu, and my brother-in-law, Yale Wang, for their encouragement.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

Multicast is a technique that can efficiently distribute information from a single source to thousands of receivers on the Internet [1]. Multicast communication is currently a topic of intense study in telecommunication companies and the research community and is growing into a true challenge for Internet engineers [1]. It is now being offered by some networking equipment manufacturers, e.g., Cisco [2], and is planned for use by a number of companies offering large-scale Internet applications and services. Many new Internet services will be based on multicasting, e.g., Internet TV, large-scale Internet Conference, etc [3]. Some commercial news vendors have already used multicast to propagate their news and trading data, for which we are bound by non-disclosure agreements with the news vendors and cannot give details of their techniques.

On modern Internet, multicast technology has a promising future in its commercial usage, which refers to Internet Service Providers (ISP) and content providers' activities of distributing information to large groups of users by multicast and generating profits by charging users for their consumption of such information.

Although multicast has received lots of attention, it still has many issues to be resolved, e.g., scalability, security, etc. For example, current multicast technology has problems in dealing with senders and receivers located in different administrative Internet domains (Autonomous Systems, AS). Another problem encountered by current multicast technology is its “anyone can send, anyone can receive” service model because of lack of access control, which makes difficulties for ISPs to charge users for multicast services. Therefore, most ISPs are reluctant to accept multicast as a solution for commercial applications. The sources of information about the ISPs’ intentions and concerns about multicast are discussion on many online forums [4] and some informal chats with ISP staff at various firms, for which we cannot give specifics due to non-disclosure agreements with these firms.

Because most current multicast technologies are not commercially feasible, we need to build a general solution that will make the commercial deployment of multicast distribution technology more attractive to ISPs.

To create such a solution, we need to analyze the requirements for multicast technology and the relationship among these requirements. While our

motivation comes from the lack of commercial adoption, and we have taken care to ensure that the commercial requirements are well enunciated, the analysis must cover all factors that constrain possible solutions. The origin of the current scenario is the confusion of the requirements for multicast technology. Generally, ISPs prefer multicast techniques that are easy to use, profitable, and manageable. However, the detailed definition of their requirements has never to our knowledge been illustrated. Therefore, no comprehensive solution has been proposed.

The first piece of work in my research is to provide a detailed discussion and definition of requirements for multicast technology, including data delivery, scalability, security, group management, reliability, and deployment.

To study and meet the requirements, we define a life cycle model of multicast sessions, which is the second innovative work in this project. The life cycle model is a generic procedure that most reliable multicast sessions should follow. In this life cycle model, the requirements will be presented.

The life cycle model will also lead us to a Session Management Mechanism. It is the general solution we propose for multicast technology and is the core

of the contributions in our work. It will provide a good framework to satisfy the requirements we summarized and cover all phases in the life cycle model.

To verify that the session management mechanism can meet the requirements of multicast technology, and to compare its performance with other multicast technologies, we designed and conducted a simulation experiment. The simulation is the fourth part of this project, and provides important verification and validation for our design. The simulation is done using a commercial network protocol simulation platform, Opnet Modeler.

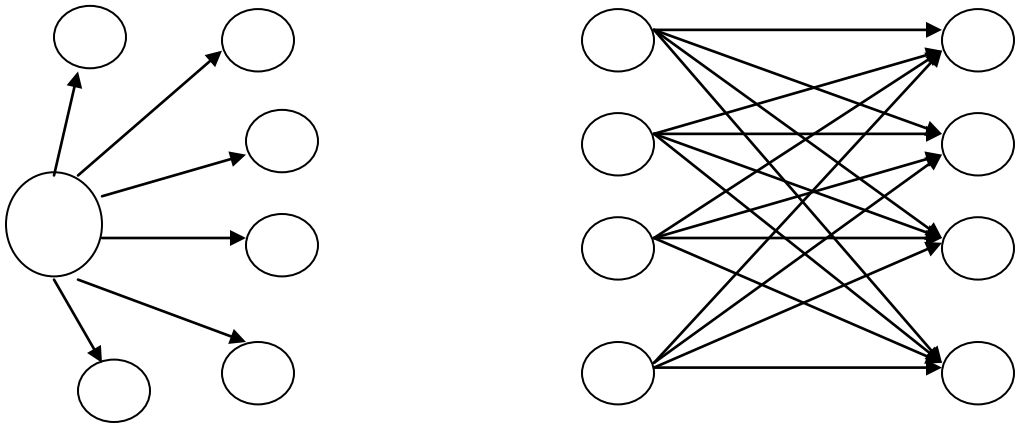
In this paper, Chapter 2 introduces and summarizes the current multicast technologies. Chapter 3 defines the requirements for multicast technology based on our research on current multicast technologies, narrows our objectives to a proper range of focus, and presents the life cycle model for a multicast session. In Chapter 4, we will provide an overview of the project, including objectives, division, and introduction. Chapter 5 describes the design of the Session Management Mechanism, including its hierarchical topology auto-configuration, detailed design of its modules, and its support for different multicast protocols. Chapter 6 presents a simulation that is used to test the feasibility of our solution and to compare its performance with

other multicast technologies. Chapter 7 summarizes this project and outlines the future research work.

# 2 Overview of Multicast Technology

There is a growing requirement for techniques that can exchange messages among the members of a large group on the Internet. When data are to be sent from one or more senders to multiple receivers, unicast has been proven to be inefficient. Multiple unicast may be feasible for small groups, but not for large numbers of receivers. Multicast has been recognized as the most efficient way to distribute information from a single source to multiple destinations. As for next-generation Internet, multicasting is one of a few techniques without which a certain class of application is infeasible, e.g., Internet TV, large-scale distributed database application, and video conferencing.

## 2.1 Group Communication and Multicast Technology



Point-To-Multipoint (1-to-N) Multipoint-to-Multipoint (N-to-M)

**Figure 1 Group Communication**

Generally, group communication can be divided into two classes, Point-to-Multipoint (1-to-N or one-to-many) mode, where data are sent from a single source to multiple destinations, and Multipoint-to-Multipoint (N-to-M or many-to-many) mode, which involves multiple senders and multiple receivers. The topologies of these group communication modes are shown in Figure 1. Our discussion is focused on the 1-to-N mode, which is the group communication scenario happening most often on the Internet. The N-to-M mode has its own features and problems, which will not be covered in our discussion.

Basically, the requirements of group communication should include efficient data delivery, effective group management, and acceptable quality of service (QoS). For some complex applications, some other requirements will be desired, e.g., reasonable security and reliability.

Generally, in 1-to-N group communication, if the number of receivers is not very large, the current unicast techniques can satisfy the requirements very well. The data delivery, QoS, and reliability can be guaranteed by current TCP/IP protocols and increasing network bandwidth. Group management and security are not very complicated in small groups and can be solved by

some extra effort in the application layer, as is used in many existing web applications.

However, when the number of receivers is extremely large, most requirements for 1-to-N group communication cannot be satisfied efficiently by the current TCP/IP client-server model. First, the server has to serve every client, and duplicated copies of the same messages are sent from the source to every single destination. Because the server has limited resources, it cannot efficiently serve all data requests from an enormous number of clients, especially when the number of receivers is over thousands or even millions. When such a situation occurs, the server's capacity will become the bottleneck of the group communication, and the requirements of data delivery, QoS, and reliability will become impossible. In this case, group management and scalability are also new challenges that must be dealt with by new techniques.

A better and more reliable alternative solution is multicast. In multicast, the sender just sends a single copy to all receivers, and this communication group can be identified by a single group address. The multicast groups are automatically organized into a distribution topology, and the data stream will

be automatically delivered along the topology to the receivers. These automatic processes are completely transparent to the senders and end receivers. Only the Internet service providers (ISPs) will be interested in the details of multicast technology. Multicasting brings up a potential to reduce resource requirements in large groups.

Multicast is an efficient way to distribute information from a single source to multiple destinations. Efficiency in multicast comes from two ways:

- Number of transmissions from a source
- Number of packets generated within the network

A source needs only to transmit once instead of n times for n destinations when multicast is used instead of multiple unicasts. Similarly, by virtue of using a source-based tree at the network level for distribution, multicast is able to reduce the number of packets within the network significantly compared to multiple unicasts [1].

Currently, the research community is developing multicast technology in three layers: network layer, transport layer, and application layer. At the beginning of multicast development, the research community focused on the

network layer. Multicast technology in the network layer is focused on routing and data distribution issues. In the early years, research groups designed some protocols that can only work well in a closed system or a single autonomous system (AS), which are called intra-domain multicast protocols. After realizing the limitation of intra-domain protocols, research groups started to develop protocols that can work across the boundaries between ASes, which are called inter-domain multicast protocols. However, the reliability of data delivery cannot be guaranteed by the network layer multicast protocols in many applications where the reliability is required.

To meet the requirements of data reliability, the research community started to develop multicast techniques in the transport layer. The main goal of transport layer multicast is to provide reliability for multicast traffic. This is a completely different problem compared to the reliability in unicast because the number of receivers is enormous and data retransmission requests can overwhelm a single source very easily. Because multicast techniques in network and transport layers are based on IP, they are called IP multicast.

Due to the immaturity of IP multicast, some research groups working on application layer techniques assume that there will not be massive support

for IP multicast from ISPs in the near future. They proposed that the fastest way to deploy multicast on the Internet is to develop multicast in the application layer and let the end systems communicate directly, without any router support. Therefore, they developed another kind of multicast technology called overlay multicast.

In the following sections in the chapter, we will introduce some existing techniques in the network, transport, and application layers, including some supporting techniques. Our discussion will be based on the understanding of these existing techniques.

## 2.2 IP Multicast

IP multicast has been developed for over ten years in the research community and in telecommunication companies. Many protocols have been proposed and developed. In this section, we will give a brief overview for existing IP multicast technology and some supporting techniques.

### 2.2.1 Introduction to IP Multicast

The IP multicast technology is focused on routing, data distribution, and reliability issues in the network and the transport layers. The IP multicast protocols rely on IP technology and router assistance to build the data

distribution infrastructure, route the data packets to destinations, gather error reports from receivers, and fulfill other operations.

Generally, there are two ways to manage a multicast group. One way is to build a source-based tree for each group. The sender is the root of the data distribution tree. For multiple groups, multiple independent trees have to be established. Another way is to send data to a central distributor and let the distributor dispatch the data along a hierarchical tree. This method is called the shared-tree method.

From the first Internet multicast experiment in 1992, the Internet multicast protocols development was focused on a single flat topology. There were several multicast routing protocols developed for this flat topology in Internet multicast standardization and deployment. The existing multicast protocols before 1997 are now called intra-domain multicast protocols. The most serious drawback of the intra-domain protocols is that they cannot handle receivers and senders in different autonomous systems (AS) [5].

From the middle of 1997, the research community realized the need for a hierarchical multicast infrastructure and inter-domain routing [5]. Inter-

domain multicast has evolved out of the need to provide scalable, hierarchical, Internet-wide multicast [5]. However, the inter-domain technology is relatively immature. Protocols that provide the necessary functionality are being considered by the IETF and are being evaluated through extensive deployment. Because the protocols lack elegance and long-term scalability, they are considered as a short-term solution and possibly only an interim solution.

Both of the intra-domain and the inter-domain multicast protocols are routing protocols developed in the network layer. Another class of IP multicast protocols is multicast protocols in the transport layer. The main goal of these protocols is to provide reliability to multicasting. The current multicast technology in the transport layer focuses on error recovery, flow control, and some other issues.

## 2.2.2 Intra-domain Protocols

### 2.2.2.1 The early Multicast Backbone (MBone) and the DVMRP protocol

First, we will introduce a specific class of multicast protocols: dense mode multicast protocols. Dense mode refers to an environment where the

multicast members are relatively densely packed and bandwidth is plentiful [6]. The DVMRP, MOSPF, and PIM-DM protocols belong to this class.

The early efforts for building a multicast-capable Internet and creation of the Multicast Backbone, MBone, were motivated by Stephen Deering's IP multicast model. In March 1992, the MBone carried its first worldwide event when 20 sites received audio from the meeting of the IETF in San Diego. While the conferencing software itself represented a considerable accomplishment, the most significant achievement here was the deployment of a virtual multicast network [5].

The original multicast routing protocol was the Distance Vector Multicast Routing Protocol (DVMRP). DVMRP constructs source-based multicast trees using Reverse-Path Multicast (RPM) protocol. The multicast tree built by DVMRP is also called a reverse shortest path tree.

A daemon process called *mrouted* was running on routers and workstations, and this process provided the multicast routing function. While receiving unicast-encapsulated multicast packets from an incoming interface, the *mrouted* process will forward the packet through a proper set of outgoing

interfaces. Connectivity among these machines is provided by a point-to-point IP-encapsulated *tunnel* [5]. Each tunnel is a logical link between end-points, but it can cross several routers.

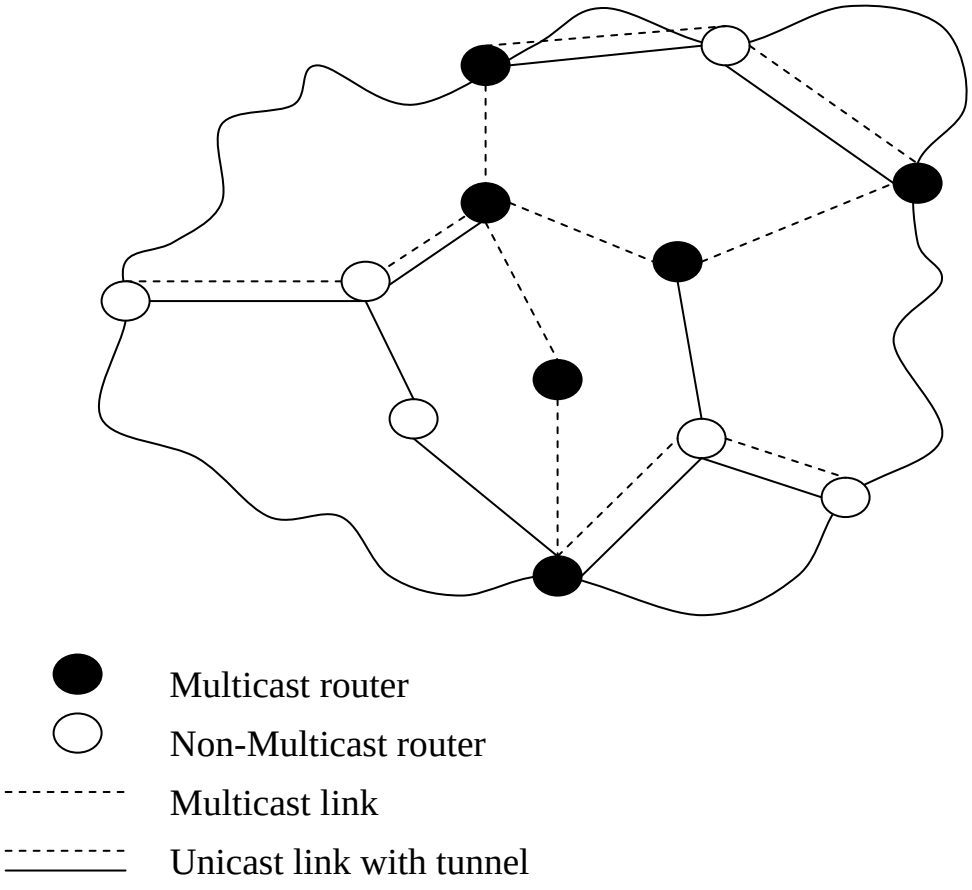



**Figure 2 A Tunnel-based Topology of Early MBone**

This method is a primitive multicast routing algorithm. It is actually a controlled form of flooding algorithm. It makes routers send a lot of prune messages and forward lots of packets to achieve a better and dynamic routing solution. Because there are thousands of routers that may be interested in this multicast session and there may also be thousands of

multicast sessions on the Internet, the large number of prunes and forwarded packets makes this algorithm inefficient and infeasible in a wide-area network.

#### 2.2.2.2 Multicast Extensions to Open Shortest Path First (MOSPF)

This method is based on extending the Open Shortest Path First (OSPF) protocol to provide multicast routing capacity. In OSPF, each router keeps topological and state information of the routing domain, by link-state advertisement (LSA) flooding. Similarly, MOSPF routers use IGMP to monitor multicast membership on directly attached subnets and flood an OSPF area with information about group receivers. This allows all MOSPF routers in this area to have the same view of group membership [5]. Each MOSPF router can independently construct the shortest-path tree for each source and group by Dijkstra's algorithm, in the same way as in OSPF. After the multicast tree is built, group membership is used to prune the branches that do not lead to subnets with group members. The result is a pruned shortest-path tree rooted at the source [7]. MOSPF is considered as a dense mode multicast protocol because the membership information is broadcast throughout the area and to all the MOSPF routers.

#### 2.2.2.3 Protocol Independent Multicast – Dense Mode (PIM-DM)

PIM-DM is similar to DVMRP. The PIM-DM uses the RPF algorithm and uses *Graft* messages to add branches that have been previously pruned. There are only two differences between PIM-DM and DVMRP. The first one is that PIM takes advantage of the IP routing information to perform the RPF checks, while DVMRP maintains its own routing table. The second is that DVMRP tries to avoid sending unnecessary packets to its neighbors who will generate prune message based on a failed RPF check. So the DVMPF router builds its routing table in a way that the routing table only includes the downstream routers that use the given router to reach the source. PIM-DM simply floods packets on all outgoing interfaces.

#### 2.2.2.4 The Core Based Tree (CBT)

The multicast protocols described above are all dense mode multicast protocols, which broadcast membership information throughout the network. Now let us discuss another class of multicast protocol, sparse mode multicast protocols. Sparse mode refers to an environment where group members are distributed across many regions of the network, and bandwidth is not necessarily widely available. In sparse mode multicast, receivers explicitly send join requests to the core router, without widely broadcasting traffic and triggering the prune message.

CBT uses the basic sparse mode paradigm to create a single *shared tree* used by all sources [5]. The root of this shared tree is called a core. All senders send their data to the core, and the core forwards these data packet to all receivers. Receivers send explicit join messages to the core. The shared tree is a bi-directional tree, which is more complicated but more efficient when a packet traveling from a source to the core comes across branches of the multicast tree.

A host first sends a *join-request* message to the local router. This step is to explicitly express its interest in the multicast session. Then the local router will contact the next-hop router on the shortest path toward the core router. The *join-request* message sets up transient join states on the routers on the path it traverses. The *join-request* travels hop by hop toward the core, until a core or an on-tree router receives this message and accepts this join request. Then the router that accepts this new child sends a *join-acknowledgement* back along the reverse path to the router that initiated the join request. When a router on the path, which received the *join-request* previously and is in join state, receives this join-acknowledgement, it updates its forwarding table, becomes an on-tree router, and forwards the *join-acknowledgement* toward the requesting router.

There is a dynamic and automatic tree maintenance mechanism in CBT. The routers can periodically send a CBT “keep-alive” (i.e., *echo-request*) to its parent router on the tree. The parent router sends a response (i.e., *echo-reply*) back to its child when it receives a “keep-alive” message from a valid child. If there is no response in a predefined time threshold, the child should send a “*quit-notification*” message toward the core and send a “*flush-tree*” message to all downstream branches. In this way, all its child routers can know the changes of the multicast tree, leave the tree, and re-join individually, if it is necessary.

#### 2.2.2.5 Protocol Independent Multicast – Sparse Mode (PIM-SM)

Protocol Independent Multicast - Sparse mode (PIM-SM) is a multicast routing protocol that can use the underlying unicast routing information base or a separate multicast-capable routing information base. It builds unidirectional shared trees rooted at a Rendezvous Point (RP) per group, and optionally creates shortest-path source-based trees for each source [7].

A Rendezvous Point (RP) is a router that has been configured to be used as the root of the non-source-specific distribution tree for a multicast group. Join messages from receivers for a group are sent towards the RP, and data

from senders are sent to the RP so that receivers can discover who the senders are, and start to receive traffic destined for the group [7].

Generally, PIM-SM has three phases. In phase one, a multicast receiver expresses its interest in receiving traffic destined for a multicast group. One of the receiver's local routers is elected as the Designated Router (DR) for that subnet. On receiving the receiver's expression of interest, the DR then sends a PIM Join message towards the RP for this multicast group. When many receivers join the group, their Join messages converge on the RP, and form a distribution tree, known as the RP Tree (RPT), for group G that is rooted at the RP. A multicast data sender just starts sending data destined for a multicast group. The sender's local router (DR) takes those data packets, unicast-encapsulates them, and sends them directly to the RP using a ‘register’ packet. The RP receives these register-encapsulated data packets, extracts the data, and forwards them onto the shared tree [7].

To obtain lower latencies, the PIM-SM protocol may optionally initiate a transfer from the shared tree to a source-specific shortest-path tree (SPT). Therefore, in phase two, the RP can choose to switch to native forwarding. To do this, when the RP receives a register-encapsulated data packet from

source S on group G, it will normally initiate a source-specific Join towards S. This Join message travels hop-by-hop towards S, instantiating source-specific multicast tree state in the routers along the path. Eventually the Join message reaches S's subnet or a router that already has source-specific multicast tree state, and then packets from S start to flow following the source-specific tree state towards the RP [7].

However, having the RP join back towards the source does not completely optimize the forwarding paths. For many receivers the route via the RP may involve a significant detour when compared with the shortest path from the source to the receiver. Therefore, in phase three, a router on the receiver's LAN, typically the DR, may optionally initiate a transfer from the shared tree to a source-specific shortest-path tree (SPT) [7]. For most commercial routers, this optional transfer is done as soon as packets begin to flow to the groups.

## 2.2.3 Inter-domain Protocols

### 2.2.3.1 Internet Standard Multicast (PIM-SM/MBGP/MSDP)

Currently, in the network layer, the best and most complete inter-domain routing plan is a set of protocols, MBGP, PIM-SM and MSDP, and also known as the Internet Standard Multicast (ISM) service model.

PIM-SM is an intra-domain multicast protocol and has some scalability problems. It is difficult to inform an RP in one domain that there is a source in another domain. The underlying assumption is that a multicast group that spans two or more domains can have multiple RPs where each domain has only one RP. There is no mechanism to connect the various intra-domain multicast trees together. When sources are located in different domains, receivers cannot discover the existence of sources in another domain using different RPs. There is no mechanism for RPs to communicate with each other when one receives a source register message. To solve the scalability problems, two other protocols were developed.

Multiprotocol Border Gateway Protocol (MBGP) is an extension of BGP and contains the administrative machinery that providers and customers require in their inter-domain routing environment, including all the inter-AS tools to filter and control routing (for example, route maps). Therefore, to enable BGP-4 to support routing for multiple Network Layer protocols the only two things that have to be added to BGP-4 are (a) the ability to associate a particular Network Layer protocol with the next hop information,

and (b) the ability to associate a particular Network layer protocol with Network Layer Reachability Information [8].

The Multicast Source Discovery Protocol (MSDP) describes a mechanism to connect multiple PIM Sparse-Mode (PIM-SM) domains together. Each PIM-SM domain uses its own independent RP(s) and does not have to depend on RPs in other domains [9].

#### 2.2.3.2 Source-Specific Multicast (SSM)

Another solution for inter-domain routing multicast is SSM. SSM is an extension of the PIM protocol that allows for an efficient data delivery mechanism in one-to-many communications.

The network layer service provided by SSM is a "channel", identified by an SSM destination IP address (G) and a source IP address S. A receiver can receive these datagrams by subscribing to the channel (Source, Group) or (S, G). Channel subscription is supported by version 3 of the IGMP protocol for IPv4 and version 2 of the MLD protocol for IPv6 [36]. The inter-domain tree for forwarding IP multicast datagrams is rooted at the source S, and is constructed using the PIM Sparse Mode protocol. SSM removes the

requirement of MSDP to discover the active sources in other PIM domains. An out-of-band service at the application level, such as a web server, can perform source discovery [10].

In SSM, routing of multicast traffic is entirely accomplished with source trees. There are no shared trees and therefore an RP is not required. It still uses PIM-SM to construct the multicast tree, so it has almost all the drawbacks of PIM-SM.

### 2.2.4 Reliable Multicast

In the transport layer, some protocols have been developed to provide reliability for multicast transport. In this section, we will introduce two existing multicast protocols in the transport layer, and current IETF work in transport layer multicast.

#### 2.2.4.1 Local Group based Multicast Protocol (LGMP)

LGMP is based on the principle of sub-grouping for local error recovery and local acknowledgement processing. Receivers dynamically organize themselves into subgroups, which are called Local Groups. They dynamically select a Group Controller to coordinate local transmissions and to handle status reposts. The selection of appropriate receivers as Group

Controllers is based on the current state of the network and of the receivers themselves. However, the selection of Group Controller is not a task of a data transfer protocol such as LGMP. To fulfill this task, the author of LGMP has defined and implemented a separate configuration protocol, which is called the Dynamic Configuration Protocol (DCP). Packet errors are first recovered inside Local Groups using a receiver-initiated approach. Missing data units are requested from the sender or a higher level Group Controller only if not even a single number of the Local Group holds a copy of the missing data unit. Otherwise, errors will be recovered by local retransmissions. Full reliability and efficient buffer utilization are ensured by a novel, three-state acknowledgement scheme [13].

DCP provides mechanisms for an automated establishment of virtual group structures and for dynamic reconfiguration in accordance with the current network load and group membership. No manual administration is necessary. The definition of subgroups is based on a combination of multiple metrics depending on the QoS requirements of the user.

#### 2.2.4.2 Reliable Multicast Transport Protocol II (RMTP-II)

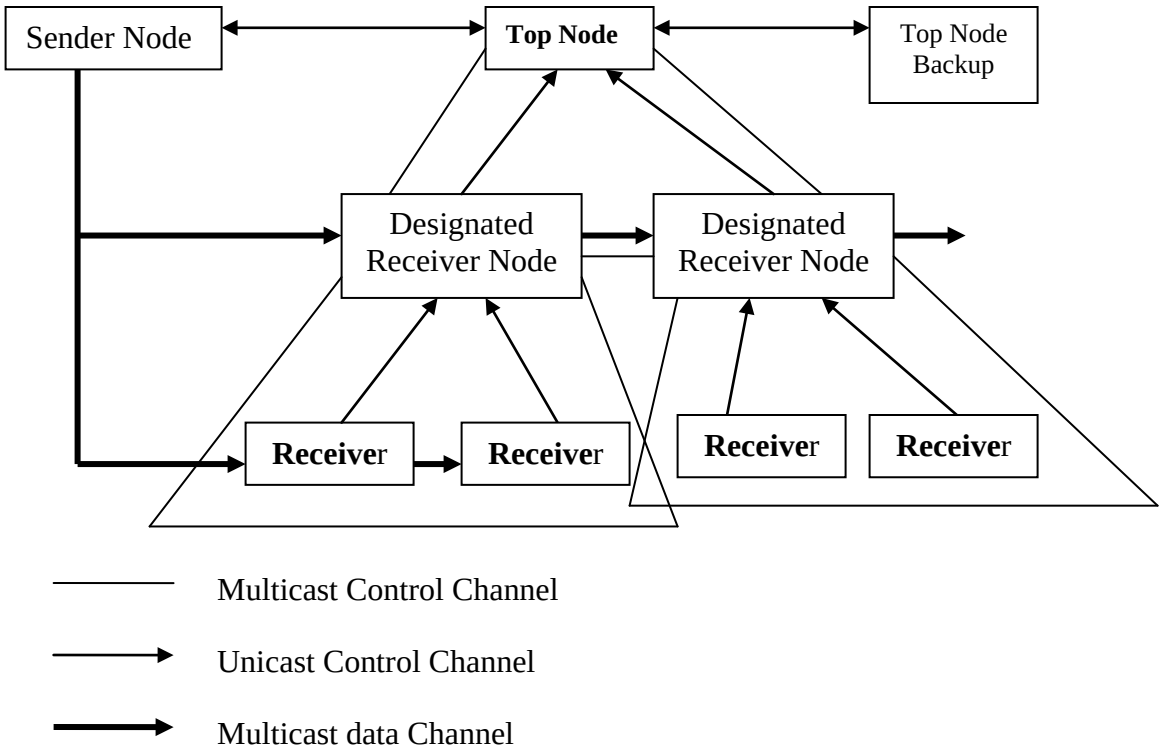



**Figure 3 RMTP - II Topology**

RMTP-II is a hierarchical protocol that provides reliable data transmission from a few senders to a large group of receivers. An RMTP-II tree consists of a single top node (TN), one or more sender nodes (SDs), many receivers (LNs), and zero or more designated receivers (DRs). There may be a backup top node, as shown in the above diagram.

The top node is assigned administratively and is the core of the tree. A receiver joins the data stream by sending an explicit JoinStream message to its parent. It can send ACK for stable data and send NACK to expedite the

recovery for missing data packets. The designated receiver can aggregate ACKs received from its children, send an aggregated ACK to its parent, and forward a received NACK to its parent. The core of RMTP-II is a set of algorithms that provide and manage Tree-Based ACKs (TRACKs), which is a key requirement of many applications that need group management and positive confirmation of data delivery to receivers.

The hierarchical structure of RMTP-II has some disadvantages. First, the TN could be a potential bottleneck in the multicast transmission because of the risk of generating more control traffic than a NACK-only protocol; and it would seriously damage the multicast group if the TN were to fail. RMTP-II provides a set of smoothing and control algorithms to manage and limit the TRACK control traffic. These algorithms do not eliminate the control traffic trade-off, but allow it to be explicitly monitored and controlled [14]. RMTP-II also minimizes the risk that the TN becomes the bottleneck of the system by minimizing the amount of work done by the TN, including restricting TN from transmission of the data packet. RMTP-II also provides an optional hot backup of the top node to eliminate the potential single point of failure of the top node. However, the risk is still very high. Even more important, there is considerable difficulty in configuring the topology of the hierarchy in a way

that is approximately congruent with the underlying physical network topology [14]. RMTP-II provides an algorithm for automatically configuring the tree if there is only a single level hierarchy, which can be sufficient for real-time applications of up to 100 or more receivers and non-real-time applications of up to 1000 or more receivers. For large deployment, RMTP-II assumes the existence of manual configuration files or a separate session manager component, to handle the configuration of interior tree nodes (DRs) [14]. So the designers of RMTP-II have left this issue out of their original design and focus their work on the core features needed for reliable delivery.

#### 2.2.4.3 IETF current work

The IETF working group on Reliable Multicast Transport (RMT) is developing building blocks, small pieces of reusable work focusing on some specific aspects of multicasting, e.g., congestion control, session tree construction, and membership management. The purpose of the building block approach is to reuse the building blocks in different reliable multicast protocols. However, until now, the IETF working group for reliable multicast transport (RMT) did not give a generic solution to multicasting in the transport layer. Currently, the IETF RMT working group focuses on design of two protocol instantiations: a NACK-based protocol and an Asynchronous Layered Coding protocol that uses Forward Error Correction.

These two protocols are designed to provide reliability to multicast in the Transport layer.

The Negative-acknowledgement (NACK) Oriented Reliable Multicast (NORM) protocol is designed to provide end-to-end reliable transport of bulk data objects or streams over generic IP multicast routing and forwarding services. NORM uses a selective, negative acknowledgement mechanism for transport reliability and offers additional protocol mechanisms to allow for operation with minimal "a priori" coordination among senders and receivers. A congestion control scheme is specified to allow the NORM protocol to fairly share available network bandwidth with other transport protocols such as the Transmission Control Protocol (TCP) [11].

Forward Error Correction (FEC) codes provide a reliability method that can be used to augment or replace other reliability methods, especially for one-to-many reliability protocols such as reliable IP multicast. The input to an FEC encoder is some number k of equal length source symbols. The FEC encoder generates some number of encoding symbols that are of the same length as the source symbols. These encoding symbols are placed into

packets for transmission. The number of encoding symbols placed into each packet can vary on a per packet basis, or a fixed number of symbols (often one) can be placed into each packet. Also, in each packet is placed enough information to identify the particular encoding symbols carried in that packet. Upon receipt of packets containing encoding symbols, the receiver feeds these encoding symbols into the corresponding FEC decoder to recreate an exact copy of the k source symbols. Ideally, the FEC decoder can recreate an exact copy from any k of the encoding symbols [12].

### 2.2.5 Supporting Technology

The IETF has developed some techniques to support multicast routing protocols in the network layer. Some of these techniques have achieved significant progress and have become Internet standards. Therefore, we need to introduce these technologies to understand multicast technology and to consider them in our design.

#### 2.2.5.1 Addressing

In IPv4, a multicast group is identified by a single group address, which is a class D address (224.0.0.0 - 239.255.255.255). According to IANA address assignment, the address range 224.0.0.0 - 224.0.0.255 is reserved for routing protocols and other low-level topology discovery or maintenance protocols.

The range 224.0.1.0 - 238.255.255.255 is used for Globally-scoped (Internet-wide) multicast addresses. The address range 239.0.0.0 - 239.255.255.255 is used for Administratively-scoped (local) multicast addresses [15]. The address range 232.0.0.0 – 232.255.255.255 has been assigned to SSM. In IPv6, multicast addresses have a more complicated format and scope definition.

To dynamically allocate the multicast addresses, keep the address unique in specific scope, and reallocate used addresses, a multicast address allocation architecture is required that is generic enough to apply to both IPv4 and IPv6 environments. The Multicast Address Allocation Architecture (MALLOC) [RFC 2908] is a multicast address allocation architecture proposed by the IETF. The architecture is modular, with three layers, comprising a host-server mechanism, an intra-domain server-server coordination mechanism, and an inter-domain mechanism [16].

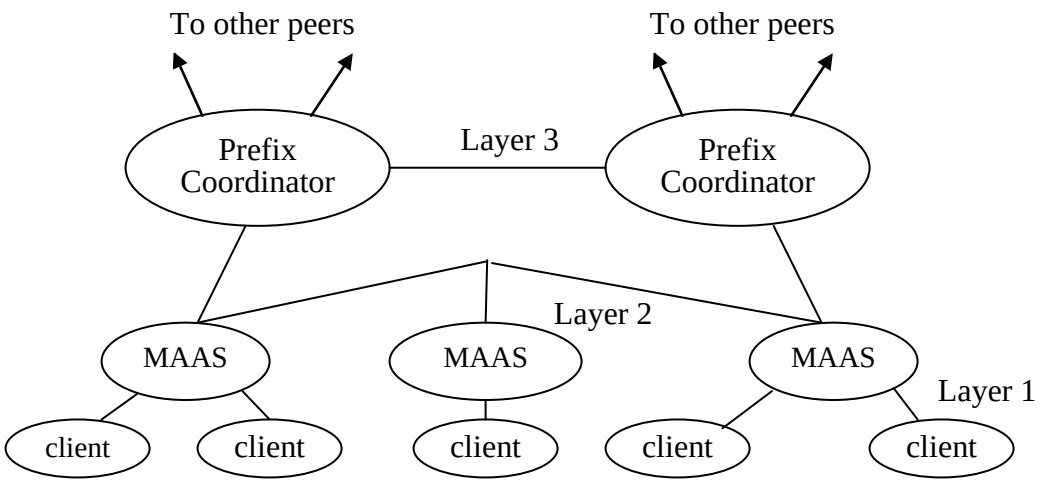



**Figure 4 Multicast Address Allocation Architecture (MALLOC)**

Layer 1: A protocol, e.g., MADCAP [17], or mechanism that a multicast client uses to request a multicast address from a multicast address allocation server (MAAS). When the server grants an address, it becomes the server's responsibility to ensure that this address is not then reused elsewhere within the address's scope during the lifetime granted.

Layer 2: An intra-domain protocol or mechanism that MAASs use to coordinate allocations to ensure they do not allocate duplicate addresses. A MAAS must have stable storage, or some equivalent robustness mechanism, to ensure that uniqueness is preserved across MAAS failures and reboots. MAASs also use the Layer 2 protocol/mechanism to acquire (from "Prefix Coordinators") the ranges of multicast addresses out of which they may allocate addresses.

Layer 3: An inter-domain protocol or mechanism, e.g., MASC [18], allocates multicast address ranges (with lifetimes) to Prefix Coordinators. Individual addresses may then be allocated out of these ranges by MAASs inside allocation domains as described above.

#### 2.2.5.2 Multicast Routing Information Base (MRIB)

PIM relies on an underlying topology-gathering protocol to populate a routing table with routes. This routing table is called the Multicast Routing Information Base (MRIB).

Multicast Routing Information Base is the multicast topology table, which is typically derived from the unicast routing table, or routing protocols such as MBGP that carry multicast-specific topology information. In PIM-SM, the MRIB is used to decide where to send Join/Prune messages. A secondary function of the MRIB is to provide routing metrics for destination addresses, these metrics are used when sending and processing Assert messages [7].

#### 2.2.5.3 Internet Group Management Protocol version 3 (IGMPv3) and Multicast Listener Discovery Protocol (MLD)

Internet Group Management Protocol version 3 (IGMPv3) is the protocol used by IPv4 systems to report their IP multicast group memberships to

neighboring multicast routers. Version 3 of IGMP adds support for "source filtering", that is, the ability for a system to report interest in receiving packets ‘only’ from specific source addresses, or from ‘all but’ specific source addresses, sent to a particular multicast address [11]. Similarly, the Multicast Listener Discovery Protocol (MLD) is used by IPv6 routers to discover the presence of multicast listeners (i.e., nodes that wish to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It provides the same “source filtering” features for IPv6 as IGMPv3 provides for IPv4.

#### 2.2.5.4 Automatic Multicast Tunneling (AMT)

AMT is a technology that allows multicast communication amongst isolated multicast-enabled sites or hosts in a multicast-incapable network, and also enables them to exchange multicast traffic with the native multicast infrastructure [41].

As shown in figure 5, AMT sites are hosts and networks with AMT support located in a multicast-incapable area. AMT gateways are hosts or site gateway routers using AMT pseudo-interfaces. The AMT interfaces are

points where multicast packets are encapsulated into unicast packets. AMT Relay routers are multicast routers configured to support transit routing between AMT sites and the native multicast backbone infrastructure.

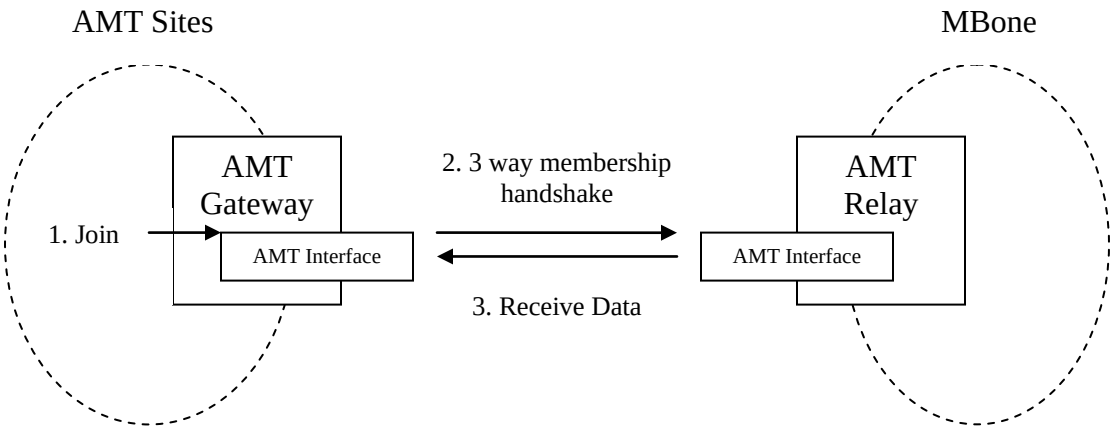



**Figure 5 AMT Topology and Messages**

First, a receiver at AMT site sends an IGMPv3/MLDv2 report joining (S1, G1). When the AMT Gateway receives the report, it originates an AMT Relay Discovery message addressed to the nearest AMT Relay Router. The closest AMT Relay Router receives the AMT Relay Discovery message and returns an AMT Relay Advertisement message. The AMT Gateway now can join the multicast group on behalf of the receivers by sending an AMT Membership Update message. Once the joining process is finished, multicast packets can be transferred from the AMT relay to the AMT Gateway [41].

The advantage of AMT is that no changes to a host stack or applications are

required, all protocols (not just UDP) are handled, and there is no additional overhead in core routers [41]. Now, AMT is designed to transfer SSM and ASM packets.

## 2.3 Overlay Multicast

In overlay multicast, hosts participating in a multicast session form an overlay network, and only utilize unicasts transmission between pairs of hosts (considered neighbors in the overlay tree) for data dissemination. The hosts in overlay multicast exclusively handle group management, routing, and tree construction, without any support from Internet routers. The key advantages overlays offer are adaptivity and ease of deployment. Overlays, however, impose a performance penalty over router-level alternatives. Generally, the average delay and the number of hops between parent and child hosts generally decrease as the level of the host in the overlay tree increases; and as hosts get closer to the root of the overlay tree, their contribution to stress of the link between the host and its directed-connected router grows [19].

Generally, the overlay multicast technologies are adaptable and incrementally deployable. However, they have some disadvantages,

including management complexity, no universal IP connectivity (hindered by NAT and firewalls), inefficiency, and information loss [20]. In this section, we will introduce overlay multicast in three fields, Ad Hoc networks, live stream, and reliable overlay multicast.

### 2.3.1 Multicast in Ad Hoc Networks

These networks inherit the traditional problems of wireless and mobile communications, such as bandwidth optimization, power control, and transmission-quality enhancement. In addition, their multi-hop nature and the possible lack of a fixed infrastructure introduce new research problems such as network configuration, device discovery, and topology maintenance, as well as ad hoc addressing and self-routing [21].

Multicasting in Mobile Ad Hoc Network (MANET) faces many challenges due to the continuous changes in network topology and limited channel bandwidth. Thus conventional multicast schemes designed for wire-line networks cannot directly apply. For typical applications, MANET is used to support close collaboration among team members. Thus, multicast support is critical and a desirable feature of ad hoc networks. Many multicast routing protocols have been proposed for MANET. For these protocols, robustness

and high overhead are key problems [22].

#### 2.3.1.1 AMRoute

AMRoute is an ad hoc multicast protocol that uses the overlay multicast approach. Bidirectional unicast tunnels are used to connect the multicast group members into a virtual mesh. After the mesh creation phase, a shared tree for data delivery purpose is created and maintained within the mesh. One member node is designated as the logical core, which is responsible for initiating the tree creation process periodically. The virtual topology can remain static even though the underlying physical topology is changing.

AMRoute needs no support from the non-member nodes, i.e., all multicast functionality and state information are kept within the group member nodes. The protocol does not need to track the network mobility since it is totally handled by the underlying unicast protocols. Other advantages are simplicity and flexibility. However, the advantages of overlay multicast come at the cost of low efficiency of packet delivery and long delay. When constructing the virtual infrastructure, it is very hard to prevent different unicast tunnels from sharing physical links, which results in redundant traffic on the physical links.

#### 2.3.1.2 Progressively Adapted Sub-Tree in Dynamic Mesh (PAST-DM).

In PAST-DM, the virtual mesh topology gradually adapts to the changes of underlying network topology in a fully distributed manner with minimum control cost. The multicast tree for packet delivery is also progressively adjusted according to the current topology. At the beginning, to construct the virtual mesh, each member node starts a neighbor discovery process using the expanded ring search (ERS) technique. Each member node keeps track of other members in its vicinity. In the PAST-DM protocol, each source constructs its own data delivery tree (a source-based tree) based on its local link state table. It supports dynamic membership in a simple and robust manner. When a node intends to join the multicast group, it starts with a normal neighbor discovery, and then exchanges link state tables with the neighbor.

#### 2.3.1.3 Location-Guided Tree (LGT)

LGT includes two position-based multicast protocols for groups of nodes modeled by complete unit graphs, in which the source of multicast messages and all destination nodes are within transmission radius of one another and aware of the geographic position of any other node in the group. In the

location-guided k-ary (LGK) algorithm, the sender node selects k nearest destinations as child nodes, groups the rest of the nodes to the k children according to close geometric proximity, and forwards a copy of the packet to each of the k child with its corresponding subtree as destinations. The process continues recursively with these children as new source nodes [23].

#### 2.3.1.4 Prioritized overlay multicast (POMA)

POMA proposes a model that improves the efficiency and robustness of overlay multicast in manets by building multiple role-based prioritized trees, possibly with the help of location information about member nodes. As with P2P networks, POMA forms a virtual network, consisting of only member nodes, on top of the physical infrastructure. Member nodes can form a short-term multicast group to perform certain important tasks. Overlay trees can have different levels of priority depending on the importance of the service they perform. This approach avoids the need to change the application layer tree when the underlying network changes [24].

### 2.3.2 Live Stream

Supporting live stream is another topic in overlay multicast research, which aims to support live video and audio streams for end users. The current proposed overlay multicast protocols for live streams are focused on routing

and QoS problems. Generally, the networks for live stream consist of three stages of nodes. The nodes in the first stage are the sources where live streams originate. A source forwards each of its streams to one or more nodes in the second stage, which are called reflectors. A reflector can split an incoming stream into multiple identical outgoing streams, which are then sent on to nodes in the third and final stage, which are called the sinks. There are two bottlenecks in the model: server bottleneck and network bottleneck. The requirements of the overlay network are minimum cost, capacity, quality, and reliability [25].

Problems with current multicast technologies for live stream: 1. few of the routers on major backbones are configured to participate in the multicast protocols, so as a practical matter it is not possible for a server to rely on multicast alone to deliver its streams. 2. Multicast trees are not very resilient to failures. If a node or link in a multicast tree fails, all of the leaves down-stream of the failure will lose access to the stream. While the multicast protocols do provide for automatic reconfiguration of the tree in response to a failure, end users will experience a disruption while reconfiguration takes place [25].

#### 2.3.2.1 End System Multicast (ESM)

End System Multicast is designed for enabling small and medium sized group communication applications on the Internet. First, ESM constructs a richly connected graph called a mesh. When a node wishes to join a group, ESM assumes that the member is able to get a list of group members by an out-of-band bootstrap mechanism. The node randomly selects a few group members from the list available to it and sends them messages requesting to be added as a neighbor. It repeats the process until it gets a response from some other member. After join, a node starts to exchange state information with it neighbors, and to achieve a high degree of robustness, every member maintains a list of all other members in the group. In the second step, ESM constructs spanning trees on the mesh, each tree rooted at the corresponding source using a distance vector algorithm [26].

ESM uses a peer-to-peer (P2P) scheme to distribute data from the source to receivers, in which receivers not only receive data from other nodes, but also contribute their bandwidth to release sender's burden by sending received data to other nodes that require it. In ESM, the source splits the video stream into m strips using Multiple Description Codec (MDC), and multicasts each strip along a separate tree [27]. A node will join at least one tree, which

guarantees that the node can receive at least a low quality video stream. The more the node can contribute bandwidth to others, the more trees it can join to receive a higher quality video stream.

#### 2.3.2.2 PeerCast

PeerCast is a tree-based overlay network called PeerCast that uses clients to forward the stream to their peers. PeerCast is designed as a live-media streaming solution for peer-to-peer systems that are populated by hundreds of autonomous, short-lived nodes. A new node n seeking the live stream needs to be able to discover an unsaturated node in the multicast group. The node n contacts the source r of the stream at the known URL. If r is unsaturated, it accepts n as its child and establishes a data transfer session with n. Otherwise, r redirects n to one of its immediate children a. Then, a attempts to set up a data-transfer session with n. The process continues iteratively, until n gets accommodated. If n is unable to find an unsaturated node within some specified number of tries, the peering layer flags a resource unavailable error to the upper application-layer [17].

### 2.3.3 Reliable Overlay Multicast

#### 2.3.3.1 PRM (Probabilistic Resilient Multicast)

PRM uses two simple techniques: 1) For neighbor discovery, a proactive component called Randomized forwarding in which each overlay node chooses a constant number of other overlay nodes uniformly at random and forwards data to each of them with a low probability (e.g., 0.01- 0.03). This randomized forwarding technique operates in conjunction with the usual data forwarding mechanisms along the tree edges, and may lead to a small number of duplicate packet deliveries. Such duplicates are detected and suppressed using sequence numbers. The randomized component incurs very low additional overheads and can guarantee high delivery ratios even under high rates of overlay node failures. 2) A reactive mechanism called Triggered NAKs to handle data losses due to link errors and network congestion [29].

#### 2.3.3.2 ALMI

ALMI is tailored toward support of multicast groups of relatively small size (several 10s of members) with many to many semantics. Participants of a multicast session are connected via a virtual multicast tree, i.e., a tree that consists of unicast connections between end hosts. The tree is formed as a minimum spanning tree (MST), where the cost of each link is an application specific performance metric. An ALMI session consists of a session

controller and multiple session members. The multicast tree is a shared tree amongst members with bi-directional links. The minimum spanning tree calculation is performed at the session controller and results are communicated to all members in the form of a (parent, children) list [30].

#### 2.3.3.3 Overcast

Overcast is designed by Cisco. An Overcast system is an overlay network consisting of a central source (which may be replicated for fault tolerance), any number of internal Overcast nodes (standard PCs with permanent storage) sprinkled throughout a network fabric, and standard HTTP clients located in the network. Overcast provides large-scale, reliable multicast groups, especially suited for on-demand and live data delivery. The key requirement of Overcast supports single source distribution of bandwidth-intensive media. It uses URLs as a namespace for Overcast groups. The goal of Overcast's tree algorithm is to maximize bandwidth to the root for all nodes. At a high level the algorithm proceeds by placing a new node as far away from the root as possible without sacrificing bandwidth to the root. The tree protocol begins when a newly initialized node contacts the root of an Overcast group. The root thereby becomes the *current* node. Next, the new node begins a series of rounds in which it will attempt to locate itself

further away from the root without sacrificing bandwidth back to the root. In each round the new node considers its bandwidth to *current* as well as the bandwidth to *current* through each of *current*'s children. If the bandwidth through any of the children is about as high as the direct bandwidth to *current*, then one of these children becomes *current* and a new round commences. In the case of multiple suitable children, the child closest (in terms of network hops) to the searching node is chosen. If no child is suitable, the search for a parent ends with *current* [31].

#### 2.3.3.4 Reliable Multicast for Heterogeneous Networks (RMX)

RMX proposes a hybrid approach to reliable multipoint communication that leverages well-understood and robust reliable unicast transport protocols and couples them with the multicast service model for efficient multi-point data delivery. Its architecture is grounded in a hybrid communication model that partitions the heterogeneous multicast receiver set into a number of small homogeneous data groups, and uses robust unicast communication protocols across data groups. The architecture relies on three key concepts. First, in order to localize the hard multicast problems of scalable loss recovery, congestion control and bandwidth allocation, it partition the large wide-area heterogeneous session into many smaller and simpler homogeneous sub-

sessions. This divide-and-conquer approach effectively decouples each sub-session from the vagaries associated with the rest of the session participants. Second, as data flows through an RMX, the RMX uses application-level knowledge to dynamically alter the content of the data or to adapt the rate and ordering of data objects. The RMX allows for the notion of semantic reliability as opposed to data reliability, that is, reliability of information rather than that of the representation of the information. Thus, by relaxing the semantics of reliability, we lift the constraint that all receivers advance uniformly with a sender's data stream; each receiver defines its own level of reliability and decides how and to what degree individual data objects might be transformed and compressed. Finally, to support these semantics, it uses the Application Level Framing (ALF) [32] protocol architecture, which says that application performance can be substantially enhanced by reflecting the application's semantics into the design of its network protocol.

#### 2.3.3.5 ROMA (Reliable Overlay Multicast Architecture)

The primary set of target applications is applications requiring reliability and high bandwidth, such as delivery of large files. ROMA is a TCP-based content delivery architecture. ROMA enables multiple-rate reception, with individual rates that match the end-to-end available bandwidth along the path, while using small buffers at application-level relays, and the standard

TCP protocol. It applies a forward-when-feasible approach, whereby each intermediary forwards only those received packets to downstream hosts that can immediately be written into the downstream TCP socket. It handles reliability at the application layer using erasure resilient codes, also known as fast forward error correction (FEC) codes. Overlay multicast typically incurs a performance penalty over IP multicast, due to factors such as link stress, suboptimal routes, increased latency, and end-host packet processing. Since IP does not provide the "best" path, measured in terms of delay or loss rates, ROMA finds that the best alternative TCP path is often a multi-hop path in which the minimum expected TCP throughput along any overlay hop is maximized [33].

PRM, ALMI, Overcast, and RMX all address the issue of reliability in distributing content to end hosts. PRM was designed for applications that do not require perfect reliability and focuses on improving the rate of data delivery while maintaining low end-to-end latencies. ALMI and Overcast employ TCP to provide reliable file transfers between any set of hosts. However, ALMI uses a back-pressure mechanism to rate-limit the sender, resulting in a single rate control. Overcast was explicitly designed with the goal of building distribution trees that maximize each node's throughput

from the source. However, the technical focus of Overcast was exclusively on topology optimization, and they did not consider issues associated with the transport protocol. Other works have also focused on the problem of efficient tree construction and on the challenges of optimizing the tree layout so as to minimize network costs such as average latency; or to minimize overlay costs, such as link stress; or to perform load balancing, such as by bounding the maximum fanout [33].

## 2.4 Session Layer

In the session layer, there are two different viewpoints on the session management issue. One viewpoint says that multicast is simply a transport mechanism that provides end-to-end delivery. All of the other services, including security, encryption, reliability, session advertisement, monitoring, billing, etc., are application-layer services that must be provided by each particular application [34]. Another viewpoint regards session management as one of the crucial protocol components in multicast protocols. The first viewpoint only considers the data forwarding functionality of multicasting and assumes that multicasting works like some unicast protocols, e.g., TCP and UDP. It ignores an important fact that the service model of multicast is completely different from the service model for unicast. Membership and

hierarchy management must be included in the multicast to handle a large number of receivers. As we discussed above, some functionalities, including security and AAA, must be added to the multicast service model. Otherwise, multicast cannot be accepted and deployed on the Internet.

Up to now, little work has been done for session management. Its functionality has not been explicitly defined and specified. In RMTP-II, the author has mentioned the importance of session management in reliable transport multicast, but left this issue to other research groups. A South Korean research group proposed a session management mechanism [35], but its model is too simple to reflect a real multicast session and only includes some transport layer functions. It ignored other important functions, such as security and billing functions.

# 3 Requirements

In this chapter, we will summarize and analyze the requirements of multicast technology. We define a life cycle of multicast sessions based on the requirements. The discussion in this chapter will lead us to a general solution for multicast, which we will propose and introduce in the next chapter.

## 3.1 Assumptions and Awareness

As for the next-generation Internet, multicast can benefit a number of applications, from different categories, e.g., multimedia conferencing, distance learning, multi-player games, news headlines, stock quotes, weather updates, etc. Streaming video to hundreds or thousands of listeners is a newer application where the Network Service Provider (NSP) and content provider (CP) can achieve very large savings in resource requirements through the use of multicast data distribution [42].

However, as we introduced in the previous chapter, many drawbacks of current multicast technology, e.g., lack of security, access control, scalability, etc., prevent the NSP and the CP from generating revenue from multicast use. For example, in stream video application, both sender and receivers

must be authenticated before joining the group, and the stream must be encrypted to ensure that only legitimate customers can receive it. A demand for a more general solution for multicast technology arises from these applications.

In this paper, our focus is the subset of multicast technologies that can be widely used in commercial applications. After studying the current multicast technologies, we can summarize some general features of the multicast techniques and applications in commercial usage, e.g., where and how the multicast technology will be used in commercial applications, and its relationships with other technologies:

- In commercial usage, multicast techniques are only valuable in large scale applications with a very large number of participants, since small scale applications can be satisfied by unicast or other techniques.
- In a commercial scenario of multicast application, there are a few senders and a large number of receivers (thousands or millions).
- In the core network, there are a few routers that are multicast capable. Most routers in local area networks are not multicast capable. The routers in local area networks will gradually become multicast capable.

- The supporting techniques for multicast will be available in the future, e.g., DNS and group addressing.
- There are many different kinds of multicast techniques used in different domains.
- In commercial usage, ISPs and content providers prefer multicast techniques that are easy to use, profitable, and manageable.
- Content providers and end users are not interested in the details of multicast techniques, and they only care about the quality of services. ISPs are very interested in the details of multicast techniques.

Some of the features are assumptions we made based on our knowledge and experience, and others are realities in multicast techniques. These assumptions and awareness are the starting points of the following discussion, and they can also limit the problem space for our discussion. They can help us to get a better understanding of multicast technologies in commercial usage.

## 3.2 Requirements for Multicast Technology

Based on the above introduction of current multicast technologies and assumptions we discussed, we can summarize the requirements for the multicast technology. These requirements include what users (ISP, content providers, and end users) expect from multicast technology, and what are essential for success of multicast technology in commercial usage. Our discussion and solution are significantly based on the requirements we introduce here.

The requirements for multicast technology can be clustered into six groups: data delivery, scalability, security, group management, reliability, and deployment. Each group has several specific requirements.

In data delivery, efficiency and robustness are the most important requirements, so we need to pay great attention to them.

- The efficiency of data delivery is the economical usage of network resources for tasks in a multicast technology, e.g., bandwidth, packet numbers, etc.
- The robustness is the stability of data delivery in the face of a user joining or leaving, node failures, and other condition changes. The

events of membership and topology changes may seriously affect the data delivery in a subset of member nodes. A problem faced by multicast protocols is the heterogeneous nodes in the group. The network bandwidth and end system's receiving capacity (e.g., CPU and bus speed) are quite different from node to node. The difference between node capacities can significantly affect the services provided by their upstream nodes. For example, when node A talks to B in unicast, the performance is limited by one path. What can be done to improve the throughput (or delay bound) is done by IP (for example, load sharing the traffic over multiple paths). In reliable multicast, when A talks to B, C, D, E, or F, should the throughput or delay be that sustainable by the slowest or average [37]? The robustness is the overall performance of a multicast group when membership and topology changes happen.

The scalability of multicast technology is its capability of reach geographic and administrative coverage as large as possible. The scalability requirements of multicast technology include:

- Support for large number of receivers: the number of receivers of a multicast session can be thousands or even millions. The enormous

number of members can affect topology establishment, data delivery, and many other aspects of multicast techniques. Support for a large number of receivers is the crucial scalability requirement for multicast technologies.

- Large geographical coverage: in commercial usage of multicast technology, the receivers will not only be a large number but also located in a very large geographical area. The distribution of receiver locations means that they are located in different administrative areas and have different distance to the senders, which multicast technologies should handle.
- Inter-domain capability: is the capability of working across different autonomous systems (AS), which is important for multicast technologies, as we discussed above.
- Collaboration with heterogeneous distribution technologies, e.g., other multicast protocols: according to our introduction in the last chapter, there are a lot of different multicast protocols developed and proposed. Some of them have already been deployed in some domain. Connection and collaboration with other multicast protocols can easily extend the coverage of a multicast technique and obviously enhance

its chance of success. Some research groups have already been aware of the importance of dealing with heterogeneity in multicast techniques. There are some other distribution techniques that are worthy to collaborate with by multicast, e.g., Peer-To-Peer (P2P), in order to get a larger coverage of audiences and applications.

- Support for multiple groups: some multicast applications and groups may have similar geographic coverage and topology. To build and manage different distribution topologies for these applications and groups is a great waste of network resources. If a multicast technique can support different multicast sessions or groups based on a single infrastructure and allow sharing among them, it will improve the scalability and save a lot of resources.

The security requirements include the data confidentiality, data source authentication, and multicast policy representation. They are the foundation of security and AAA (authentication, authorization, and accounting) mechanisms. Because the security is not my focus in this project, we will not discuss it in detail in this paper. The security of multicast is a topic in our research group and has been investigated by my colleague, Mr. Ritesh

Mukherjee [43]. AAA mechanisms have been investigated by my colleague, Mr. Salekul Islam [44].

According to the features of end users and network connection changes, the group membership changes in a multicast session are significant and almost constant. The capability of effectively and efficiently managing a group is one of key features for multicast technology. The requirements of group management for multicast technology include:

- Ability to name groups: each multicast group should be able to be located by a unique address or identifier. Multicast technology should have the capability of dynamically allocating the identifiers or addresses, avoiding name collisions, and manage and maintain the identifiers or addresses. The known best solution is MALLOC, which is introduced in the previous chapter.
- Dynamically and automatically creating/terminating a group: after allocating an address to a group, multicast technology should be able to establish the group. The creation of a multicast group should include assignment of each node's role and functionalities, mechanism of binding nodes into a hierarchical topology, announcement of the group's existence, etc. When the data transfer is

over, multicast technology should terminate the group, including announcing the end of transfer and releasing resources on network and each node,

- Dealing with membership changes, e.g., member join/leave etc: dealing with the highly dynamic membership changes is one of the most challenging problems in the development of multicast. The enormous number of member nodes and their constant joining and leaving make the membership and topology of the group in an endlessly unstable status. Multicast technology should be capable to effectively and efficiently manage the membership and keep the effects of membership changes on other nodes to the minimum level.

In commercial applications, the group management may need support from other techniques, e.g., AAA and security mechanisms. In the meantime, the group management supports many other techniques, e.g., data distribution and reliability. The relationship between group management and other techniques is a comprehensive and pervasive topic in multicast technology.

The reliability of multicast technology specifies guarantees that the multicast technology can provide with respect to the delivery of messages to the

receivers. The Reliability requirements mainly concern the data delivery and quality of service on receiver side, and include:

- 100% reliability with no time bound: this is the highest level of reliability, and may require the longest time to transfer data because of the potential error recovery and retransmissions. It is suitable for transferring files and crucial data.
- Reliability suitable for live stream applications: according to features of many real-time applications, time bound must be considered. Low delay is the most important goal in this case, and users are willing to accept reasonable loss to achieve the lowest delay.

There may other kinds of applications that may need no specific reliability, and the best-effort data delivery can satisfy their requirements. Such kinds of application are not our focus of reliability discussion in this paper.

As we can see, different applications have very different requirements for reliability provided by multicast technology. There is no single solution that can meet all of the requirements. All we can do is to provide a general infrastructural facility that can support techniques meeting the reliability requirements.

The above five sets of requirements are mainly concerned with capability of multicast technology. There is another set of requirement relating to deployment, which highlights the relationship between multicast technology and the operational and administrative environment where multicast technology will run. The deployment requirements include:

- Working with different underlying hardware and software.

Multicast technology may need to work on all kinds of hosts, e.g., personal computer and commercial routers. The multicast technology has to adapt itself to the available network hardware and software on these hosts. Some hosts may have native IP multicast software and hardware, and others may be in a ‘dumb’ network (TCP/UDP only). Dealing with the heterogeneity of underlying hardware and software is an important fact that can affect the deployment of multicast technology.

- Ease of deployment (incremental deployment).

Nowadays, due to immaturity of IP multicast technology, some ISPs do not allow multicast traffic to go through their routers. There are a lot of old-fashioned non-multicast-capable routers still being deployed and used on some LANs. Therefore, we will encounter a problem that the core network may be multicast-capable but most edge routers on the Internet may be

multicast-incapable. As long as multicast technology becomes more efficient and mature, multicast technology may gradually be accepted from core network routers to edge routers. Before that, multicast services have to collaborate with both multicast and unicast routing protocols.

As we introduced in the last chapter, the overlay multicast mostly works in the application layer and is suitable for network environments without or with little router support. It sacrifices some performance over IP multicast to achieve fast deployment. Some of them have already achieved significant progress and even been used in some commercial applications. However, IP multicast has its advantages in performance and scalability. As the multicast service market grows, more and more multicast-capable routers may be deployed. The multicast technology may become a combination of overlay multicast and IP multicast. Finally, the IP multicast may be fully deployed on the Internet.

As a result, multicast technology needs to be suitable for this incremental deployment of multicast-capable routers.

- Customizability

Generally, the requirements of data delivery, scalability security, group management, and reliability represent a combination of functionalities that multicast technology should have. To provide services to customers and productively manage services, ISPs would like to be capable of adding or removing modules to and customize these functionalities in their operation of multicast techniques. Multicast technology should provide its users an interface that can be easily used to add new modules, remove undesired ones, and control their operational details.

- Flexibility

Along with the development and deployment of multicast technology, many requirements of multicast technology may vary in the future, e.g., security and group management, and new requirements may be added. This fact necessitates that multicast technology can be easily changed according to the constantly changing requirements. The flexibility requirement is mainly concerned with the extensibility of multicast technologies for future growth.

After summarizing the requirements, let us take a look at the current multicast protocols we introduced in the last chapter. In the following tables, we evaluate and compare the protocols based on requirements of multicast

technology, in network layer protocols, transport layer protocols, and overlay multicast protocols. Please note that we do not evaluate and compare all overlay multicast protocols because we do not have sufficient sources of details for those protocols. As a result, we only compare the common features in three different groups: Ad Hoc, live stream, and reliable overlay multicast protocols.

| | | Mbone / DVMRP | MOSPF | PIM-DM | CBT | PIM-SM | SSM |
|---|---|---|---|---|---|---|---|
| Data delivery | Efficiency | yes | yes | yes | yes | yes | yes |
| | Robustness | yes | yes | yes | yes | yes | yes |
| Scalability | Large number of receivers | not efficient, flooding routing algorithm | not efficient, flooding routing algorithm | yes | yes | yes | yes |
| | Large geographical coverage | no | no | no | yes | yes | yes |
| | Inter-domain capability | no | no | no | no | with difficulty, need help of MBGP, etc. | with difficulty, need help of MBGP, etc. |
| | Collaboration with heterogeneous distribution technologies | no | no | no | no | no | no |
| | Support for multiple groups | no | no | no | yes | yes, but only works well with a single RP | yes, but only works well with a single RP |
| Security | | no | no | no | no | no | no |
| Group management | Ability to name groups | no | no | MALLOC for native IP multicast | MALLOC for native IP multicast | MALLOC for native IP multicast | MALLOC for native IP multicast |
| | Dynamically and automatically create/terminate a group | no | no | yes, with help of other protocols, e.g., MALLOC | yes | yes, with help of other protocols, e.g., MALLOC | yes, with help of other protocols, e.g., MALLOC |
| | Dealing with membership changes | no | yes, using IGMP | yes, using IGMP | yes, using IGMP | highly dynamical | highly dynamical |

| | | Mbone / DVMRP | MOSPF | PIM-DM | CBT | PIM-SM | SSM |
|---|---|---|---|---|---|---|---|
| Reliability | 100% reliability without time bound | no | no | no | no | no | no |
| | Reliability suitable for live stream applications | yes | yes | yes | yes | yes, best efforts | yes, best efforts |
| Deployment | Working with different underlying hardware and software | yes, IP multicast tunnel | only within multicast-capable domains | only within multicast-capable domains | only within multicast-capable domains | only within multicast-capable domains | only within multicast-capable domains |
| | Ease of deployment | no | no | no | no | difficult, ISPs are reluctant to accept it | difficult, ISPs are reluctant to accept it |
| | Customizability | no | no | no | no | difficult to add new functions, e.g., security and AAA have to be added as upper layer functions | difficult to add new functions, e.g., security and AAA have to be added as upper layer functions |
| | Flexibility | no | no | no | no | no | no |

**Table 1 Requirements Comparison of Multicast Protocols in Network Layer**

| | | LGMP | RMTP |
|---|---|---|---|
| Data delivery | Efficiency | yes | yes |
| | Robustness | yes | yes |
| Scalability | Large number of receivers | yes | no |
| | Large geographical coverage | yes | yes |
| | Inter-domain capability | yes, logic links | yes |
| | Collaboration with heterogeneous distribution technologies | no | no |
| | Support for multiple groups | yes | yes |
| Security | | no | no |
| Group management | Ability to name groups | no | no |
| | Dynamically and automatically create/terminate a group | yes | yes, but a single level hierarchy |
| | Dealing with membership changes | yes | yes |
| Reliability | 100% reliability without time bound | yes | yes |
| | Reliability suitable for live stream applications | yes | yes |
| Deployment | Working with different underlying hardware and software | yes | yes |
| | Ease of deployment | yes | yes |
| | Customizability | no | no |
| | Flexibility | no | no |

**Table 2 Requirements Comparison of Multicast Protocols in Transport Layer**

| | | Overlay Multicast in Ad Hoc network | Overlay Multicast for Live Stream | Reliable Overlay Multicast |
|---|---|---|---|---|
| Data delivery | Efficiency | no | no | no |
| | Robustness | no | no | no |
| Scalability | Large number of receivers | no | no | no |
| | Large geographical coverage | no | yes | yes |
| | Inter-domain capability | no | yes | yes |
| | Collaboration with heterogeneous distribution technologies | no | no | no |
| | Support for multiple groups | no | no | no |
| Security | | no | no | no |
| Group management | Ability to name groups | no | Application level group ID, no unique Internet-wide ID | no |
| | Dynamically and automatically create/terminate a group | no | no | no |
| | Dealing with membership changes | yes | yes, but sacrifice performance | yes |
| Reliability | 100% reliability without time bound | no | no | no |
| | Reliability suitable for live stream applications | yes | yes, best effort for live streams | yes |
| Deployment | Working with different underlying hardware and software | no | no | no |
| | Ease of deployment | yes | yes, because no router support is required, but sacrifice performance to achieve it | yes |
| | Customizability | no | yes | yes |
| | Flexibility | no | no | no |

**Table 3 Requirements Comparison of Overlay Multicast Protocols**

According to the above tables, we can conclude some common features that current multicast technologies have with respect to the requirements we summarized in the chapter.

For data delivery, IP multicast has better performance than overlay multicast, both in efficiency and robustness. The main reason for it is that most IP multicast techniques usually have better infrastructure and router support for data transfer, and overlay multicast techniques rely on their hosts' capability to fulfill the data transfer.

For scalability, in IP multicast, early dense mode techniques, e.g., MBone/DVMRP and MOSPF, only have small group size and coverage due to the inefficient flooding routing protocols, which also lack inter-domain capability. The intra-domain protocols, both dense mode and sparse mode, improve the coverage and group size, but cannot work across boundaries between different autonomous systems (AS). The inter-domain IP multicast protocols will have better coverage than intra-domain IP multicast protocols. Because of the feature of source-based tree, most dense mode cannot support multiple groups in a single topology. Most sparse mode multicast protocols use a shared tree, which make it possible that different senders can share the

same root node and the tree structure. However, the root can be the bottleneck in data distribution and group management. In overlay multicast, there is usually no router support or centralized management, and group management has to be inefficient, trivial, and distributed among users. The overlay multicast protocols usually use source-based tree. Therefore, overlay multicast techniques have a large coverage and inter-domain capability, but support smaller group size and cannot support multiple groups. So far, there is no multicast technique that can work with other multicast techniques.

For group management, current IP multicast protocols will use IP address allocation protocols, e.g., MALLOC, to identify multicast groups. With IP multicast protocols based on shared trees, it is possible to dynamically create and terminate groups. Overlay multicast protocols do not have internet-wide unique identifiers for groups and use application level identifiers. For dealing with membership changes, IP multicast protocols will be better than overlay multicast protocols because overlay multicast protocols usually use distributed group management mechanism and sacrifice performance of group management and data delivery to get easier deployment.

For reliability, both network layer IP multicast protocols and overlay multicast protocols need help from other techniques, e.g., transport layer multicast protocols, to achieve 100% reliability. All multicast protocols can be used for live stream applications, but may need a QoS mechanism built on them.

Overlay multicast protocols usually are easier to deploy than IP multicast because they do not need router support and only work on end systems. Because overlay multicast is working in the application layer, new functionality is easier to be added as new modules and be customized compared with IP multicast. However, overlay multicast protocols may have some hardware and software limits because of their application-based natures. Currently, IP multicast protocols have to work within multicast-capable domains and are relatively difficult to add and customize functionalities. Until now, no multicast techniques are designed in a very flexible way.

According to above comparison and discussion, we can discover the problems of current multicast technologies, which make them commercially infeasible.

The first problems are lack of access control and inability to meet the requirements of security. There are no existing multicast protocols that can independently support identify group members, authorize members in a group, charge user for their usage, or guarantee the data confidentiality. Generally, the current distribution model for multicast is “anyone can send, and anyone can receive”. For a closed system, e.g., a network in a university or a company, or a friendly system, e.g., the original Internet, this model is feasible. However, given the transition to the commercial Internet, it becomes infeasible because an ISP cannot generate revenue by charging content providers and receivers for their usage of a multicast session.

Another problem is that current multicast protocols cannot meet scalability requirements. Although IP multicast protocols can support a large number of receivers, large coverage, and inter-domain capability, they cannot transfer data to domains without multicast capability. Overlay multicast does not need router support, but it cannot support large groups. Some research groups have realized that connecting different multicast protocols can extend scalability of current multicast protocols significantly. However, there is no existing method to coordinate different multicast protocols.

The third problem is the reliability. The network layer IP multicast protocols cannot provide reliability independently, and need support of reliable multicast protocols. However, there is no existing method that can coordinate multicast protocols in network layer and transport layer seamlessly, or allow an ISP to choose different flow control schemes for different reliability levels. Because of lack of reliability, ISP cannot guarantee the quality of service (QoS) for end users, and content providers and end users will not choose multicast.

The unawareness of deployment requirements for multicast technology in current multicast protocol design makes the acceptance of IP multicast even more difficult. Most IP multicast protocols are not capable of incremental deployment and cannot go through the domains where ISPs do not accept IP multicast techniques. Although overlay multicast protocols are easier to be deployed than IP multicast, they achieve it at the cost of performance. Furthermore, lack of customizability and flexibility make current multicast technology unable to meet ISPs' various requirements in their commercial activities.

## 3.3 Session Life Cycle

Before we can propose any solution for problems in current multicast technology, which can meet most of the requirements in the last section, we should find out how the requirements are related to each other.

Generally, all activities of a multicast group happen in a multicast session, which may have a series of phases in order to accomplish the data transmission in the group. Different requirements will affect each other in each phase of a multicast session. For example, the group membership management ability of a multicast technology will have a critical impact on its scalability when facing a large number of receivers. Each requirement will affect one or more phases of a multicast session, and all requirements can be mapped to the phases of a multicast session. Therefore, we can analyze and satisfy the requirements by studying the features of multicast sessions.

Although there is not a single multicast protocol that can meet all requirements of multicast applications, we can find a generic procedure that most reliable multicast sessions should follow. In this procedure, we need to

consider all aspects of a multicast session, e.g., session creation, authentication, security, congestion control, termination, etc. Derived from the generic procedure of multicast sessions, we can define a model of multicast session life cycle. Based on this model, a session management mechanism can be created. With support for session management, we can find ways to meet the requirements that we defined above. Now, we will look at the procedure as follows.

The first step of a session is to create a session. In this step, a content provider (CP) that is trying to create a new session should inform its intention to some Internet service providers (ISP) that are capable of organizing multicast groups. ISPs will prove the CP's rights of initiating the new session, configure some service nodes to support the new session, and reserve the resources for the group. In a multicast topology, service nodes are the nodes that can receive and forward data, aggregate and forward control information, and even manage group membership and other functions. If the sending message should be charged for, ISPs can use authentication, authorization, and accounting (AAA) function on service nodes to manage the group and calculate relative costs for this sender. After authentication, a naming service server should try to allocate a multicast

address for this session. Now, the session information is available on the service nodes.

After a session is created, the session information should be announced on the Internet as broadly as possible, so potential receivers can know the existence of the session. There are many out-of-band methods of session announcement, e.g., Session Announcement Protocol (RFC 2974) [38], E-mail, online bulletin board, web-based merchant, etc. Receivers need to know the existence of a new session and session information by these means before they can join the session. If service nodes close to receivers are aware of the information about multicast session, it can help the establishment of a session tree by shortening receivers' joining process. Therefore, sending session announcement to some service nodes even before the session tree is built can be another helpful step in a session.

Now, the session is ready for receivers to join. We need a session topology auto-configuration mechanism to help sender and receivers join the group. The sender can join the session topology by connecting to a service node assigned by ISP. A receiver first finds out the session information by some out-of-band ways and then informs service nodes about its interest in a

specific session. The receiver can use IGMPv3 [8] or other mechanism to talk to the service nodes. The next thing is to find a proper parent node for a receiver. This task is the main goal of most multicast routing protocols and should be completed with the help of service nodes and other receivers. After the selection of the parent node, the receiver can try to join the session topology and all communication activities in this session by binding to the selected parent node. During the procedure of receiver joining, some intermediate service nodes can also join at the same time in order to build the topology, e.g., designated nodes in RMTP-II.

For a secure multicast session, the service nodes should check the sender and receiver's identity (authentication), check sender and receivers' right for this multicast session (authorization), and use an accounting function to monitor this sender and receivers' account balance (accounting). The sender authentication can avoid the situation of a notorious sender creating an illegal session. The service nodes will reject receivers if the receivers have no rights to join the group or their account balances cannot afford the cost of multicast traffic.

After AAA checking, service nodes should allocate an encryption key for

child nodes. The key is used for encrypting and decrypting the data stream. The key management mechanism in service nodes should generate keys and distribute to the child nodes.

After the keys are generated and distributed, the service nodes should forward the data packets received from the upstream nodes to their children nodes. In the sender and service nodes, the data packets will be encrypted for a secure multicast session. The receiver nodes will decrypt received packets by the key distributed by its parents.

The data stream will be managed by a flow control mechanism. The flow control mechanism may need to check the received packets, buffer some packets for potential retransmissions, aggregate error reports, request missing packets from upstream nodes, forward requested packets to downstream nodes, and so on. The flow control mechanism should have QoS services in end systems (receivers) for multicast sessions with reliability requirements. The QoS services will monitor the status of nodes' network connection and submit the QoS reports to upper layers.

The session topology is constantly changing because receivers frequently

join and leave the session. It needs the topology auto-configuration mechanism to monitor the membership changes and to optimize the topology dynamically. The optimization of a session topology involves not only dynamical changes in the hierarchical topology of a session to adapt to changes of membership and network loads, but also managing the data flow to obtain the desired reliability. Many cases can lead to optimizing the topology: 1) nodes join and leave, 2) service nodes rejects some nodes due to incapability of handling with these nodes, 3) received data stream cannot meet pre-set QoS requirements, 4) service nodes reject nodes according to periodic membership and accounting checks, and 5) underlying network topology changes (e.g., network connection and node failure).

When the session is over, we need a mechanism to terminate it. Without such a mechanism, all nodes have to maintain session state information and resources (e.g., memory and bandwidth) until they can positively detect the session termination by other means, e.g., QoS changes or node membership changes. It is a time-consuming and misleading process. The termination process can be initiated by the sender or services node and propagate gradually to receivers. All nodes should release all resources allocated for the session and clean up the session state information, e.g., session topology

information, membership information, and buffers used for this session. The session termination mechanism should also release the multicast address of the session for reuse. It also needs to inform all nodes bound to the session about the session termination and update necessary accounting information.

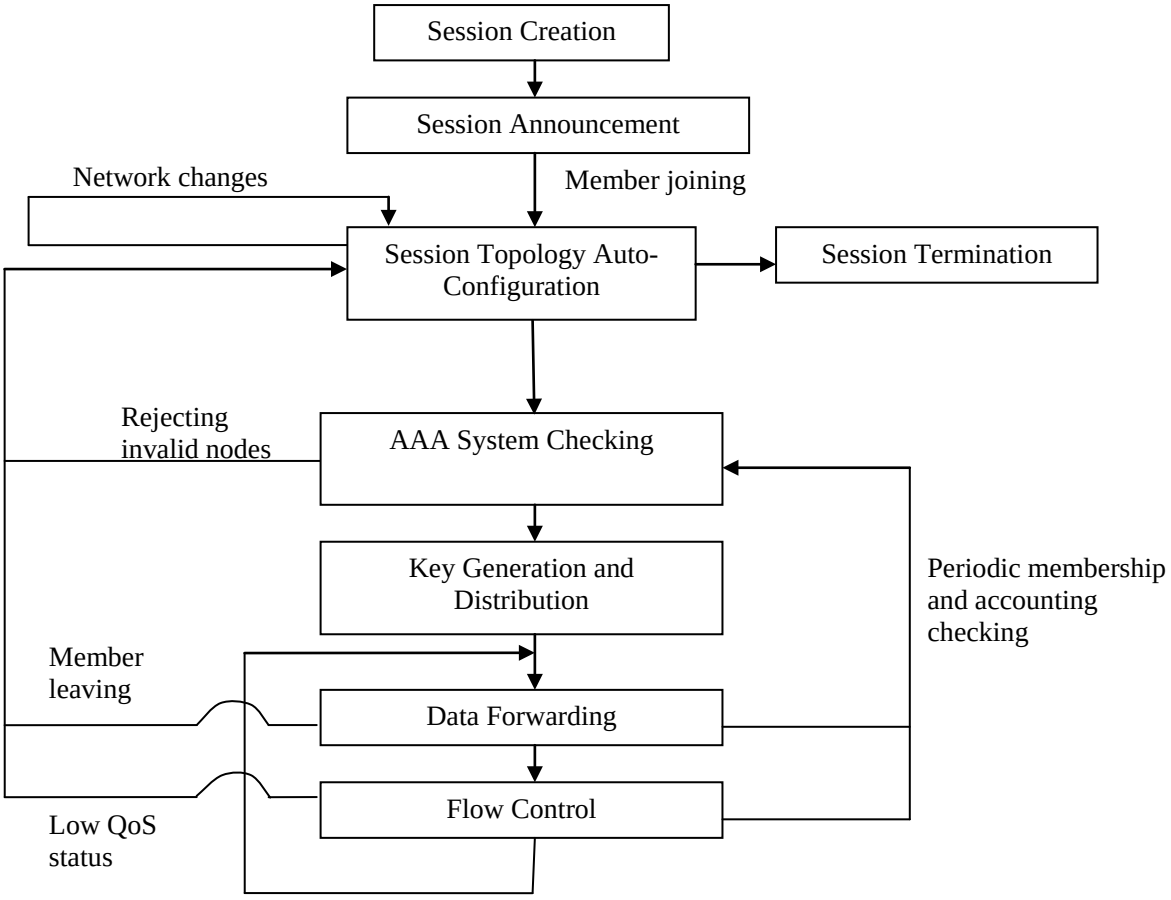



Figure 6 Multicast Session Life Cycle

According to the above discussion, we can define a model for multicast session life cycle, as shown in figure 6. Different multicast sessions may leave some stages of this model out, e.g., some multicast session may not need AAA and security. However, this model reflects requirements for most

multicast sessions in commercial applications.

Now, let us look at the mapping between the life cycle defined in this section and the requirements that we summarized in the previous section, as shown in table 4. As we can see in this table, some requirements may have effects in more than one phase in the life cycle model for multicast sessions, and life cycle phases can also be affected by more than one requirement.

| Phase of Life Cycle | Mapped Requirements |
|---|---|
| Session announcement and session creation/termination | ***Group management*** (dynamically and automatically create/terminate a group) |
| Session topology auto-configuration | ***Group management*** (member join/leave), ***Scalability*** (large number of receivers, large coverage, inter-domain capability, collaboration with different distribution techniques, supports for multiple groups) |
| Data forwarding | ***Data delivery, Scalability*** (collaboration with different distribution techniques ) |
| AAA and Key management | ***Security, Group management*** (member join/leave) |
| Flow control | ***Reliability*** |
| Almost every phase | ***Deployment*** |

**Table 4 Mapping between Life Cycle and Requirements**

For a general solution of multicast technology for commercial usage, the requirements of group management for dynamically and automatically creating/terminating a group should be met in session announcement and session creation/termination of multicast session life cycle model. The ISP should be responsible for creating sessions for content providers, announcing multicast sessions on Internet routers, and terminating sessions when they finish. Another group management requirement, dealing with

membership changes, should be met in the phases of session topology auto-configuration, AAA checking, and Key management. In the session topology auto-configuration phase, service nodes will try to find proper parent nodes for new receivers and delete states for leaving nodes. For secure multicast sessions, AAA checking and key management will give proper access for new receivers and reject unauthorized receivers.

In session topology auto-configuration, most scalability (large number of receivers, large coverage, different AS, collaboration with different distribution techniques, and support for multiple groups) requirements should be satisfied. These requirements are related to the topology used by a multicast session and should be considered in the session topology auto-configuration phase. Another phase that will be affected by scalability requirements is the data forwarding phase. In this phase, data packets should be translated between different protocols to support different distribution techniques.

The data forwarding phase will deal with data delivery requirements. The AAA and Key management will be responsible for security requirements. The target requirement of flow control is the reliability.

The deployment requirements are actually located in almost every phase of the life cycle model. The requirement of working with different underlying hardware and software may need support in data forwarding and session topology auto-configuration phases. The requirement of easily deployment, customizability and flexibility requirements are about the design quality of multicast technology and should be considered in all phases.

As we can see in this section, we need gluing techniques to integrate the requirements and life cycle phases into a general solution, which we will introduce in the next chapter.

# 4 Overview of Project

In the last chapter, we summarized the requirements for multicast technology and defined a life cycle model for multicast sessions. In this chapter, we will propose a general solution based on the above discussion, with emphasis on satisfying commercial requirements, and introduce our project.

## 4.1 Objectives

The main objective of our research group is to create a general solution for multicast technology that will meet most of the requirements for multicast technology that we summarized in the previous chapter.

Although multicast technology has been researched and developed in the research community and telecommunication companies for over ten years, most ISPs are still reluctant to provide multicast to their customers or allow multicast to be used in their administrative domain. The difficulty of multicast deployment in commercial networks is due to not only the immaturity of multicast technology but also some drawbacks in current multicast models, which make revenue generation and control over the

multicast groups infeasible. In other words, many requirements for multicast technology in commercial usage we introduced in the last chapter cannot be satisfied by current multicast techniques.

Although current multicast techniques, both IP multicast and overlay multicast, have already gained many significant achievements in data delivery, reliability, and group management, various problems still prevent them from real commercial success. A long-term and more general solution is being expected by the ISPs.

To create a real commercially feasible multicast solution, we need not only to provide better functionalities to improve data distribution, scalability, security, group management, and reliability of multicast technology, but also enhance its relationship with operational and administrative environments in its deployment. ISPs can accept multicast technology as a commercial feasible solution only if they can profit from the multicast technology that meets most of the requirements and becomes truly operational in commercial applications.

Therefore, our research group proposed a general solution that is based on a framework concept. This framework concept consists of the following parts:

- Security
- Access control (AAA)
- Supporting infrastructure for these two facilities
- Manageability
- Flexibility (to current techniques and future development)

The security part deals with encryption key generation and management, encryption key distribution, security policy management, etc. It is the foundation for access control and revenue generation for ISPs and content providers.

Access control provides the complete set of functions for Authorization, Authentication, and Accounting, which are essential for revenue generation where using multicast technology. It allows ISPs and content providers to identify users and charge them for their information consumption. The access control is an important part of group management requirements for multicast technology in commercial usage to deal with member join/leave.

The supporting infrastructure is a framework that can be deployed all over the Internet and provide necessary services for other facilities, e.g. data forwarding, reliability, scalability, and membership management. It should be able to connect as many existing multicast technologies together as possible. It is the foundation of deploying our solution on the Internet.

The manageability allows the ISPs to easily configure the framework's functions and modules through a unified control plane. The flexibility is the framework's features of easily accepting new technology and new functional modules, and removing out-of-date functions and modules. These two characteristics can effectively improve the capability of the framework to satisfy deployment requirements.

## 4.2 Division of Solutions

Our solution for commercial multicast technology can be divided into three groups:

1. Session management and supporting hierarchy for other facilities.
2. Security mechanism.
3. Authentication, Authorization, and Accounting (AAA) system.

The security mechanism is done by Mr. Ritesh Mukherjee, who was a Ph.D. student in our research group and has finished his part. His work is to create a hierarchical encryption key distribution and management mechanism for multicast [43]. The AAA system is being developed by Mr. Salekul Islam, a Ph.D. candidate in our research group [42] [44]. His work is to build a framework for the use of AAA protocols to manage IP Multicast group membership. The Security mechanism and AAA system are dealing with the security requirements and cover the AAA checking and Key management phases in the life cycle model of a multicast session.

The focus of my project will on the provision of session management and supporting hierarchy for other facilities. The three topics, security, access control, and supporting infrastructure, are developed as separate and independent projects, but the three projects are logically connected and go towards the same main objectives: the commercial version of multicast technology.

Due to limits of time and research resources, we focused our research on a smaller set of requirements:

- Data delivery
- Scalability
- Group management (dynamically and automatically create/terminate a group, dealing with membership changes)
- Deployment (ease of deployment, working with different underlying hardware and software)

For the other requirements, including security, reliability, and naming a group (group management), there are some other people's works in our research group or some existing technology in these fields, e.g. MALLOC and FEC schemes. This project should consider and reserve places for them in our solution, but does not bring up with any new ideas or techniques about them. Other deployment requirements, including customizability and flexibility, are embodied almost everywhere in our design, so they will be considered but the details will be not discussed in this project.

My project will cover the life cycle model phases of session creation, session announcement, session topology auto-configuration, data forwarding, flow control, and session termination.

| Requirements | | Tianyu Wang | Ritesh Mukherjee | Salekul Islam |
|---|---|---|---|---|
| Data delivery | Efficiency | Yes | | |
| | Robustness | Yes | | |
| Scalability | Large number of receivers | Yes | | |
| | Large geographical coverage | Yes | | |
| | Inter-domain capability | Yes | | |
| | Collaboration with heterogeneous distribution technologies | Yes | | |
| | Support for multiple groups | Yes | | |
| Security | | | Yes | |
| Group management | Ability to name groups | | | |
| | Dynamically and automatically create/terminate a group | Yes | | |
| | Dealing with membership changes | Yes | | Yes |
| Reliability | 100% reliability without time bound | | | |
| | Reliability suitable for live stream applications | | | |
| Deployment | Working with different underlying hardware and software | Yes | | |
| | Ease of deployment | Yes | | |
| | Customizability | | | |
| | Flexibility | | | |

**Table 5 Requirements Covered by Projects in our Research Group**

In table 5, we list all requirements we tried to cover in the projects within our research group, as we discussed in this section.

## 4.3 Project Introduction

This project on session management is based on the requirements of

multicast technology for commercial usage and the life cycle model for multicast sessions, which we introduced in the last chapter. Basically, the project is divided into three parts:

- Hierarchical topology auto-configuration
- Session management mechanism
- Support for different multicast protocols

The hierarchical topology auto-configuration is the foundation of the project, and will be used in the topology session auto-configuration phase of the multicast session life cycle model. Generally, there are remarkable needs for hierarchical topology, and a sophisticated algorithm managing the hierarchical topology, in almost every aspect in multicasting.

As we can see in the above discussion of current multicast technology, in the “one sender and multiple receiver” model, when there are many receivers, and reliable transmission is required in some scenarios, a hierarchical topology must be used to avoid overwhelming the sender. In multicasting, because the hosts may be located over a large area (sparse mode) and may join or leave the multicast groups dynamically, multicast protocols need sophisticated hierarchical topologies to organize numerous

hosts into groups and flexible mechanisms to manage the multicast groups. When reliable transmission is required, because a single node cannot handle enormous error reports from receivers and deal with retransmission, the hierarchy becomes more important to aggregate error reports and to provide local error recovery.

The tree structure is widely used in multicasting. In the network layer, the data distribution needs the tree structure to forward the data flow from the sender to the receivers. A host needs to find a proper parent node and bind to the data distribution tree, and to know its child nodes for data forwarding. In the transport layer, a host also needs to find its parent node for error recovery. The reliability of multicasting depends on sending error reports to upstream nodes and retransmission of missed data packets. In the session layer, the session establishment relies on finding the appropriate data distribution and error recovery trees.

In overlay multicast, some different hierarchical topologies are used to connected nodes into a manageable group. Wireless and Peer-To-Peer, (P2P in ESM and PeerCast) environments do not have infrastructures that consist of fixed intermediate nodes. The advantage is that they do not need router or

other intermediate node support. The disadvantage is that the nodes may experience high service interruption because nodes serve as both receivers and routers, and ancestor nodes may join and leave dynamically.

In the routing function of multicast protocols, the flood and prune method has been proven to be acceptable only for dense mode in multicast because it will generate an overwhelming number of control messages on the Internet when the number of receivers is extremely large. Therefore, the research community developed “sparse” mode multicasting, in which a receiver should send an explicit join request to its parent node. The method for finding an appropriate parent node becomes critical. At the routing level, unicast routing tables exist at all hosts, and are easy to see the path back towards the root of the multicast tree. In some environments, the “reverse path” towards the root of the multicast distribution tree is not the same as the unicast path towards that same root node. In this case, a hierarchical routing mechanism may be used and intermediate multicast routers or service nodes must maintain information about their parent node in the tree. A receiver can send a join request “towards” the root of the multicast tree, using the unicast routing as the reverse path routing, as appropriate. The parent node should respond to the receiver’s join request.

In error recovery, reliable multicast needs to find merge points up the tree to aggregate error reports and to provide retransmission. Error recovery is a transport-level function, not a routing function. It is not the responsibility of the routers. Currently, the multicast error recovery tree in the transport layer is often different from the multicast data distribution tree in the network layer. The congruence of error recovery tree and data distribution tree may bring benefits for management of data distribution and membership. Co-locating error recovery with the router and asking its help may be a drain on the router. However, having at least a “point of capture” has been shown to be very important. The IETF working group on reliable multicast protocols tried to build a generic router assist (GRA) mechanism that is a general mechanism located at routers. It enables end-to-end multicast transport protocols to take advantage of information distributed across the network elements in a given multicast distribution tree. The GRA has been gradually discarded by the IETF, because of its complexity. The most desired function in transport layer is to locate neighbors on the hierarchy in the right direction.

For session establishment, hosts need to locate or build a tree for the group to be joined and establish the right to be a member of this group. In the

session layer, a session management mechanism should establish a group and allow receivers to join the group. The Security and AAA (Authentication, Authorization, and Accounting) mechanisms will safeguard the group, authenticate the identity of permitted receivers, and bill receivers according to their uses of the services, etc.

Basically, distribution trees are used for forwarding origin data and retransmitted data from sender to receivers in all multicast technologies, and reverse trees are used for aggregating control information and error reports to upstream nodes in reliable multicast technologies. In current multicast models, the distribution tree and reverse management tree may not be the same trees, and service nodes on different trees are not congruent. This will introduce a lot of management problems.

We need a hierarchical topology auto-configuration mechanism that allows nodes to be connected as a well-organized body. It should meet the requirements of group management (dealing with the membership changes), scalability (large number of receivers, large geographic coverage, and inter-domain capability, and support for multiple groups). It should also provide support for requirements of data delivery, security, reliability, and

deployment. We defined this auto-configuration based on a mesh topology, which is a set of pre-deployed service nodes. The mesh topology can map the reverse management trees onto the distribution trees, allow multiple tree instances, and support much functionality we discussed, e.g., security and AAA.

In the previous chapter, we have introduced the life cycle model for multicast sessions and how the requirements for multicast technology in commercial usage can be mapped to the life cycle. Derived from the life cycle model, we can build a multicast session management mechanism.

The session management mechanism is the kernel of our design and presents a general solution for how all nodes can collaborate with each other. It provides a framework on which modules of the multicast session life cycle model can be glued together. Therefore, many requirements for multicast technology in commercial usage can be satisfied by support of the session management mechanism.

In the next chapter, we will construct the architectural model of the session management, including the specification of all modules and interfaces

between them. The detailed design of each module will also be given, with specifications of internal design for all modules, including state machine, primary data structure, etc.

Until now, we have introduced a lot of multicast protocols developed by the research community and telecommunication companies. There will be even more multicast protocols proposed and developed in the near future to meet different application requirements. However, the more heterogeneous multicast protocols are proposed and developed, the more complicated situations will rise in deployment and commercial application of multicast technology. Generally, when groups using different protocols are interested in the same contents from a single source, these protocols will establish their own hierarchies, may need different content streams of the source, and may have no sharing among them.

Therefore, our solution should be able to connect different protocols together and provide a stable and shareable infrastructure for multicast sessions. This problem indicates a new challenge for multicast protocols: how to collaborate with different multicast protocols? This is a specific topic in scalability requirements of multicast technology.

As we discussed in chapter 2 and 3, native IP multicast has limited coverage and good overall performance, and overlay multicast has unlimited coverage but poor performance in many aspects, e.g., data forwarding and group management. This scenario leads us to a solution that glues native IP multicast and overlay multicast to get unlimited coverage and acceptable overall performance. This solution will give multicast technology much greater deployment ability.

Now, we need to highlight the focus of this project. As we have introduced above, we do not deal with AAA mechanism and security in this paper. We will not propose new techniques in some other parts in the session life cycle, e.g., flow control and session announcement mechanism, because some research groups, e.g., IETF working groups, have already achieved remarkable progress in those fields. Our project will try to combine these techniques with our solution.

Although security and AAA are critical parts in commercial usage, they are not required for some “open” groups, which may allow anyone to access the group. Therefore, security and AAA are optional techniques in multicast

technology. The focus of this project is the important techniques that can improve scalability and many other factors for all kinds of multicast technology.

Our solution has many advantages over most current multicast technologies. First, it is designed based on requirement analysis of multicast technology in commercial usage and provides supporting services for important functionalities for commercial usages, e.g. security and AAA. It makes our solution an excellent infrastructure for multicast technology in commercial usage. It can collaborate with different multicast technologies, and suit underlying software and hardware. It gives our solution unlimited coverage and adaptability.

The life cycle phases and requirements covered in the project are shown in table 6.

| Life Cycle Phase | Requirements |
|---|---|
| Session announcement and session creation/termination | Group management (dynamically and automatically create/terminate a group) |
| Session topology auto-configuration | Group management (member join/leave) Scalability (large number of receivers, large coverage, inter-domain capability, collaboration with different distribution techniques, support for multiple groups) |
| Data forwarding | Data delivery, Scalability (collaboration with different distribution techniques ) |
| Flow control | Reliability |
| Almost every phase | Deployment (work with different underlying hardware and software, and ease of deployment, i.e., incremental deployment) |

**Table 6 Life Cycle Phases and Requirements Covered in this Project**

Before our detailed discussion in the next chapter, it is necessary to summarize the research status of the three parts in this project. For hierarchical topology auto-configuration, it is my previous work for my master's degree. The reason I put it in this project is that it is the foundation for the other two parts, and readers cannot completely understand the other two parts without it. For multicast session management, some research groups may think about its importance and propose some simple and basic theoretical models. In this project, the life cycle model in the last chapter and the detailed design described in the next chapter are original and innovative.

For the support for different multicast protocols, although the Scalable Adaptive Multicast (SAM) Research Group, IRTF (Internet Research Task Force), has realized the importance of the topic and started working on it, their work is still at the beginning and very limited right now. Our proposed solution for this topic in the next chapter is much more complete and totally innovative.

The goal of this project is not to replace the existing multicast protocols designed in network layer and transport layer, but to establish a framework that coordinates the different multicast protocols in multicast sessions and to provide a commercially feasible solution for multicast technology. It should be deployed upon other multicast technologies on Internet to provide comprehensive services for ISP, content provider, and end users.

# 5 Design

In this chapter, we will introduce the architectural and detailed design of our proposed solution. Section 5.1 will introduce the hierarchical topology auto-configuration mechanism. Section 5.2 will introduce the session management mechanism. Section 5.3 will introduce the techniques supporting heterogeneous multicast protocols.

## 5.1 Hierarchical Topology

In my previous work for the master's degree, I designed a new algorithm to establish a hierarchical topology, which is the foundation for the other two topics in this project. It may not the most optimal solution for this problem, but provides a useful and manageable one. Essentially, the algorithm is based on three techniques: Mesh, Local Group, and our new Controlled Expanding Ring Search (CERS) algorithm. The hierarchical topology generated by this algorithm comprises a small number of Senders, a pre-deployed Mesh, and a large number of local groups that have a local Service Node and some Receivers, as shown in Figure 7.

This algorithm should be used in the session topology auto-configuration phase of the multicast session life cycle model, and it will satisfy

requirements of group management and scalability (large number of receivers, large coverage, inter-domain capability, collaboration with different distribution techniques, and support for multiple groups).

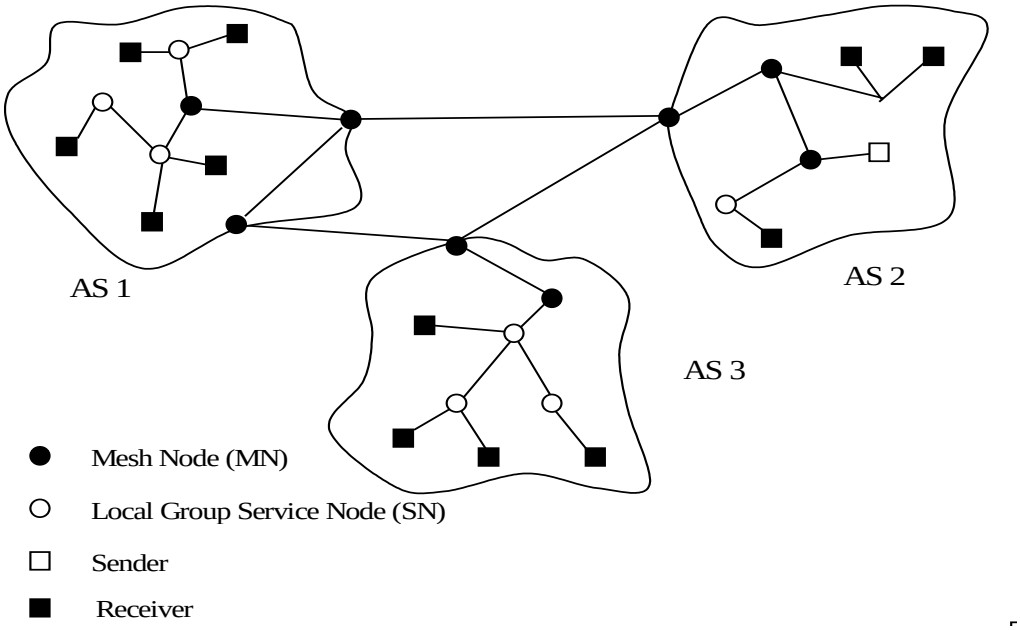



**Figure 7 A Session Tree Generated by Hierarchical Topology Auto-Configuration**

## 5.1.1 Overview of Hierarchical Topology

The hierarchical topology generated by our proposed algorithm consists of several parts. The core of this topology is a Mesh that is the infrastructure of the generated topology. A sender node is directly connected to the Mesh. The receivers are organized into multiple Local Groups. Each local group has a Service Node (SN) working as local group controller, which is responsible for managing the local group and contacting other service nodes. Logically, a multicast session group is organized into a tree: the root is the

Mesh node (MN) that is directly connected to Sender, other MNs and local SNs are intermediate nodes, and receivers are leaf nodes.

The Mesh approach was originally introduced in the IETF draft “Reliable Multicast Transport Building Block: Tree Auto-Configuration” [39]. The Mesh is a set of pre-deployed service nodes, which form the infrastructure of the hierarchical topology. At the beginning, MNs are not necessarily aware of any multicast session. Each MN knows a subset of MNs as its immediate neighbors. Each MN has a Forwarding Table that contains the information of next-hop to reach a destination MN. Each MN can “broadcast” information to all other MNs [39]. In our design, we assume that the service provider configures the core network nodes as mesh nodes, and the routing among them is the job of the existing routing protocols.

The mesh approach has many advantages over shared trees and source-based trees. First, all MNs can be chosen as a root node for a multicast session, so there is no need to build a backup root node. With the mesh, another benefit is that multiple groups can be better supported by assigning different root MN nodes for each group. Second, the mesh approach also provides a long-term solution for inter-domain multicast communication. If we configure

border routers in different domains as MNs, the mesh can establish the connection between autonomous systems (ASes) and use BGP to compute the inter-domain forwarding routes. The border MNs only need to trust MN nodes in another AS, instead of trusting all nodes in the other AS. Finally, because the connections between these MNs can be built before other nodes join the tree, it can solve the problems of join latency and bursty source in dynamic groups. In general, the mesh approach will improve the scalability of multicast technology.

However, the mesh approach proposed in the IETF draft has a problem because it needs a direct binding between mesh service node and receivers. When there are millions of receivers in a multicast group, a large amount of resources will be used to maintain those bindings. Moreover, the direct connection will limit scalability of multicast groups, because only one level of hierarchy can be extended outside the Mesh and some receivers in a LAN far away from the mesh cannot share a common connection.

Therefore, we adapted the Local Group concept to our hierarchical topology. The Local Group concept was originally proposed in LGMP [13], which we have introduced in section 2.3.2.1. We use a similar concept to LGMP's

local groups, but we use different mechanisms to organize local groups and bind local groups to our hierarchical topology. All hosts in a Local Area Network (LAN) form a local group. The local group is a two-layered topology. Each local group has a group controller, called a Local Service Node (SN). All receivers in this local group directly connect to the SN. The reasons to limit only hosts on a LAN into a local group are that it can simplify local group auto-configuration, and can allow receivers on a LAN to share a common connection to external hosts.

The SN is the manager of the local group. To obtain the optimal multicast capability, we can statically configure the local router with an interface to the Internet as the local SN. Another way is to let receivers elect one of themselves as the SN, in case the local router is not multicast-enabled. We designed and implemented an SN Election procedure to do this work [40], which is similar to the PIM-SM Designated Router (DR) election. The election procedure is also useful to re-configure the group when the SN fails. The details of the SN Election Procedure are out of scope for this paper. Like MN, the SN is initially not aware of any multicast session and does not bind itself to any other SN or MN. The SN has a child list that stores information about all children.

Not only functioning as a local group manager, an SN can also be an intermediate service node on the path from a local group to the mesh. It can accept another SN's join request to a specific session and add accepted SN into its child table.

Local group concept is one foundation of our algorithm. The division of hosts into local groups can significantly reduce control traffic on the Internet and improve scalability and manageability of multicast protocols. Serving as intermediate service nodes, SN nodes can reduce the joining cost of local groups and help hierarchical topology with its extension. An SN can be the core of data forwarding from the upper layer to its children, error report aggregation, local error recovery, etc.

A multicast session tree generated by my algorithm will be a hierarchical topology that has multiple layers. The root is the nearest mesh node to the sender. Several upper layer nodes will be mesh nodes, which may cross many autonomous systems (AS). The intermediate layer nodes are local SN nodes, and some of them can be the relay nodes for other SN nodes on the path towards the mesh nodes. The leaf nodes are receivers. Theoretically, a

multicast session tree based on this algorithm can have any number of tree levels and support any number of receivers.

As we can see in the above discussion, the hierarchical topology auto-configuration satisfies the requirements designated for it. The mesh and local group methods can significantly improve the scalability of multicast technology, since it can have a large number of receivers, large geographic coverage, inter-domain capability, and support for multiple groups. The processes introduced in the next section will provide a group management plan.

In our simulation experiment, which will be introduced in chapter 6, some scalability requirements, including supporting large number of receivers, large coverage, and multiple groups, will not be included, due to limits of research resources. However, because of support of local groups and mesh, any number of receivers and groups can be supported by our algorithm. The inter-domain capability of our algorithm, which will be proved in the simulation, can efficiently solve the most important problem in covering large geographical area. Therefore, these requirements should be effectively met by our algorithm.

## 5.1.2 Node Joining Process

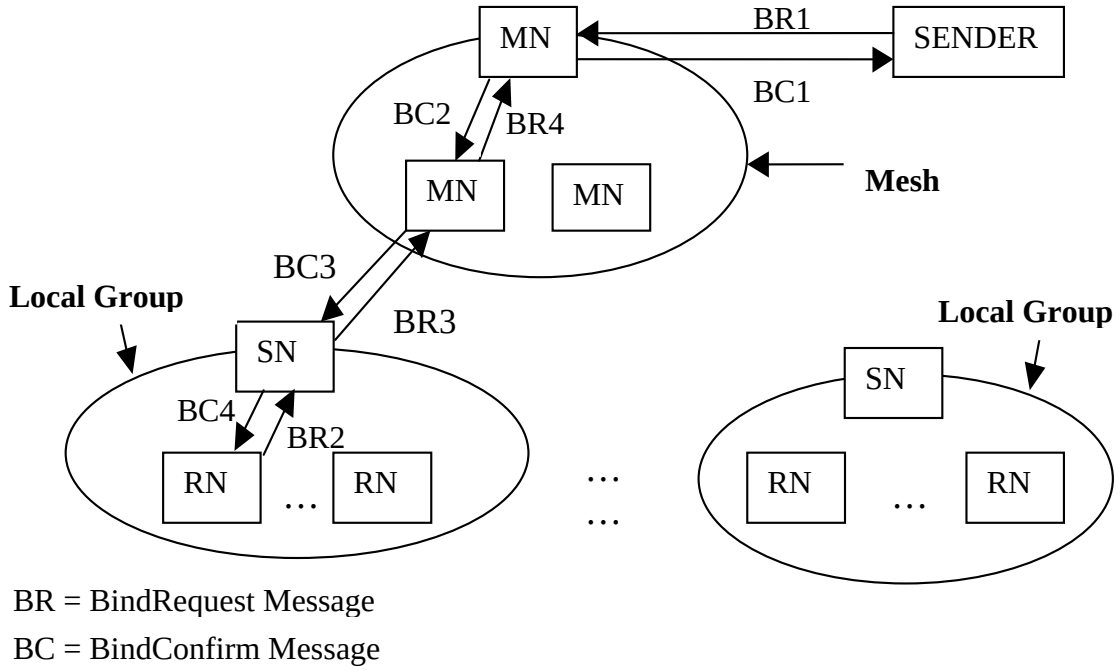



**Figure 8 Member Join Procedure in Hierarchical Topology Auto-Configuration**

In my previous project for my Master's degree, I designed and implemented a new multicast group auto-configuration algorithm [40], as shown in the above figure, which has five steps:

1. Mesh construction. (It is introduced in section 5.1.1.)
2. Sender locates a neighbor MN, sends a Session Announcement, and binds to the closest MN in the mesh. MN 'broadcasts' this session announcement on the Mesh.
3. Receivers locate a local service node (SN) and send a BindRequest message to this local SN.

4. Local SNs join the session tree and accept receivers' bind request.
5. Session tree on the mesh is built.

The second step is to let sender start the multicast session. When a Sender wants to start a new multicast session, it tries to find the closest MN by contacting an MN assigned by the network operator or using dynamic methods to choose an MN from several candidates. The sender sends a BindRequest (BR1) message the closet MN. The MN accepts the sender by replying with a BindConfirm message (BC1) and becomes the root of the session tree. After being accepted, the Sender sends a Session Announcement to the root. This Session Announcement should contain session ID, multicast address, port number, and other session information. The root will "broadcast" this session announcement to all MNs.

In the third step, the receivers should locate the local SN and try to join the local group. A receiver broadcasts a BindRequest message (BR2) on its LAN. If there is no SN on this LAN, the SN Election procedure can establish a local SN for this LAN before the local group joins any multicast group. If there is an SN on this LAN, it responds to the BindRequest message. If the SN is already on the session tree, it sends back a

BindConfirm message (BR4), builds an entry in its child list for this new receiver, and ends the process. If SN is not on the hierarchical topology yet, it sends a BindACK to the receiver. This message causes the receiver to wait while the SN processes its request. The SN needs to join the session tree before accepting any children.

The most important and difficult function of a local SN is neighboring node discovery and selection. In the fourth step, if an SN is already connected to the hierarchical topology, it will send a JoinRequest for the session to its parent nodes. Otherwise, the SN needs to automatically choose a parent node, which should be the nearest MN or SN that is already on the session tree and can accept another child. We designed a new algorithm, called Controlled Expanding Ring Search, to fulfill this task. We will discuss this new algorithm in the next section. After the parent node is chosen, the SN node sends a BindRequest message (BR3) to its parent, and then waits for the reply from its parent.

In the fifth step, if the MN that receives BindRequest messages is not a node on the session tree yet, it uses the next-hop information of the forwarding table entry for the root MN to build the shortest path to the root. The MN

sends a BindRequest message (BR4) to the next-hop and waits for a response. This process ends when a MN on the session tree or the root accepts a bind request. After the session tree on the mesh is built, all MN nodes should respond to the BindRequests that they received, using a BindConfirm message (like BC2).

Each SN sends a BindConfirm message (BC4) to its children after receiving a BindConfirm (BC3) message from its parent, and binding itself to the parent. If it receives a BindReject message, it tries to find another parent and binds to it. When the receivers get their BindConfirm messages, the algorithm ends.

Another question that we should consider is the formation of the session tree. There should not be any loop in the generated session tree. The IETF draft “Reliable Multicast Transport Building Block: Tree Auto-Configuration” [39] introduced a feasible and efficient algorithm that solves this problem.

The hierarchical topology auto-configuration also has a node leaving process, which is another important function for dealing with membership changes. A receiver node or SN explicitly sends a LeaveRequest about a session to its

parent node, and then waits for LeaveConfirm message from its parents. An SN can leave a session only when all its children have left the session tree.

### 5.1.3 Controlled Expanding Ring Search (CERS) Algorithm

The CERS algorithm works as follows: A local SN tries to trace the IP route to the root MN by a function like the traceroute program that can find all the intermediate routers on the path, and then the SN sends the Query messages to all the routers on the path, with a specific TTL (time to live) value, and waits for the reply for a specific interval, SolicitPeriod. If there are one or more replies from those routers within a SolicitPeriod, the SN calculates the round trip time (RTT) of the message between the routers and itself, and then chooses the closest node as its parent. If there is no reply at all, the SN will increase the TTL and query for a parent again. This process ends when at least one reply has been received or the TTL becomes greater than a maximum TTL, TTLMax. If TTL is greater than the TTLMax, the binding has failed and the local SN will inform all receivers about the result.

This algorithm has some advantages over the expanding ring search (ERS) algorithm. In Expanding Ring Search (ERS) proposed by IETF draft [8], the new node sends Query messages in the multicast channel. ERS floods the

query message all over the whole multicast group. Expanding ring search (ERS) is an effective technique in a local subnet or intranet (especially when the IP multicast routing protocol is dense-mode based). However, ERS is not practical or efficient in a multi-domain network or for the sparse-mode-based routing protocols, because it can add significant control traffic overhead. The CERS algorithm queries only the nodes that are on the path to the root. Other nodes will not be involved in this process. This feature can significantly avoid unnecessary control messages.

CERS can also avoid some inefficient tree branches as created by the ERS algorithm, as shown in Figure 9. In this topology, we assume that there is a multicast tree rooted at N1 and there is already a service node on the tree, N2. When a new service node, N3, wants to join a tree, it uses ERS and multicasts a Query message with an initial TTL. If the TTL is long enough to allow the Query message reach N2 but not reach other nodes, the new SN may consider N2 as the best parent candidate and bind to it. Clearly, N2 is not the best choice, and it even needs to get multicast data via N3. The controlled ERS algorithm can avoid such inefficient connections by only querying routers on the shortest path to the root.

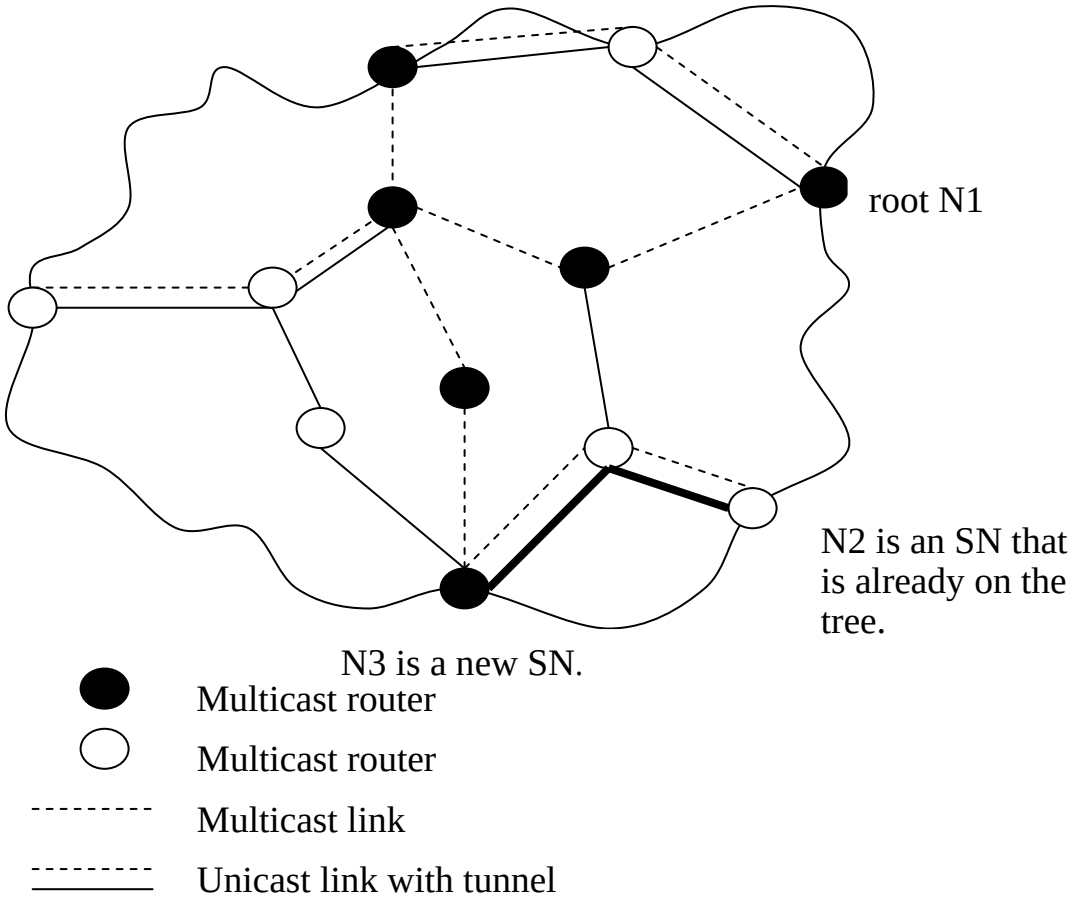



**Figure 9 Inefficient Tree Created by ERS**

## 5.2 Session Management Mechanism

After building a hierarchical topology, we need to build a mechanism that can control every phase of the multicast session life cycle. This session management mechanism will be placed in every node of the multicast hierarchical topology and will allow nodes to collaborate with each other. The architectural model of the session management mechanism is shown in Figure 10.

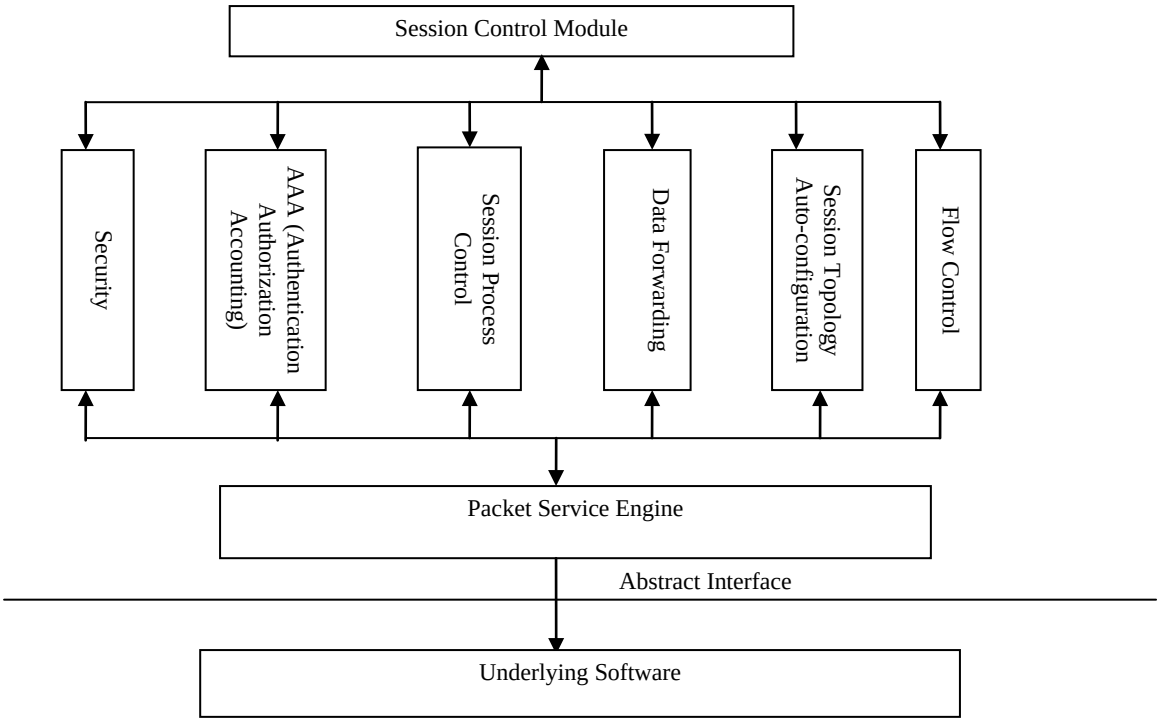



**Figure 10 Session Management Mechanism**

Basically, the session management model is divided into three layers. The Session Control Module is in the top layer. There are six modules in the intermediate layer: security, AAA (Authentication, Authorization, and Accounting), Session Process Control, Data Forwarding, Session Topology Auto-Configuration, and Flow Control. The bottom layer has a Packet Service Engine module.

The session management mechanism is located on every node of the hierarchical topology generated by our algorithm introduced in section 5.1. The session control module is the central control module that manages the behaviors of all other modules.

The intermediate layer modules provide functionalities for each phase of the multicast session life cycle model. Different types of nodes in the topology will use different functionalities of the intermediate level modules. The Packet Service Engine is responsible for transferring data between underlying software and the modules in upper layers.

For Flow Control module, because of limits of time and research resources, we cannot provide full coverage of current reliability techniques for multicast. Therefore, we will not provide detailed discussion of flow control and reliability. Alternatively, we will provide an example in our simulation in chapter 6, which will show the capability of supporting 100% reliability in our design.

As we have discussed above, the Security and AAA modules are out of this paper's scope. We will focus on the other modules and introduce their detailed design in this section.

### 5.2.1 Session Control Module

The Session Control Module is the dominant module in our design. It

maintains critical data structures, e.g., session list, mesh node forwarding tables, child list for each session, etc. It controls all other modules.

#### 5.2.1.1 Important Data Structure

##### *5.2.1.1.1 Session Table*

Because every node can support multiple sessions, i.e., multiple groups, we need a special data structure, a session table, to store the information of all sessions that the node knows. Each table entry stores the information of the session root, sender, session ID, session group address, and pointer to a child table.

1) Session Root Information: It is the IP address of the root node (mesh node) and port number using for this session by the root node.
2) Sender Information: It is the IP address of Sender node and port number using for this session by the Sender node.
3) Session ID: It is a unique integer used for identifying the session on the whole hierarchical topology. Although there may be many other ways to identify a session, e.g., the group address, we still need a way to represent a session in case there is no IP addressing service for a multicast group. This number is assigned by the root node, fed back to the sender, and sent to all other nodes.
4) Session Group Address: If the addressing service is available for the

IP multicast and the session is using IP multicast, the root will request a group address for this session and add it the session information.

5) Pointer to a Child Table: Each session has its own child table, which will include the information of the node's direct children. The child table will be discussed later.

To support multiple multicast protocols, which we will discuss later in this chapter, all nodes should know some information of how the session will operate. We should also include such information in the session table:

6) Session Ending Condition: Each session can stop when some conditions are met, or operate constantly. The Session Ending Condition should indicate those conditions or constant operation. The ending conditions can be number of packets, length of a file, ending time value, or other conditions. Each session ending condition should contain at least two fields: condition types and condition value.

7) Flow control scheme: Each session will have a unified flow control scheme, which will be supported in our solution and all other domains that are connected to ours. The flow control scheme can be best-effort, FEC enabled, etc. The choice and implementations of flow control schemes are out of scope of this paper.

There are many other aspects of a session that should be covered in this

section, e.g., AAA settings and QoS, which are out of scope and can be added as future work.

#### *5.2.1.1.2 Forwarding Table*

The forwarding tables are only created and maintained by mesh nodes, which store the next hop information used to reach other mesh nodes. This table is important for building the session tree, as described in 5.1.

#### *5.2.1.1.3 Child Table*

Child Tables are used to store information about a nodes' direct children in a session. Such information is obtained when a child node requests binding to this node. Each entry of the Child Table will contain:

1) Address Information: a child node's IP address and port number for this session.
2) Sequence Number: the current sequence number that the child is requesting.
3) Optional QoS parameters: e.g., RTT, latency, or data rate.

#### *5.2.1.1.4 Parent*

This value is maintained and used only by receivers and SN. Because a receiver or an SN will only need one connection to the hierarchical topology for all sessions it joined, the parent information will contain the IP address,

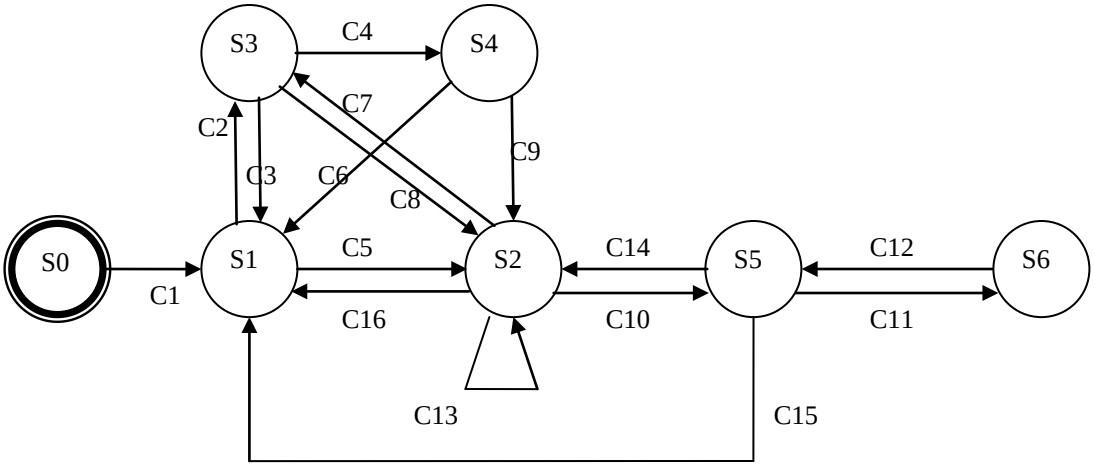
port number, Last Response Time, and some other information about its parent.

#### 5.2.1.2 State Diagram

In this section, we present state diagrams that represent state transitions of the session control modules in mesh nodes, local service nodes (SN), senders, and receivers. In the state diagrams, each state represents a set of functions in the intermediate level modules, which we will discuss later. Each transition (edge in the state diagram) is labeled with a transition condition. When the transition condition holds, the session control module will move from its current state to a new state. The state information and transition conditions are given in the state diagram. In the following discussion, because we need to give a general mechanism of session management, we have to talk about some basic functionalities of security, AAA, and flow control, which will not be covered in this paper in more detail.

In the mesh node state diagram, Figure 11, the state S0 is the start state where a mesh node starts up and gets ready for its functionality. In the state S1, a mesh node waits for an event that the sender establishes a session and informs the mesh node about the new session. If the mesh node is requested by a sender to establish a new multicast session, the mesh node will become

the root for this session. The root of this session will check the sender's rights of establishing a new session, in state S3. If the session is granted successfully, an encryption key will be generated and sent to sender in state S4, and then the root will go back to state S1. The root also needs to inform all other mesh node about the existence of the session, in state S1. If the root rejects the sender in state S3, the root will inform the sender about the rejection and go back to state S1.




S0: Start state
S1: Session Creation and Session Announcement (Session Process Control)
S2: Session Topology Auto-configuration
S3: AAA Checking (AAA)
S4: Key Generation (Security)
S5: Data Forwarding
S6: Flow Control
S7: Session termination (Session Process Control)

C1: Node starts
C2: Sender join
C3: Sender is rejected
C4: AAA Checking succeeded
C5: Session starts successfully and a new child joins
C6: Generate key for sender
C7: A new child joins, or periodically re-checks membership
C8: New child is rejected
C9: Generate key for children
C10: New Data packet is received
C11: Require flow control
C12: Flow control finished
C13: Parent fails, children leave, or node periodically optimizes the topology
C14: Received packet is processed
C15: Session is over
C16: A new sender wants to join

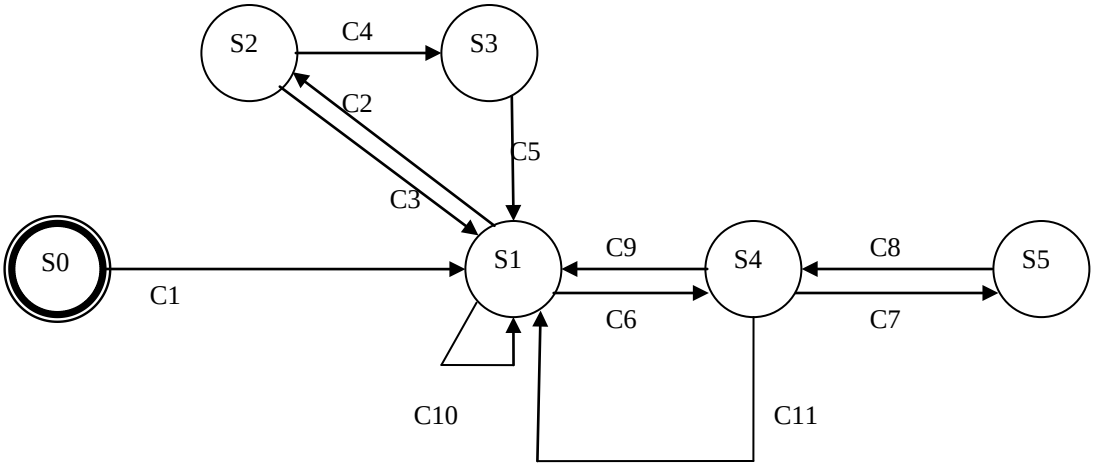
**Figure 11 Mesh Node State Diagram**

When the session information is created in the session table and a new child issued a join request to a mesh node, the mesh node will enter the next step, establishing the hierarchical topology, in state 2. At any time, when a new child wants to join the session, the node receiving the join request will check the new child's rights in the session, which is done in state 3. If the child is

acceptable, a new encryption key is generated and sent to the new member in state 4. Otherwise the mesh node will reject the new child node.

A mesh node will periodically probe the existence of its neighbors on the topology in state 2, and also periodically recheck its children's rights for sessions in state 3. The hierarchical topology auto-configuration module will be called when child nodes leave, a node's upstream node fails, or periodically topology optimization is necessary, in state 2. The key generation will also update a valid child's encryption key periodically in state 4. If a child node loses its rights in this session or its account balance is insufficient, it will not receive an updated key and not be able to receive data any more.

The data forwarding module will be called when new packets are received on a node in state 5. In state 6, the flow control module will be automatically called when necessary, e.g., requesting missed packets or QoS requirements have not been met. In state 5, the session is terminated when a sender reaches the end of the data stream and informs the root about it. When a session is over, the mesh node will return to state 1 and wait for the creation of another session. The process of session termination will be discussed later.

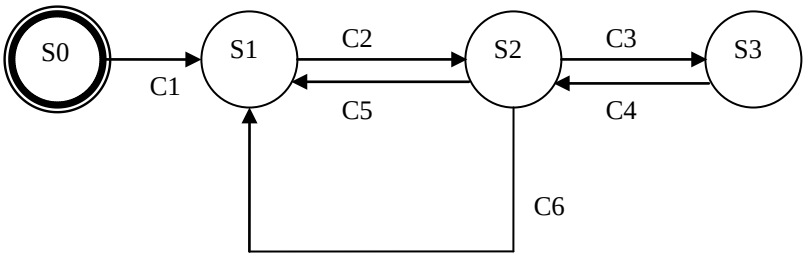



S0: Start state
S1: Session Topology Auto-configuration
S2: AAA Checking (AAA)
S3: Key Generation (Security)
S4: Data Forwarding
S5: Flow Control

C1: Node starts
C2: A new child joins, or periodically re-checks membership
C3: New child is rejected
C4: AAA Checking succeeded
C5: Generate key for children
C6: New Data packet is received
C7: Require flow control
C8: Flow control finished
C9: Received packet is processed
C10: Parent fails, children leave, or node periodically optimizes the topology
C11: Session is over

**Figure 12 Local Service Node State Diagram**

The local Service node (SN) has a similar state diagram as the mesh nodes, as shown in Figure 12. An SN will wait for join requests from other nodes (state 1), maintain session topology (state 1), check children's rights (state 2), generate keys for valid children (state 3), forward data to its children (state 4), and recover a lost data packet (state 5). When all sessions are over, the SN will return to state 1 and wait for another session.

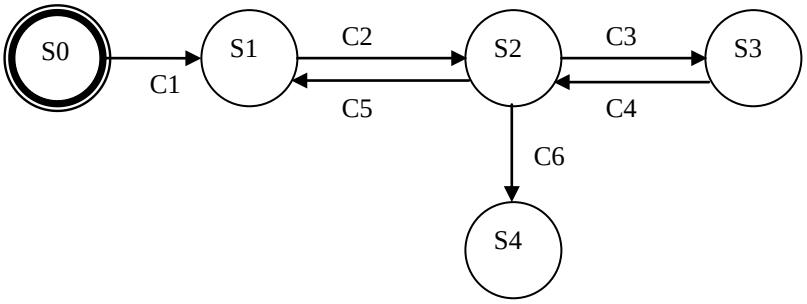



S0: Start state
S1: Session Creation (Session Process Control)
S2: Data Forwarding
S3: Flow Control

C1: Node starts
C2: New Data packet is generated
C3: Require flow control
C4: Flow control finished
C5: Data packet has been processed
C6: Session is over

**Figure 13 Sender Node State Diagram**

The state diagram for sender node is relatively simple, as shown in Figure 13. The sender node forwards data packet to the root and retransmits missing data packets for a session if required. When the session is over, the sender node will return to state 1 and be ready for next session it may create later.




S0: Start state
S1: Session Topology Auto-Configuration
S2: Data Forwarding
S3: Flow Control
S4: Stop

C1: Node starts
C2: New Data packet is generated
C3: Require flow control
C4: Flow control finished
C5: Data packet has been processed
C6: Session is over, or the session ends accidentally

**Figure 14 Receiver State Diagram**

In Figure 14, the receiver will request to join a session (in state 1) when it starts up. As we described above, if there is no local group controller on the

LAN, one of the receivers will be elected as a new local group controller, and it will change its role and functions to a local group controller. After binding successfully to a local SN, it will receive data packets (in state 2) and request lost data packets (in state 3). When a session is over, the receiver process will terminate itself.

### 5.2.2 Session Process Control Module

The Session Process Control module is responsible for session creation, session termination, session announcement (on Mesh), and other aspects of session maintenance. Correspondingly, it will cover the session creation, session termination, and session announcement phases in the multicast session life cycle model. It will provide functionalities to meet group management requirements of multicast session.

The session creation process is shown in Figure 15. In session creation, the sender of the session will create a sender JoinRequest message with its own address and other session information, and then will locate a root (mesh node) for this session and send a request to the possible root. The root location can be pre-assigned by the ISP or by some other ways. If the session is accepted by the root and a JoinConfirm message is received from the root,

the sender will create an entry for this session in its own session table and start to ask the Data Forwarding module to send data to the root node.

A mesh node is chosen as the root for the session. It will request a group address for this session, if the addressing service for multicast is available, and assign a unique session ID for the session. The root should also take care of setting up important parameters for this session, e.g., flow control scheme, QoS parameters, security and AAA options.

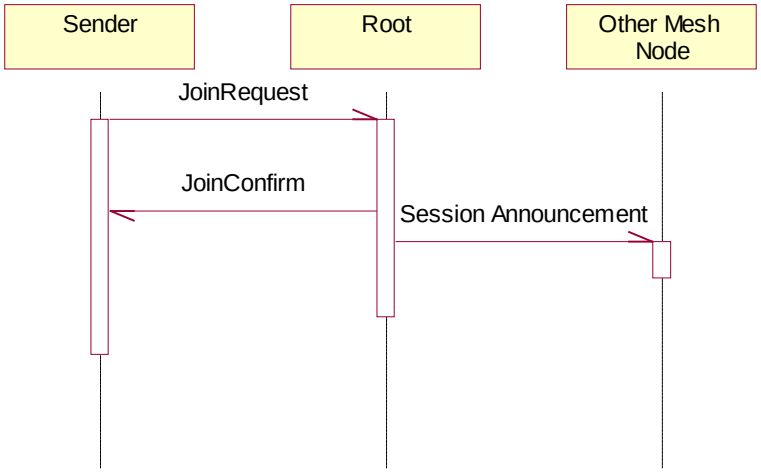



**Figure 15 Session Creation**

After the session is created, the root should inform all other mesh nodes about the existence of the new session, which is the task for session announcement. All important session information, including root information, group address, session length, data rate, flow control scheme option, etc., will be inserted into a session announcement message created

by the root. The session announcement will be distributed to all mesh nodes by the session announcement mechanism, which has a ‘broadcast’ system among mesh nodes and will update the mesh node session information periodically. The mesh nodes will periodically exchange session information with their neighbor to ensure that each mesh node has an up-to-date session table.

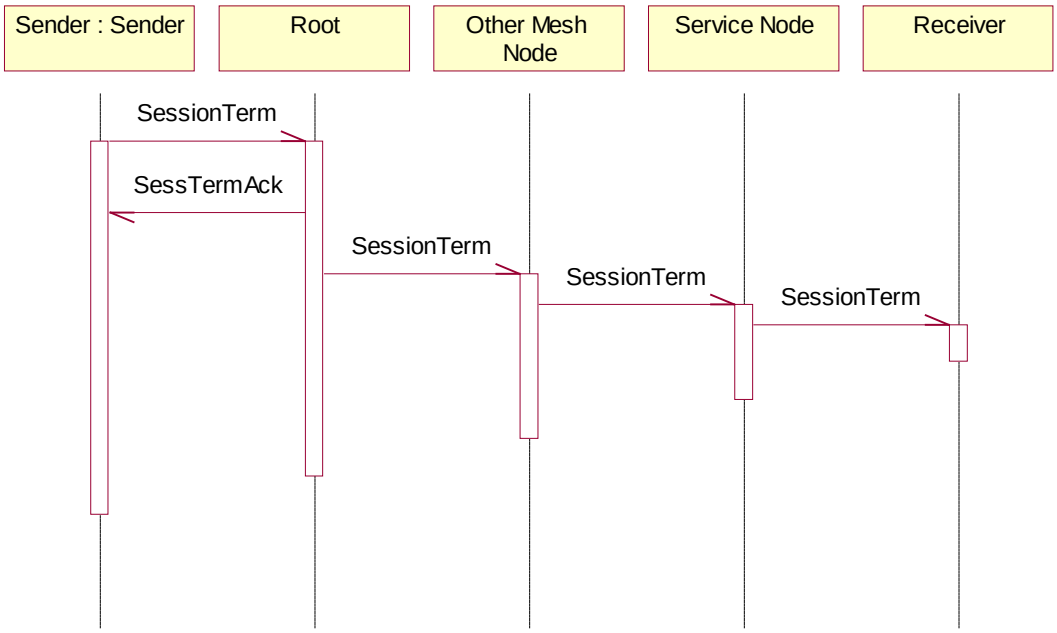



**Figure 16 Session Termination**

In the session termination process, as shown in Figure 16, when the sender finishes sending data of this session, it should inform the root about it by a SessionTerm message. The root responds the sender with a SessTermAck message and announces the session termination to all child nodes of this session by SessionTerm messages. The SessionTerm message will be

multicast to the group. If a node fails to receive this message, it can still find out the session termination by topology optimization mechanism of the session topology auto-configuration module. Therefore, group members do not need to respond to the SessionTerm message, and mesh nodes and local Service Nodes (SN) can stop forwarding data of the terminating session immediately.

Each node that receives the session termination information should first look for the session information in its session table. If the terminating session is in its session table, and its child table for this session is not empty, it should inform all its children by SessionTerm messages. A node should delete the entry for this session in its session table, release any buffer in its memory used for this session, delete the child table for this session, and update all other information relevant to the terminating session. If the node is a local SN or receiver, and there is no other session running on it, the node can leave the topology.

### 5.2.3 Data Forwarding Module

Data forwarding module is responsible for forwarding data packets to the destination. This module will receive data packets from other nodes, identify

the session for the data packets, check child information in the child table, and distribute the data to the children. If the node is also a receiver for this session, the packet is forwarded to the upper layer.

Another responsibility of the data forwarding module is to translate the packets between different multicast protocols, which will be discussed later in the chapter. The packet formats of different protocols may quite different and need to be translated at the borders between network domains where different multicast protocols are interconnected.

The functionalities in the module will meet the requirements of data delivery and scalability (collaboration with different distribution techniques), and this module will cover the data forwarding phase in the multicast session life cycle model.

### 5.2.4 Session Topology Auto-configuration Module

The Session Topology Auto-configuration module will automatically configure the topology, maintain the session tree, and optimize the topology, according to the algorithm for session topology auto-configuration we

introduced above. It will provide functionality for session topology auto-configuration phase in the multicast session life cycle model. It should meet requirements of group management (member join/leave) and scalability (large number of receivers, large coverage, inter-domain capability, collaboration with different distribution techniques, support for multiple groups)

The first task of this module is to automatically create the topology, and is discussed in detail in section 5.1. After a local service node or a receiver node finds a proper parent node and binds to the parent node, it does not need to find another parent node for another session, and all data packets for different sessions will come from the same parent node.

The second task of this module is to automatically maintain the topology. A node can find out failure of its neighbors by periodic Heartbeat messages. A node maintains a ***neighbor list*** that stores the information (IP address, port number, the last receipt time of heartbeat messages from a neighbor, etc.) about its neighbor nodes. A node should periodically send Heartbeat messages to its neighbors and wait for Heartbeat messages for neighbors. If a neighbor has not respond to heartbeat messages for a certain time interval

$T_{heartbeat}$, the node can assume that this neighbor node has failed. If a parent node is detected to be failed, the node should initialize the session topology auto-configuration module to find a new parent node. Otherwise, the node should update its neighbor list and session table to forbid data forwarding to the failed neighbor node.

The topology optimization is also done according to the session topology auto-configuration algorithm defined above. If data flow from its parent node decreases under a certain QoS level, or there are some other events that can significantly worsen the data receipt of a node, the node can choose and bind to another parent node. To fulfill this task, a node should keep monitoring the data flow from its parent node. If the flow control mechanism detects that the QoS parameters of the data flow drop to a certain level, it will inform the Session Control Module about the QoS degradation to the Session Control module, which triggers the session topology auto-configuration module to find a better parent node for this node.

```
Use session topology auto-configuration algorithm to find a parent node;
while(1)
{
        Monitor heartbeat messages from neighbors;
        if(a neighbor has not sent any heartbeat for T_heartbeat)
        {
                if(the neighbor node is the parent node)
                {
                        Use session topology auto-configuration algorithm to find an
                        alternative parent node;
                }
                else
                {
                        //a neighbor node fails
                        Delete the node from the neighbor list;
                        Delete the node from all session entries in the session table;
                }
        }
        Flow control mechanism monitors the data flow from its parent;
        if(QoS parameter drops under a certain threshold)
        {
                Use session topology auto-configuration algorithm to find an
                alternative parent node;

        }
}
```

**Figure 17 Pseudo Code for Session Topology Auto-configuration Module**

## 5.3 Support for Different Multicast Protocols

Many multicast protocols have been proposed and developed by the research community and telecommunication companies. These protocols have different design perspectives and focuses. As a result, the current multicast protocols use different topologies to organize member nodes, different control mechanisms to manage data flow, and obviously different packet formats for control and data packets (please refer to our previous discussion

in chapter 2).

Currently, there is no way to connect the heterogeneous multicast protocols together and make them work collaboratively. The scalability of multicast services will be constrained by this scenario significantly. First, in some domains, nodes cannot receive multicast streams carried by any other multicast protocol that the local routers do not support or not allow to run in the local domain. For example, nodes on two domains that exclusively allow PIM-SM and ESM cannot join the multicast groups in the other domain. Because different multicast protocols aim to solve specific types of multicast problems, another problem of the multicast protocol heterogeneity is that they cannot deal with other multicast problems independently. For example, multicast protocols in network layer (e.g., PIM-SM) cannot provide reliability without support from reliable multicast protocols. Therefore, we need a mechanism to connect different multicast protocols together, which is a new challenge for multicast technology.

To begin our discussion, we assume that each multicast-enabled domain allows at least one well-known or existing multicast protocol. This assumption allows other multicast protocols to be involved in the local

multicast traffic. The nodes in domains without multicast capability will join a multicast group by IP connections or other methods, e.g., AMT.

To connect heterogeneous multicast protocols together, we should consider where protocols should be connected, and how they can work together.

For the first question, we propose a solution based on the session management mechanism we introduced above. Our session management mechanism should run on the core network, which works as the infrastructure of the large-scale multicast topology, and multicast protocols in local domains can communicate with each other via our session management mechanism, as shown in Figure 18. Basically, the topologies used by existing multicast protocols are trees with a single root, which can be classified into two types, source-based tree and shared trees, according to our discussion above. Ideally, the connections between local multicast protocols and session management mechanism should be established on root nodes of local groups and the nearest mesh nodes on core network. The connection establishment could be static, which means that the connections are set between specific nodes by network operators before any nodes join the local groups, or dynamic, which means the connections are requested by

a specific node in a local group to a mesh node.

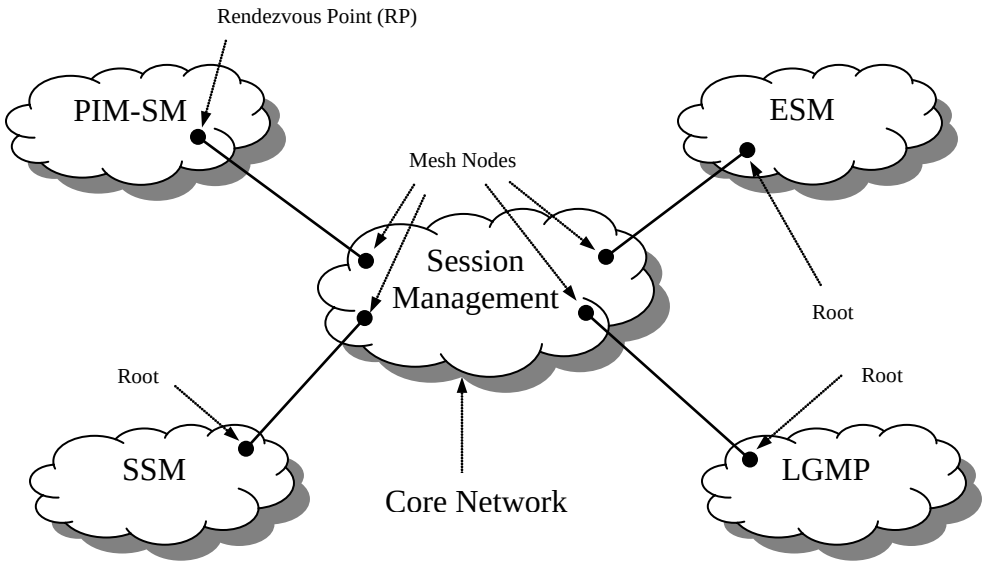
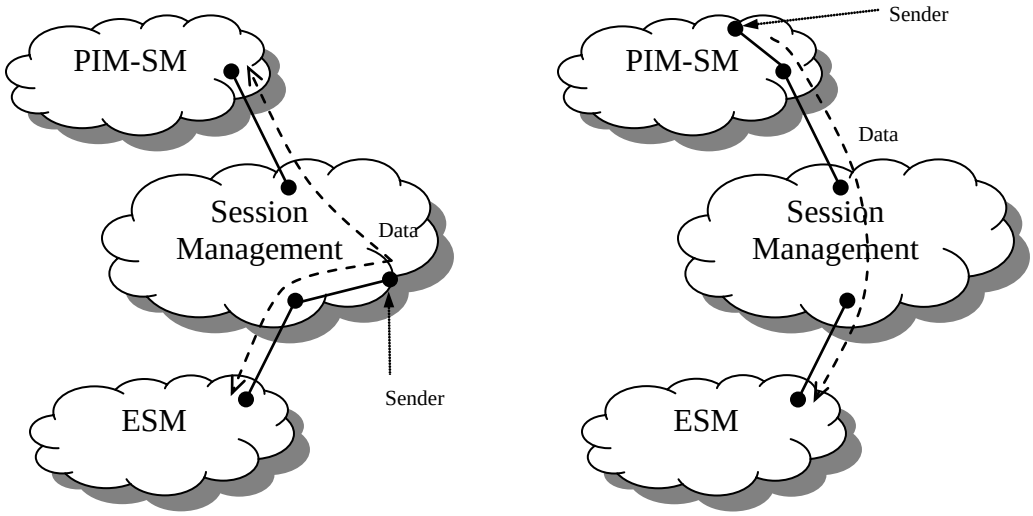



**Figure 18 Supporting Multiple Multicast Protocols**

Our session management is an excellent choice that can connect different multicast technologies together. First, it can control all aspects of multicast, so multicast technologies in network layer, transport layer, and application layer can be merged seamlessly. Even if some aspects are missed in some multicast technology, e.g., reliability missed in network layer multicast, the functionality can still be supported in the core network and other domains. Second, new technologies can be easily supported in the future. Second, because our session management covers all phases of multicast sessions, it is easy to implement and install new modules for new technologies for any phase without changing existing software modules. Third, it can provide a good infrastructure for large scale deployment of multicasting. As we can

see, the support of an infrastructure is important for multicasting. Even some technology without infrastructure, e.g., some overlay multicast protocols, can benefit from it.

Next, we need to figure out how protocols can be connected to our session management mechanism. The answer to this question is not as straightforward as it seems to be. Heterogeneous multicast protocols use various topologies, and nodes on these topologies have different functionalities. We need to map those topologies and functionalities onto our solutions, so the protocols can be connected to our solution smoothly. The mapping has three aspects: topology mapping, packet translation, and functionality mapping.




(a) Sender uses our solution (b) Sender uses other multicast technologies

**Figure 19 Session Sender Location**

In our solution, the Session Topology Auto-Configuration module will map different sub-topologies required for different protocols into the topology generated by our solution. The mechanism of connections between our mesh and local protocols needs some changes in nodes' functionalities. The location of the session sender will affect this connection mechanism, too.

If the session sender uses our session management mechanism, session packets will first flow on our solution and then be forwarded to the local domains running other protocols, as shown in case (a) of Figure 19. The mesh nodes connected to the local groups can work as an external source for the local groups that do not use our Session Management mechanism. For shared trees in these local groups, this idea would not affect the local group much, because the root of the tree topology in such local groups are core nodes shared by multiple source nodes and ready to receive packets from any source node and forward them to receivers. However, in a source-based tree, the root is only a source of a multicast session, and has not the capability to receive packets from other sources. Therefore, for source-based tree, the functionality of the root must be extended to receive packets from an external source and forward them to receivers, or we need to create a special node in the local group, which works as a root and receives packets

from mesh nodes.

If the sender node of a session is located in a local domain running a multicast protocol other than our Session Management mechanism, as shown in case (b) of the above diagram, the connection mechanism is a little different from case (a). The nearest mesh node connecting to the sender's local domain should become the root node on the mesh for this session, and be responsible for all session management tasks.

To support different multicast protocols, we must consider how to translate the packets between protocols. The Data Forwarding module will translate the packets from one protocol's format to other protocols' formats. The translation will occur at the mesh nodes that are connected to local groups and only involve the packets that need to be transferred between domains. The local protocols must be well-known protocols, in which packets' functionalities and formats are fully standardized and understandable. Therefore, the mesh node can translate the local packets into packet formats used in our session management mechanism, and vice versa. To fulfill the translation, we need to know full definitions of every bit of the packets and the functionalities of the packets for the local protocols. We need to build a

translation mechanism for each different multicast protocol.

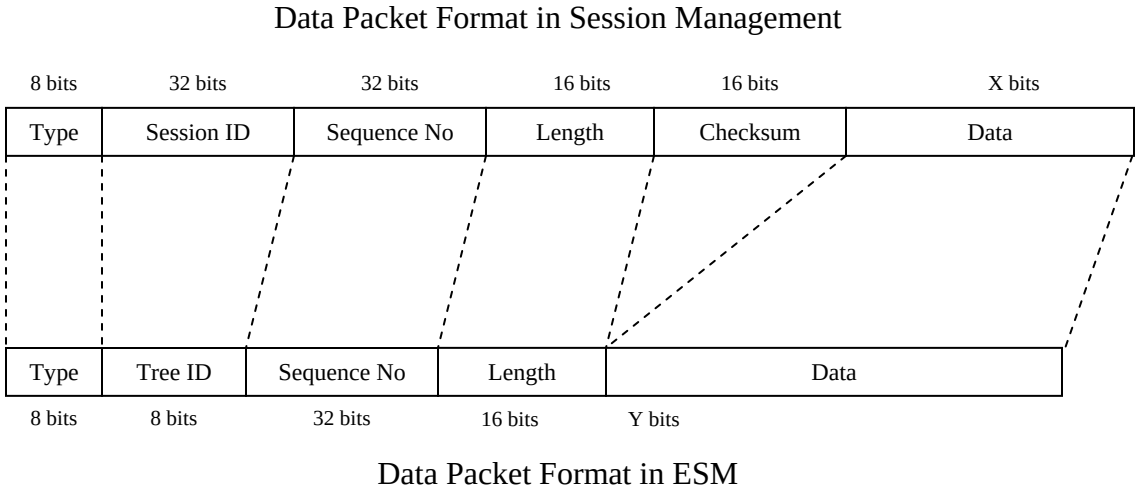



**Figure 20 Translations of Data Packets**

Here, we will give an example of the translation mechanism, which is the translation of data packets between our multicast session management and ESM. This translation is used in our simulation, which will be introduced in the next chapter. Because we do not have a formal protocol specification of ESM, the data packet format is designed by ourselves according to the ESM description in [26]. Both formats are quite simple and only include some necessary fields.

The translations between these two formats are shown in Figure 20. For fields with similar meanings, lengths, and functions, such as Type, Sequence Number, and Length, the translations are relatively simple. We can define an

invertible function that converts a value in field $f_1$ of format A to a value in field $f_2$ of format B, or vice versa.

For fields with similar meanings but different lengths, the translation function must also be invertible. Some special techniques are required to deal with such cases. For example, in the above diagram, our solution uses Session ID with 32 bits and we designed the Tree ID in ESM with 8 bits, which identifies a distribution tree in ESM. To translate the packets and map the functionalities of two protocols, we either limit the number of sessions running on the ESM domains, or add some special optional fields in ESM data packets, which allow ESM to identify different data streams with identical Tree IDs. The design of such invertible functions may vary for different situations and is out of scope for this paper.

For fields in format A without corresponding fields in the format B, such as checksum field of the first format in the above figure, they can be ignored in the translation from A to B. However, when the translation is from B to A, these fields must be recalculated and refilled.

When a data packet is sent from our Session Management mechanism to

ESM, data fields will be retrieved from the original packet. If the data length exceeds the ESM maximum data length, the original data will be divided into smaller pieces to fit the ESM data packet length limit. The Data Forwarding module will calculate new sequence numbers for the ESM data packets. Other fields will be recalculated and refilled according to the above discussion. After the translation, the packet will be sent on the ESM distribution tree. A similar translation will happen if a data packet is sent from ESM to our solution.

The functionality mapping of multicast session is a difficult problem for supporting heterogeneous multicast protocols. A lot of important functions in a multicast session require smooth and tight collaborations among all nodes in different domains running different protocols. These functions include reliability, security, AAA functions, and so on. In this project, our discussion will only focus on reliability function mapping as an example.

For applications requiring low latency and relative low data loss, or for applications with high reliability requirement, it will be a critical problem to support reliability. There are some aspects in reliability mapping we must consider. First, different multicast protocols in local domains may have

different support for reliability. As we have discussed above, all multicast protocols designed in the network layer are focused on data transfer, and have no reliability mechanism, e.g., PIM-SM. For reliable multicast protocols designed in the transport layer, they are focused on reliability of a multicast session, e.g. RMTP-II. Second, the multicast protocols in local domains may use various techniques for reliability. For example, RMTP-II uses a tree-based ACK scheme, called TRACK, to aggregate the error reports and send reports to the upper layer nodes. Currently, the IETF working group for reliable multicast transport (RMT) is working on NACK and FEC code, which we have already introduced above. Therefore, a more general solution of reliability needs to collaborate with the domains with or without reliability support, and needs to coordinate different reliability techniques, too.

The heterogeneity of reliability functionality in different multicast protocols requires us to find a solution, which allows each multicast session to define reliability for itself and collaborate with different multicast protocols in local domains to support reliability. In our session management mechanism, the flow control module will take care of reliability of the session. Multiple flow control schemes, e.g., FEC and Peer-to-Peer, can be integrated into our

solution as sub-modules controlled by flow control modules. The flow control scheme and reliability parameters of a session are selected at the beginning of the session by the sender or root node according to the reliability requirements of applications. The root node will put the reliability information into the session announcement, and all other nodes in the session will follow the selected flow control scheme.

For local domains running protocols without reliability capability, because adding reliability to the protocols must change the protocols significantly, the application using multicast techniques should be responsible for the reliability. It can fulfill this task by its own reliability functions or by support from other reliable protocols. For local domains with reliability capability, the protocols must provide the same level of reliability required by the multicast session. It means that the local protocols must have the same reliability settings, e.g., QoS parameters and flow control scheme, as the settings in our session management mechanism. At the beginning of a session, the local domain root node connected to the mesh will be informed by the mesh node about the QoS parameters and flow control scheme. If the flow control scheme is not supported in the local domain, the local flow control scheme can be used with the received QoS parameters. In such a

case, the root node still needs to use the pre-set flow control scheme to collaborate with the mesh node. The mesh node will also be responsible for retransmitting the lost packets reported by the local nodes.

In Figure 21, the state chart shows the process of supporting different multicast protocols. Each rectangle in this diagram represents a specific state in the session life cycle, and the actions for supporting heterogeneous multicast protocols in this phase are represented as sub-states in this diagram. The actual work flow of this process will go along with the process of session life cycle.

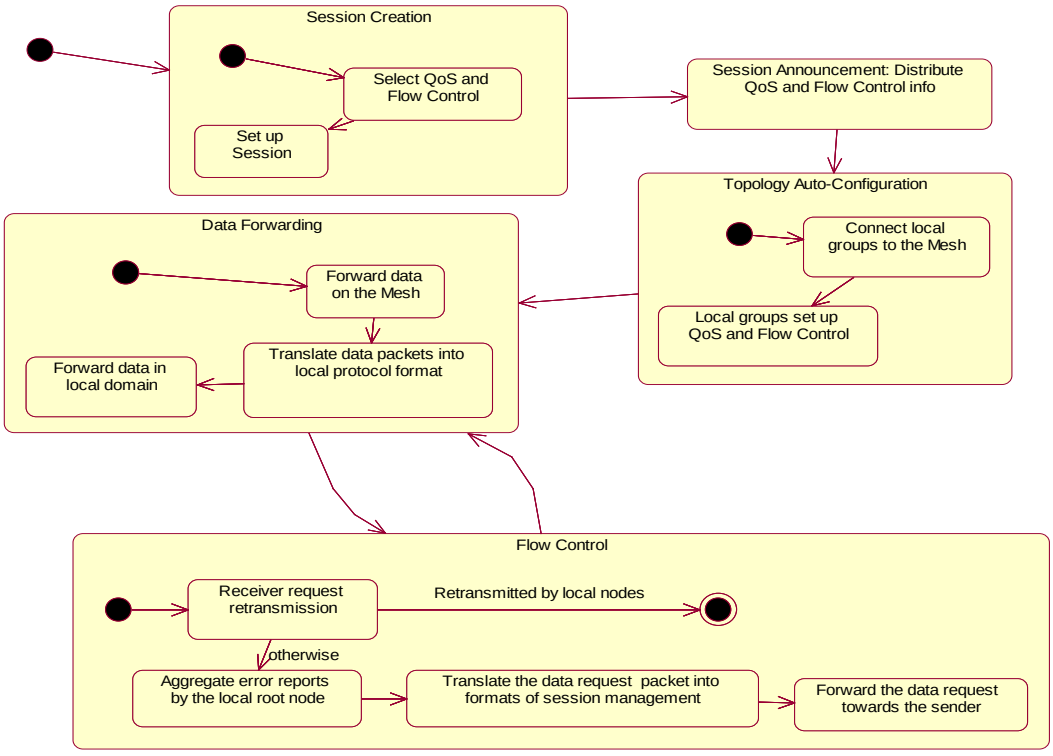



**Figure 21 State Chart of Supporting Different Multicast Protocols**

The process of supporting multiple protocols starts with the session creation. The sender or the root node of the session will choose the QoS and flow control settings in session creation phase. These settings will be put in the session announcement and be propagated on the mesh. When a local domain running another multicast protocol tries to join the session, the root of the local domain will set up QoS and flow control according to the session announcement retrieved from the mesh. After the local domain joins the topology, the data forwarding module will translate the data packets and forward them to the local domain. The root of distribution trees for the local domain will be responsible for forwarding data packets and aggregating error reports. Some lost data packets can be recovered by local nodes. Others will be requested by the root node from upper stream node in the topology. The data requests sent to our session management mechanism will be translated into the packet formats of our solution and will be forwarded towards the sender. One service node or the sender will do the retransmission of lost packet backwards to the local domain. The retransmitted packet will follow the similar path of regular data packets.

The techniques for supporting heterogeneous multicast protocols take place

in the session topology auto-configuration and data forwarding phases in the multicast session life cycle model. The techniques will significantly help multicast technology to meet scalability requirements we summarized in chapter 3.

Until now, we have introduced all of our design for multicast session management. To help readers understand the flow of ideas we introduced above, I summarize the relationship among requirements, life cycle phases, and our design, shown in table 7. In this table, reader can find the life cycle covered by this project, and which requirements should be met in a life cycle phase. The more important relationship in this table is mapping between requirements and parts of design covering the requirements.

| Life Cycle Phase | Requirements | Design |
|---|---|---|
| Session announcement and session creation/termination | Group management (dynamically and automatically create/terminate a group) | Session Management Mechanism |
| Session topology auto-configuration | Group management (member join/leave) | Hierarchical topology auto-configuration, Session Management Mechanism |
| | Scalability (large number of receivers, large coverage, inter-domain capability, collaboration with different distribution techniques, support for multiple groups) | Hierarchical topology auto-configuration, Session Management Mechanism |
| Data forwarding | Data delivery, Scalability (collaboration with different distribution techniques ) | Support for different multicast protocols, Session Management Mechanism |
| Flow control | Reliability | Session Management Mechanism |
| Almost every phase | Deployment (work with different underlying hardware and software, and ease of deployment, i.e., incremental deployment) | Session Management Mechanism |

**Table 7 Relationship among Life Cycle Phases, Requirements, and Design**

# 6 Simulation

To support our research and design, we established a simulation plan, which simulated our solution and compared it with two other multicast technologies, PIM-SM and ESM. The simulation is created and tested in Opnet Modeler, a commercial telecommunication simulation platform. The results of this simulation show positive support for our research.

## 6.1 Simulation Purposes

Our simulation has two main goals, feasibility checking and performance evaluation. The feasibility checking is to prove the feasibility of our solution. The performance evaluation is to evaluate the performance our solution and compare its performance with other multicast techniques.

The feasibility checking will check important features of our solution. These features include:

1. Topology auto-configuration: Mesh and session tree auto-configuration, which are the algorithms we developed before.
2. Session management: the capabilities of controlling all phases of a multicast session that follows the processes of the session life cycle we defined in section 3.3.

3. Connecting heterogeneous multicast techniques: the topology mapping, the packet translations, and the reliability function mapping.

The performance evaluation has three steps:

1. Building a simulation model for our solution.
2. Monitoring and analyzing the model's performance.
3. Performance comparison with overlay multicast and IP multicast.

The simulation will provide evidence of our solution's capability of satisfying a set of requirements: data delivery, group management, reliability, scalability (inter-domain capability, collaboration with different distribution techniques), and deployment (working with different underlying hardware and software, and ease of deployment, i.e., incremental deployment).

The simulation will cover several phases of the multicast session life cycle model, including session creation, session announcement, session topology auto-configuration, data forwarding, flow control, and session termination.

## 6.2 Simulation Plan

We choose Opnet Modeler as the platform for our simulation because of its

extraordinary environment and functions. The simulation contains three connected subnets. Each of the subnets uses a different multicast technology, our session management, PIM-SM, and ESM.

### 6.2.1 Platform

Opnet Modeler provides a comprehensive development environment supporting the modeling of communication networks and distributed systems. Both behavior and performance of modeled systems can be analyzed by performing discrete event simulations [39].

In Opnet Modeler, all objects are organized into different hierarchical levels of a model, network, node, process, link, packet, etc. Opnet Modeler provides different graphical editors for those objects. The editors can help the user to design and model objects' features and behaviors, organize all objects in a domain, and provide and manage interfaces and statistics to the simulator.

Opnet Modeler also provides a set of data collection and analysis tools, which help the user to evaluate the performance of a model very conveniently and efficiently. Opnet Modeler allows the user to collect not only global statistical samples for a model, but also local samples within all

level objects including nodes, processes, etc. The collected data samples can be drawn into different formatted graphical charts according to the user's requirements.

### 6.2.2 Topology

In our simulation, we will compare the performances of three different multicast technologies, PIM-SM (section 2.2.2.5), ESM (section 2.4.2.1), and our solution. PIM-SM and ESM are representative techniques of IP multicast and overlay multicast. To fairly evaluate and compare their performance, the techniques should work independently in a similar environment. Therefore, we design a special topology for our simulation plan. In this topology, three different multicast technologies run independently on three domains, as shown in Figure 22. These domains are connected to each other and have similar topologies. The modeled distance between two domains is 500 kilometers. Each of the three domains will be a single autonomous system and will use BGP to exchange inter-domain routing information.

One domain, as shown in Figure 24, will run our session management mechanism and have native IP multicast support on all routers and gateways.

Another domain will only enable PIM-SM protocol on all routers and gateways, displayed in Figure 23. In Figure 25, the last domain has no native IP multicast enabled on routers and gateways, but ESM is installed on all hosts.

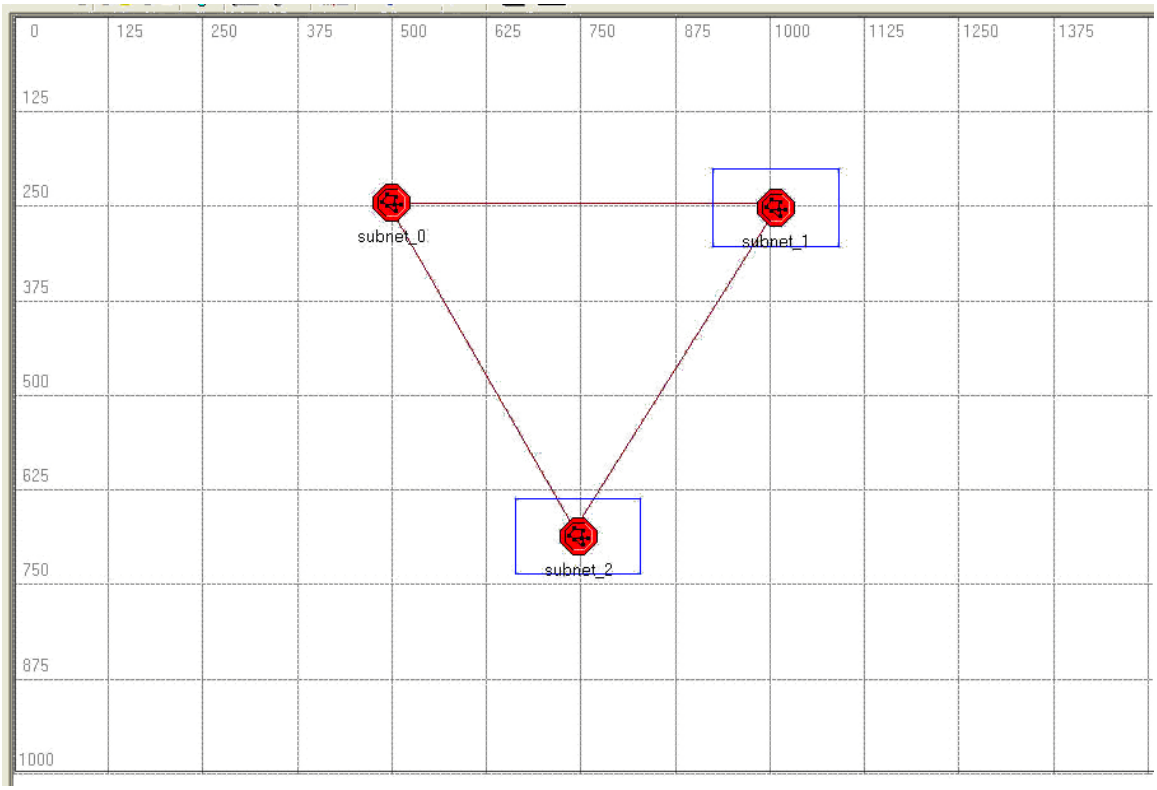



**Figure 22 Simulation Topology**

The topologies in these domains are similar. Each domain has 4-6 routers and 12-13 gateways. The routers establish the infrastructure of each domain. The gateways are directly or indirectly connected to the routers. Each gateway controls a LAN, which has 1-3 receiver nodes. Those routers, gateways, and receivers are built on node models provided by Opnet

Modeler. The routers are derived from a model based on Cisco Series 4000 routers, gateway nodes are originated from an Ethernet gateway model, and receivers are created from an Ethernet workstation model.

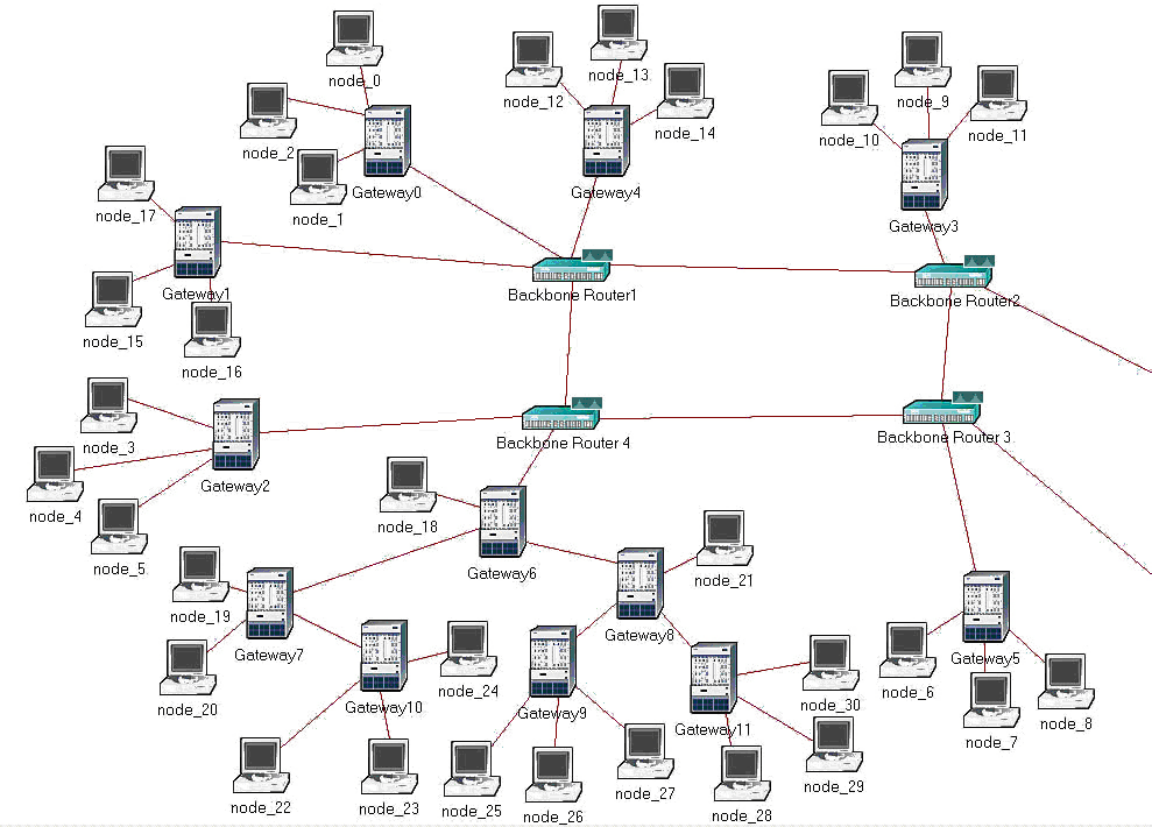



**Figure 23 Topology of Domain 1 (PIM-SM)**

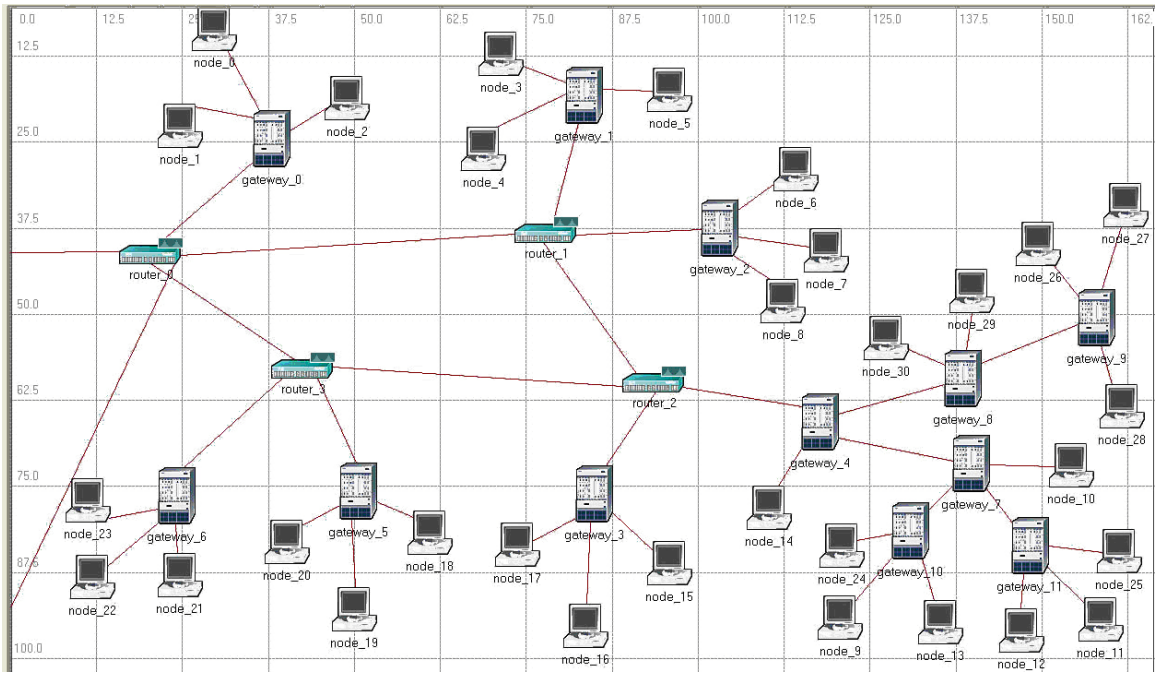



**Figure 24 Topology of Domain 2 (Session Management)**

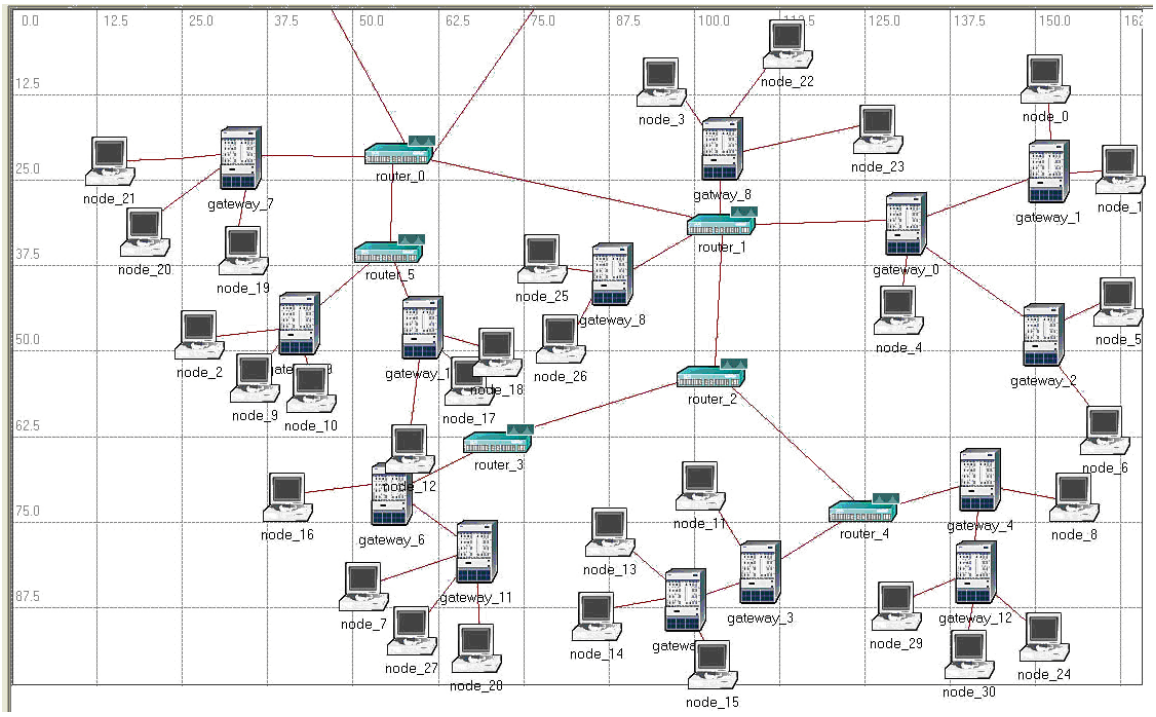



**Figure 25 Topology of Domain 3 (ESM)**

The connectivity within each domain is different from other domains, but has same effects on multicast topology. Each domain has a root node

(***Backbone Router2*** in Domain 1, ***router_0*** in Domain 2, and ***router_0*** in Domain 3), which forward data to all other nodes. Other nodes will be organized into a tree managed by the multicast techniques running on that domain. Each receiver node will get data packets from its parent node. To fairly compare the performances, each domain has an identical number of receivers at the same level of the distribution tree. To be at the same tree level, two nodes will have the same number of intermediate nodes on the path from the local root to itself. For example, each domain has exactly three receivers that have only one intermediate node (a gateway) on the path between the root and itself. The next table shows the number of receivers with the same intermediate nodes in each domain.

| Number of Intermediate Nodes | Number of Receivers in Domain |
|---|---|
| 1 | 3 |
| 2 | 12 |
| 3 | 4 |
| 4 | 3 |
| 6 | 9 |

In the simulation, the connections between specific types of nodes have similar properties. The connection between two routers is duplex Ethernet connections operating at 1000 Mbps. The connection between a gateway and a router (or another gateway) is duplex Ethernet connections operating at

100 Mbps. The connection between a gateway and a receiver is duplex Ethernet connection operating at 10 Mbps.

As you can see in the above topologies, there is not any other host in each LAN. The first reason for designing such topologies is that it can help us focus on the topology of multicast distribution tree and data traffic, which are our real interest in this simulation. The second reason is that we can simulate the effects of other hosts on multicast techniques by adding background traffic on each links, so there is no need to draw non-multicast hosts in the topologies. The background traffic loads are shown in the following table. In addition, we did not simulate different topologies in the LAN, e.g., star and bus. We focus our research and programming efforts on the multicast techniques for a relative larger area, instead of the effects of LAN topologies on multicasting.

| Type of Connection | Average Background Data Loads |
|---|---|
| 1000 Mbps Ethernet Link | About 620,000,000 bits/second in each direction |
| 100 Mbps Ethernet Link | About 45,000,000 bits/second in each direction |
| 10 Mbps Ethernet Link | About 1,500,000 bits/second in each direction |

### 6.2.3 Models

In our simulation, we use three different multicast technologies: PIM-SM,

ESM, and our solution. We used and modified some models provided by Opnet Modeler, and designed several new models to simulate new techniques. In this section, we will discuss these models in our simulation.

First, we will introduce the model of our solution, which is the kernel of our simulation. We designed four types of nodes: sender node, mesh node, local service node, and receiver node. Most functions of these nodes are introduced in chapter 5 and designed according to the state diagrams shown in that chapter. We ignored the functions related to AAA, security, and other aspects out of scope. Each node type uses some standard node models provided by Opnet, e.g., mesh node is based on Cisco 4000 series router model, receiver and sender are built on advanced Ethernet workstation model, and local service nodes are originated from an Ethernet gateway model. In each node model, we added a process model that simulates a protocol entity of session management.

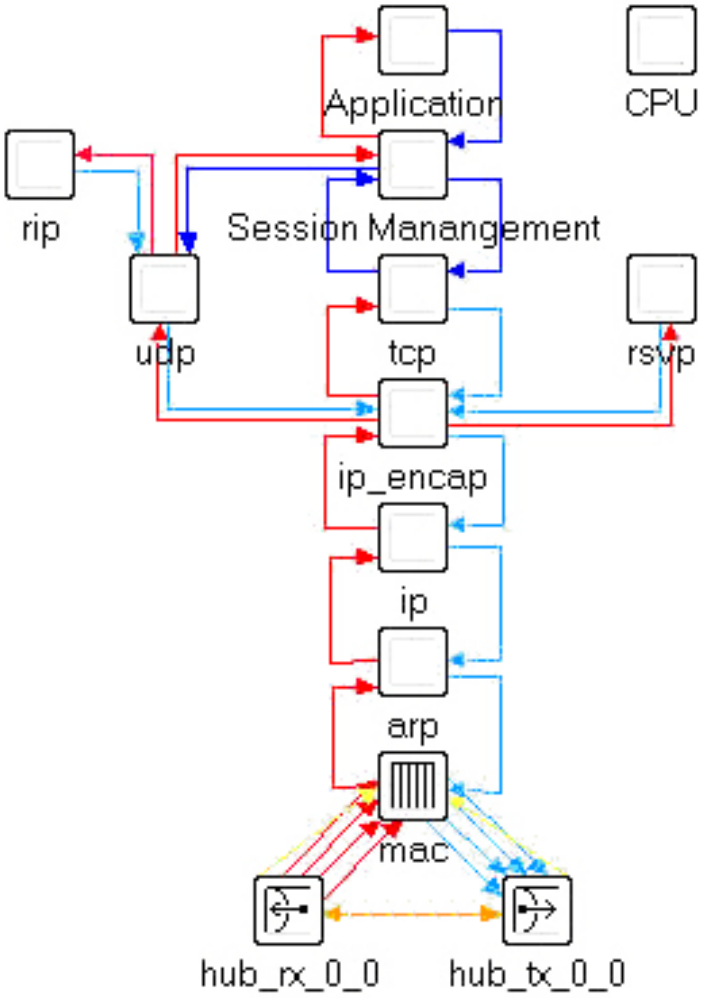



**Figure 26 Session Management Receiver Node Model**

Figure 26 presents the node model of a receiver model. As we can see in this model, our solution is built upon UDP and TCP models, instead of IP model, which is the best place for our solution. The reason for this is a technical issue, rather than a protocol design issue: it is much more difficult to design a process collaborating with IP models in the Opnet Modeler. Please refer to chapter 5 for details of the design of our models.

Now, we will introduce PIM-SM and some modifications we made. The PIM-SM multicast techniques are included in the model library provided by Opnet Modeler. It follows the PIM-SM v2 IETF RFC 2362 [41], which is an old version designed in 1998 but has most features of current version. In Opnet Modeler, PIM-SM cannot work alone without support of IP, IGMP, and applications models. For instance, the PIM-SM should join a multicast group identified by a valid class D address, and the packets received by a receiver should be delivered to valid video or audio applications. Basically, this mechanism will not be completely suitable for our purposes, which is to use PIM-SM to build a multicast group within a domain and transfer packets from an outside source to all receivers. We made some changes: the model of the RP can accept packets from outside sources; and receivers just join a group and wait for packets sent to the group, but do not need to forward multicast packets to a certain application.

For ESM protocol, we designed some new models to simulate it. Because a formal protocol specification for ESM was not available to us, we implemented the Narada protocol model that is described in the paper “A Case for End System Multicast” [27]. We designed two types of node in

ESM, root and receiver. Please refer to section 2.4.2.1 for details of the ESM techniques. However, to simplify our design, we ignore some features of End System Multicast. First, because our purpose in this simulation is not for testing multicast usage in Multimedia data transfer, we did not implement the Multiple Description Codec (MDC) or the multiple disjoint tree structure, which are used to deliver multiple video and audio streams with different qualities. Second, we allow the root node to collect and propagate information about all members in the ESM group, which was described as an out-of-band bootstrap mechanism in ESM. Third, like PIM-SM, ESM root node can accept data packets from an outside source and forward to receivers. Finally, to check the capability of supporting different multicast technologies, we implement a simple reliability mechanism in ESM, which can collaborate with our session management mechanism.

### 6.2.4 Scenario

In our simulation, there is only one multicast session, one source for this session, and different multicast protocols are used in each domain. The domains are connected by our session management mechanism. The session will run for a fixed period. The reliability is required by ESM and our solution.

First, we create only one session. All receivers and service nodes are listening to this session. The session information are created and “broadcast” within our mesh. There is only one sender in the scenario, which is node_0 located on domain 2 running our solution, as shown above. This sender will be connected to a mesh node, router_0 in the same domain. The mesh node is statically set for this sender.

As we introduced above, there are three types of multicast technologies that are used in different domains in this scenario. Our session will be supported in these domains. To support this session, at the borders between our solution and PIM-SM or ESM, the mechanisms of supporting different multicast protocols are built in the RP node for PIM-SM or the root node of ESM.

The session will last for 30 minutes, and the total simulation will last for 40 minutes. First, the sender will join the topology at the beginning of the simulation. The sender will continue send data packets and retransmit requested packets. The sending rate of the sender is 1 packet per second. The receivers will start to send join request at approximate 10 seconds after the

session is created. A receiver will leave the session when it received 1800 valid packets.

To completely test the capability of our solution, we designed some special tests of reliability in our simulation. Every connection in this scenario will simulate 5% packet loss. The flow control scheme in this scenario will guarantee 100% reliability in domain running ESM and our solution, which means that all lost packets will be requested by receivers and be recovered by upstream nodes. We do not guarantee the reliability in a domain running PIM-SM. Because PIM-SM is a multicast protocol in Network layer without flow control scheme, we assume that the reliability in that domain will be controlled by applications using PIM-SM.

## 6.3 Results and analysis

In this section, we will show the simulation results and the analysis on these results, compare the performances of the multicast techniques, and make our conclusions of the simulation.

The pictures used in this section are generated by Opnet Modeler based on statistical data collected in our simulation. To fairly compare the

performance of different multicast technologies, the statistical data are collected only for each domain. For instance, in ESM and PIM-SM domains, the data packet delay are only to be calculated when data packets are received by the root of local distribution trees. The delays between the sender and the root of local distribution trees are ignored.

The first task of our simulation is to prove the feasibility of our session management mechanism. In our simulation, the results show:

1. Topology auto-configuration: The Mesh is configured manually, and the session tree is automatically configured according to the algorithm we introduced in chapter 5. The results indicated that our topology auto-configuration is feasible, and its performance is excellent comparing with ESM, which we will show in Figure 36 – Figure 39.
2. Session management: The results of simulation show our solution's capabilities of controlling all phases of a multicast session. A session can be successfully created in the session creation phase, and the information about the session can be propagated on the mesh. The auto-configuration modules can organize the nodes into our hierarchical topology quickly and efficiently. The data forwarding functionality of each node can consistently forward data packets to

downstream nodes. The flow control can effectively recover lost data packets. The session can be terminated when the session is over.

3. Connecting heterogeneous multicast techniques: The capability of supporting different multicast technologies has been proven in our simulation, including the topology mapping, the packet translations, and the reliability function mapping. PIM-SM and ESM are connected to our session management mechanism successfully, data packets and control information packets are translated between different formats smoothly, and the flow control functionality has effectively worked among different multicast technologies.

Some detailed results will be shown in the section of performance comparisons of different multicast technologies.

The topology auto-configuration proves that our algorithm can effectively satisfy the requirements of group management (member join/leave) and scalability (inter-domain capability). In our simulation, the algorithm can deal with member joining/leaving efficiently in the domain running our session management mechanism, and data packets can be transferred across the borders between domains.

Our session management mechanism can meet the group management requirements of dynamically and automatically creating/terminating a group, by successfully supporting the session creation, termination, and announcement phases in the multicast session life cycle model.

Flow control module in our design has been proved to meet the reliability requirements of 100% reliability with no time bound. The ESM and our session management mechanism domains support 100% reliability for multicast session. In our simulation, the desired reliability level is successfully guaranteed in both domains.

The results of connecting heterogeneous multicast techniques in our simulation prove that our solution can satisfy the scalability requirements of collaboration with different distribution techniques. In our simulation, PIM-SM, ESM, and our session management mechanism can seamlessly work together. The three levels of mapping discussed in chapter 5 are successfully executed. The heterogeneous multicast techniques will significantly improve inter-domain capability for multicast technology, too.

Our simulation shows that our solution can transfer data across the domains

with different protocol models (PIM-SM and ESM), and across various kinds of workstations, gateways, and routers. The results indicate that our solution can work with different underlying hardware and software, which is an important deployment requirement for multicast technology. Another deployment requirement satisfied by the simulation results is ease of deployment, i.e., incremental deployment. Our solution can extend the coverage of multicast from native multicast domains to domains without IP multicast support, which allows ISP to gradually deploy multicast technology from core network to different local domains.

The second task of our simulation is to compare the performances of different multicast protocols. First, we will compare the data transfer of PIM-SM, ESM and our solution. For other aspects, e.g., flow control capability, because PIM-SM does not have those capabilities or there are some programming difficulties, we will only focus our discussion on ESM and our session management mechanism.

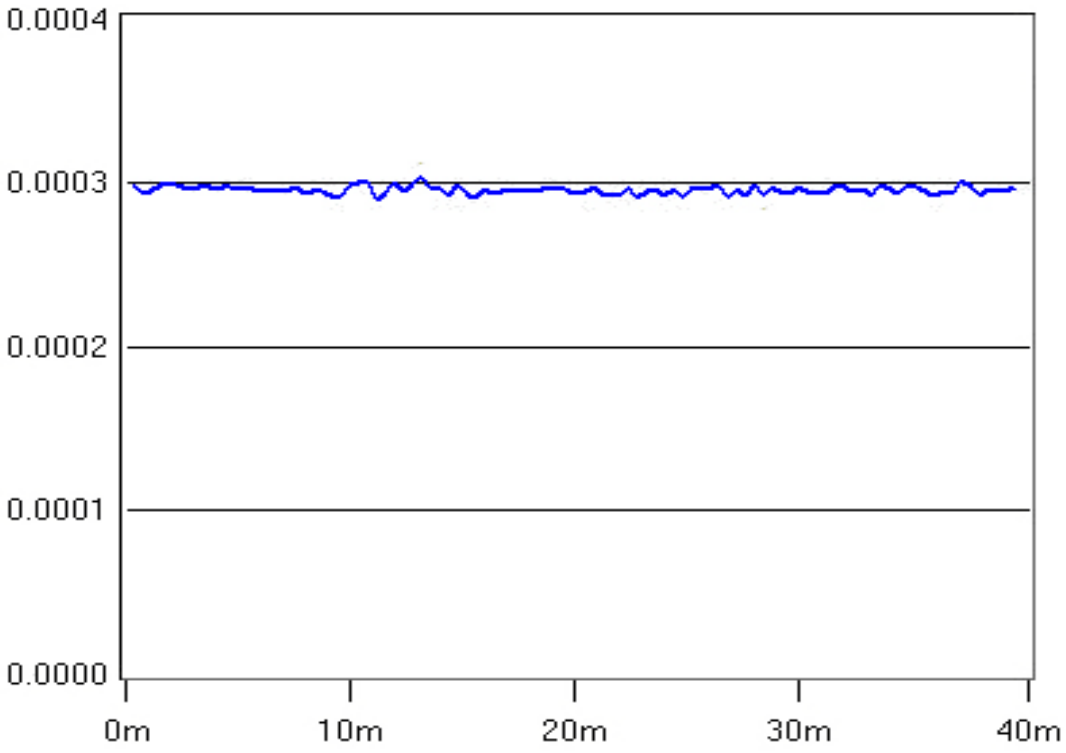



**Figure 27 PIM-SM Average Data Packet Delay (Seconds)**

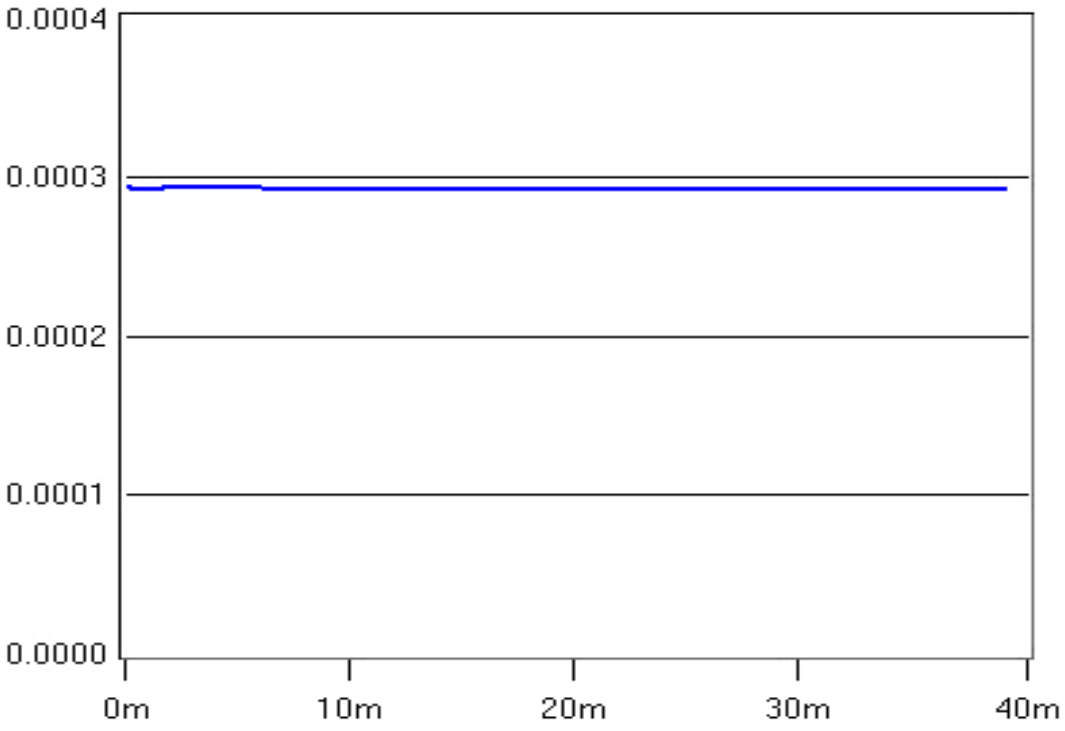



**Figure 28 Time Average of PIM-SM Data Packet Delay (Seconds)**

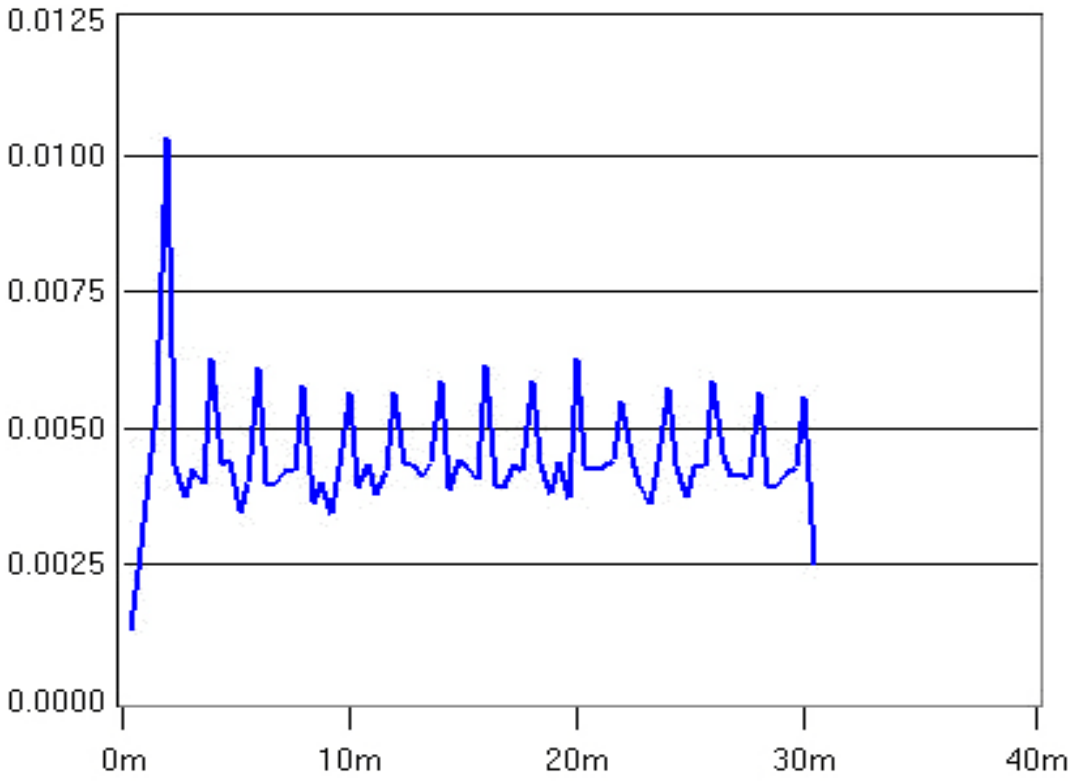



**Figure 29 ESM Average Data Packet Delay (Seconds)**

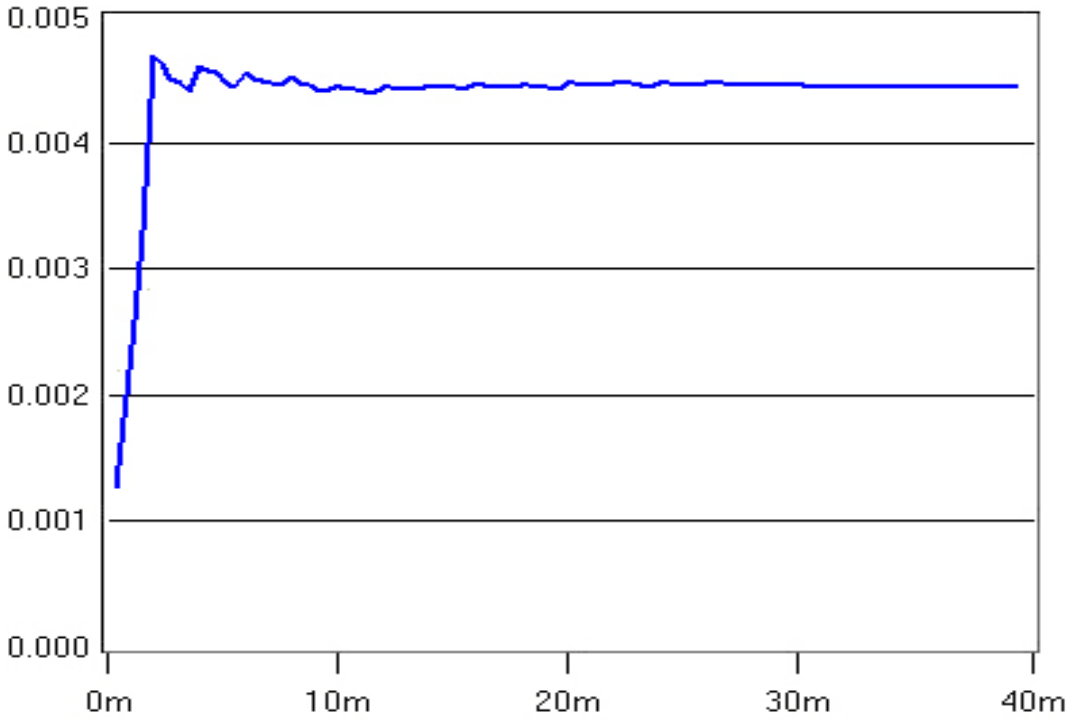



**Figure 30 Time Average of ESM Data Packet Delay (Seconds)**

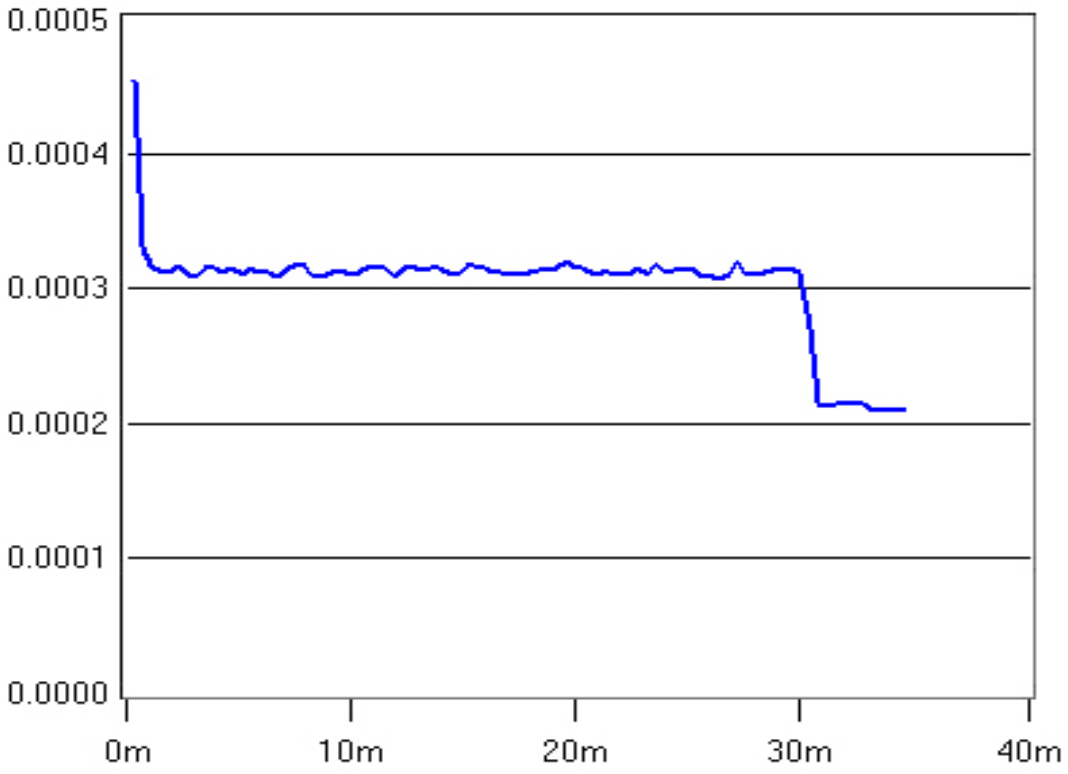



**Figure 31 Session Management Mechanism Average Data Packet Delay (Seconds)**

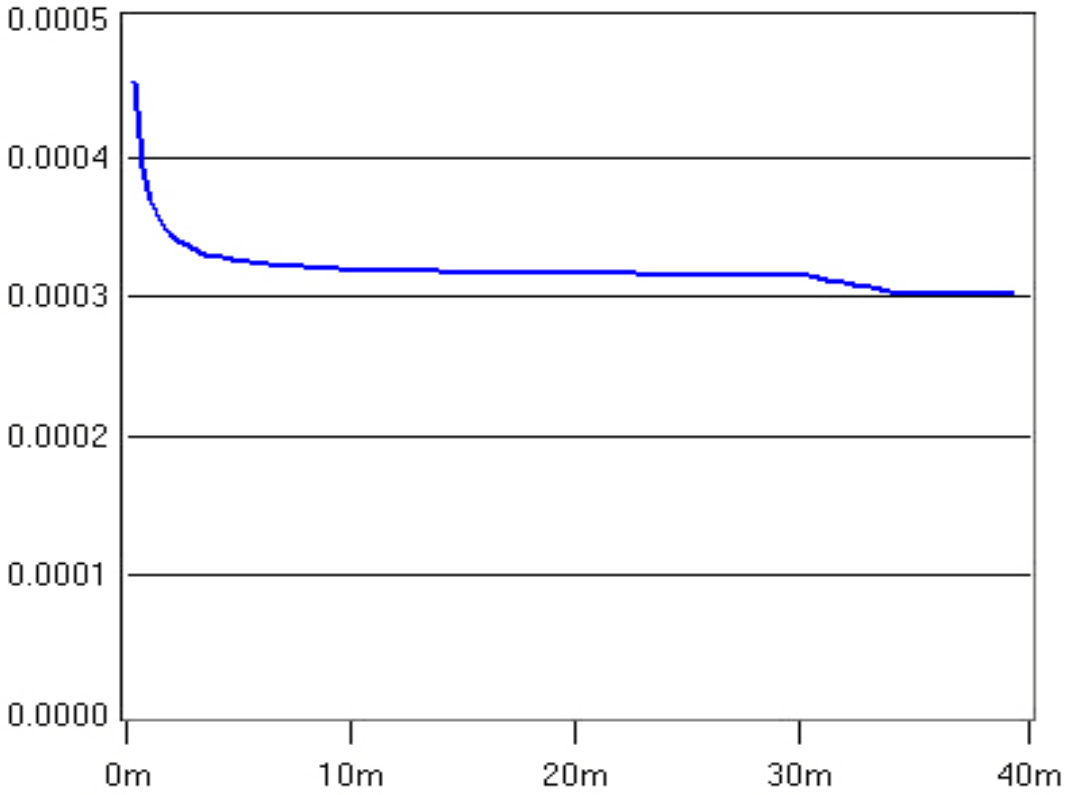



**Figure 32 Time Average of Session Management Mechanism Data Packet Delay (Seconds)**

The above six diagrams show the average delay of data packets and time average of the data packet delay for PIM-SM, ESM, and our session management mechanism. The average data packet delay indicates the mean of data packet delays collected at a specific time in our simulation. The time average of data packet delay shows the trend of data packet delay changes during a period of time. The average delay of data packets and time average of data packet delay are measured in seconds shown in y axis, and the value of x axis in the diagrams is the elapsed simulation time in minutes.

As we can see in figure 27, the PIM-SM has the lowest average data packet delay, about 0.0003 second, and the time average curve of data packet delay shown in figure 28 for PIM-SM is very stable. It means that the PIM-SM is the fastest multicast techniques of the three techniques compared in our simulation, and it has a steady performance in data packet transfer.

As we can see in figure 29, ESM is the slowest multicast techniques in our simulation. At the beginning, the average data packet delay will increase when more and more receiver nodes join the distribution tree, and its peak is over 0.01 second. ESM average data packet delay changes remarkably from

time to time. The lowest average data packet delay is below 0.005 second. One of reasons for its unstable curve is that ESM randomly optimizes its topology and the data packet delays are affected by its topology changes. The time average of data packet delay of ESM in figure 30 shows that the range of data packet delay is between 0.004 second and 0.005 second when all receiver nodes joined.

We can also find out a fact from the above diagrams that our session management mechanism is the second fastest multicast techniques. In figure 31, at the beginning, the average delay of data packet reaches its peak, over 0.0004 second. As long as more and more receiver nodes join the session tree, the average delay of data packet will decrease and maintain at about 0.0003 second. At the end of the session, the average delay of data packet will decrease to the lowest value, about 0.0002 second. The average delay of data packet is a relatively smooth curve. The time average of data packet delay is distributed between 0.0003 second and 0.0004 second, as shown in figure 31.

In conclusion, PIM-SM is the fastest multicast mechanism, the speed of session management mechanism is almost at the same level as PIM-SM, and

the ESM is the slowest and almost 10 times slower than the other two techniques. The reasons for the results are the features of the different multicast technologies. First, the PIM-SM is located in the network layer, and ESM and our session management are located in higher layers. As a result, PIM-SM packets go through fewer layers in each node. Second, PIM-SM and our session management mechanism have support from underlying infrastructures, i.e., PIM-SM gets router support in IP network and the session management mechanism build a Mesh as its infrastructure. The consequences of infrastructure support are better topologies and shorter distance from the sources. ESM does not have such an infrastructure. Finally, the PIM-SM and session management mechanism build their distribution tree on the shortest path to the root according to TCP/IP routing information, but ESM builds its distribution tree by randomly choosing neighbors from the member list. This difference will remarkably affect the efficiency of the session topologies.

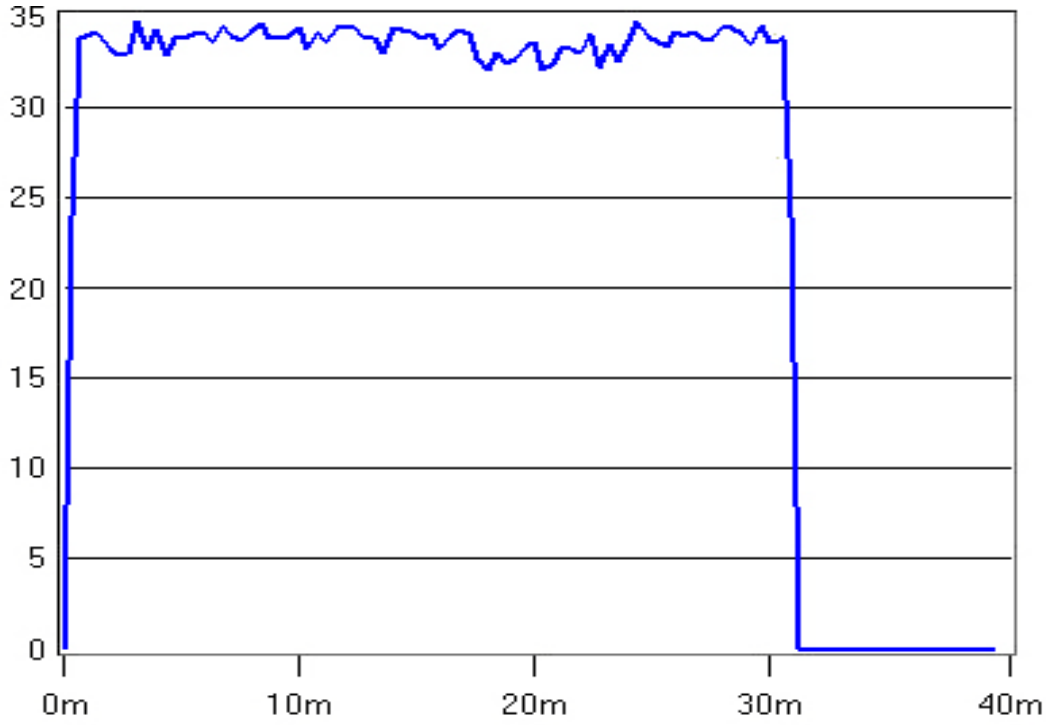



**Figure 33 PIM-SM Total Data Load (Packets/Sec)**

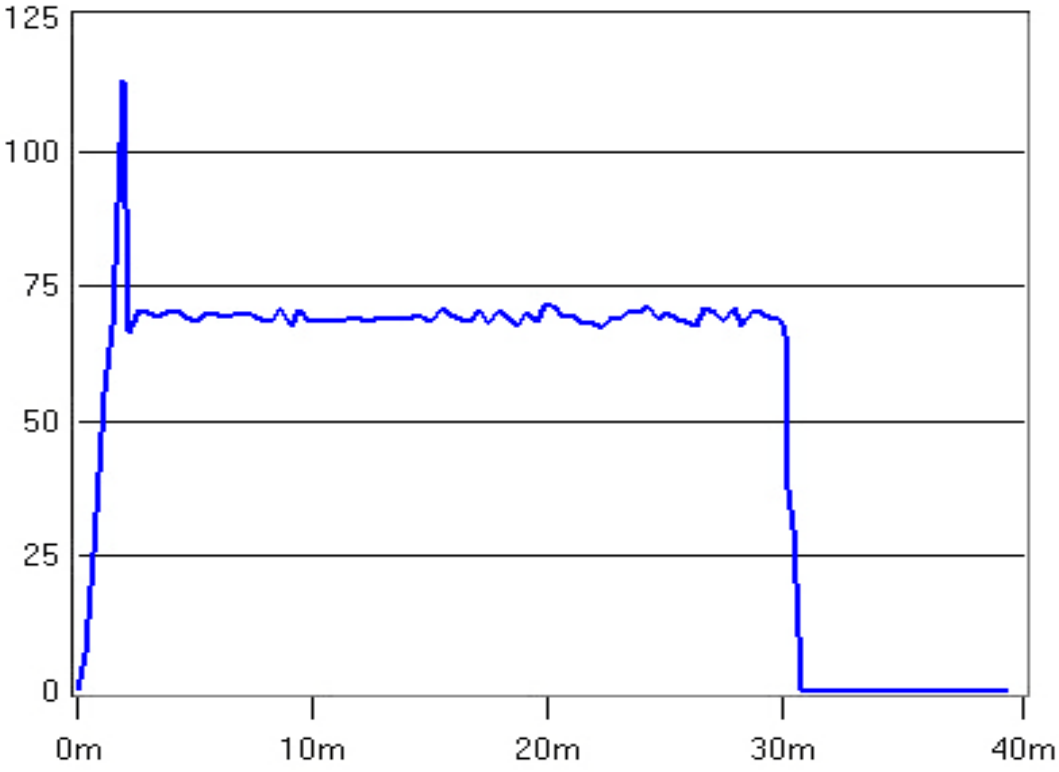



**Figure 34 ESM Total Data Load (Packets/Sec)**

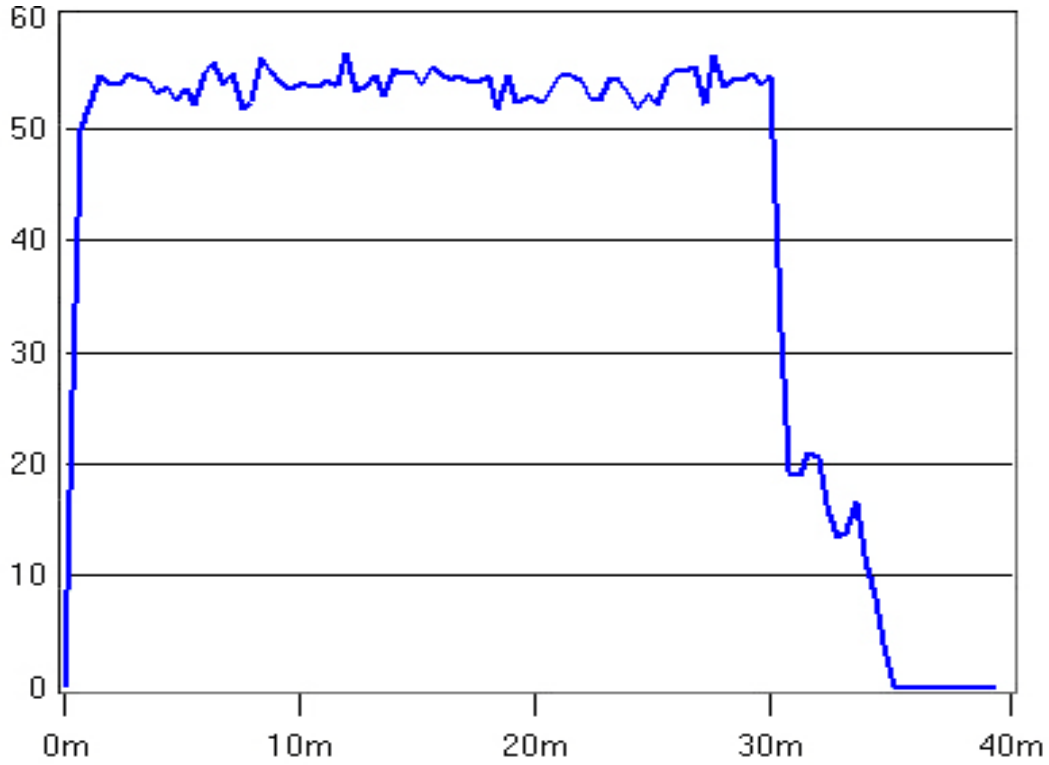



**Figure 35 Session Management Mechanism Total Data Load (Packets/Sec)**

The above three diagrams show the effective data load in each domain. The data load represents the total number of data packets received at a specific time collected at all nodes in a domain, and the retransmitted data packets are also counted in the data load. In these diagrams, we can observe the data transfer volume in each domain for the same task and the trend of data transfer in a session. The y axis represents the number of packets per second, and the x axis is the elapsed simulation time in minutes. Because we used a fixed data packet size (1 Kbytes/packet) in our simulation, the above data load (packets/second) can also be used to calculate the data load measured in bits/second.

Basically, the data load in each technique is relatively stable. For PIM-SM, the data load maintains between 30 and 35 packets per second. For ESM, the data load ranges from over 50 to about 75 packets / second when most receiver nodes join the group. For our session management mechanism, the data load keeps at 50–60 packets per second except for the end of a session. A reason of the fact that the data loads of our session management mechanism is as high as the data load of ESM is that we collect the data packet received by the Mesh nodes and local group controllers, which contributed about one third in the statistics of data load, as well as the receiver nodes in our session management mechanism. If only counting the data packets received by receiver nodes, the data load of our solution will be lower than the data load of ESM. Because of some programming difficulties, we did not count the data packets received by routers and gateways in PIM-SM. Therefore, PIM-SM has a very low data load. Another reason for the low PIM-SM data load in our statistics is that PIM-SM does not have any reliability function that requires data retransmissions to recover lost packets.

We can conclude from the above results that PIM-SM and our session simulation will need fewer data packets to fulfill the same tasks, but ESM

will need more data packets to do the same job. The differences among data transfers are mostly due to the different techniques used in topology establishment and reliability. In ESM, nodes randomly choose neighbors and periodically optimize the topology, so the topology is inefficient and keeps changing in a session. Therefore, there are many mistakes and redundant data transfers in a session in ESM, and more retransmissions are required by receiver nodes. In our session management mechanism, the topology is very stable because of the support of the Mesh as infrastructure, and the effective management in local groups will provide more powerful local recovery for lost packets.

The above results reveal that our session management mechanism can meet the requirement of data delivery. The data packet delay, time average, and data load for our session management mechanism shows that our solution can efficiently save network resources for data packet transfers. The data forwarding module of our session management mechanism can effectively forward data packet from upstream nodes to downstream nodes. The session topology auto-configuration module can establish an efficient topology to support the data forwarding. The stable time average of data packet delay also shows that the session topology auto-configuration module has a stable

performance when membership and network changes. The flow control module also contributes to the successful session transfer.

Next, we will compare some other aspects of our session management mechanism and ESM. Most of the comparisons are under similar conditions. An example is the same control flow schemes used by both two techniques, which can significantly affect many other things, e.g., control packets for requesting retransmissions.

The first things we will compare are the time for receiver nodes to join the distribution topology and membership changes during the simulation. The timeout values of a join request sent from a receiver node are the same for both techniques, so the receivers will resend their join requests at a similar frequency. As we have introduced above, in both techniques, all intermediate nodes on the distribution topology will accept the same number of child nodes. Therefore, the distribution trees will have the same height and are automatically built under the same conditions.

The following diagrams for joining times show the mean of receiver join times during the simulation in ESM and our solution. The y axes in figure 36

and 38 represent the time needed by a node to join the session, measured in seconds, and the x axes are the elapsed simulation time. The y axes in figure 37 and 39 indicate the total number of nodes in the session at a specific time, and x axes show the elapsed simulation time.

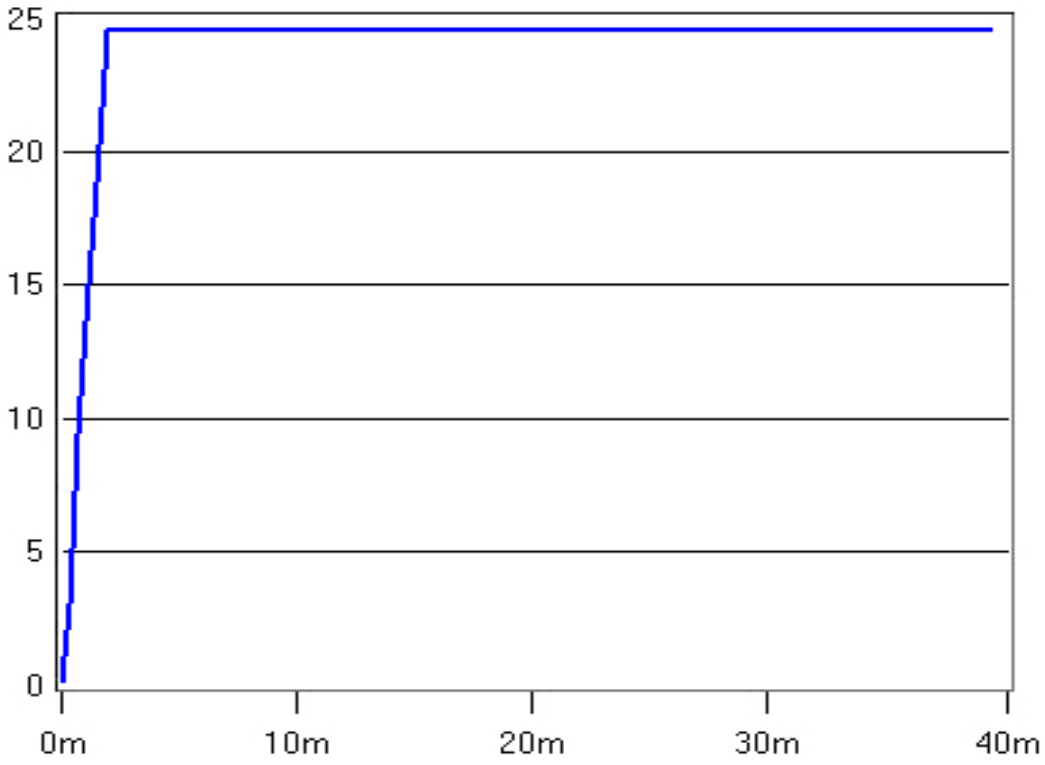



**Figure 36 ESM Receiver Node Joining Time (Seconds)**

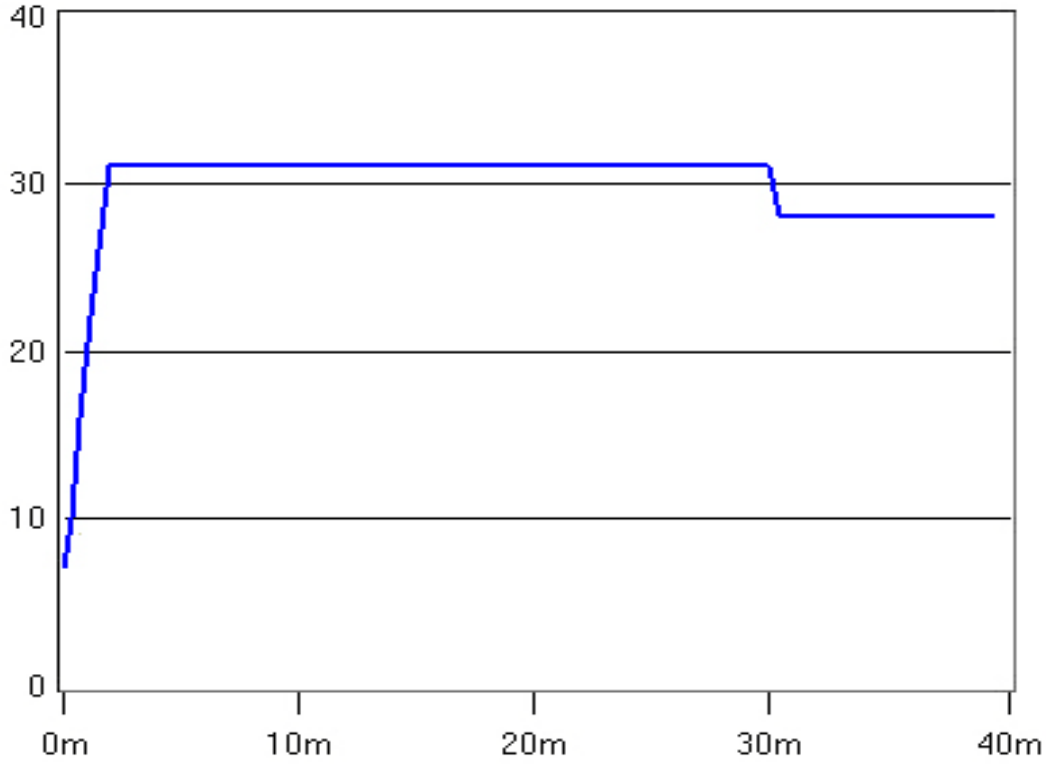



**Figure 37 Member Number of ESM (Number of Hosts)**

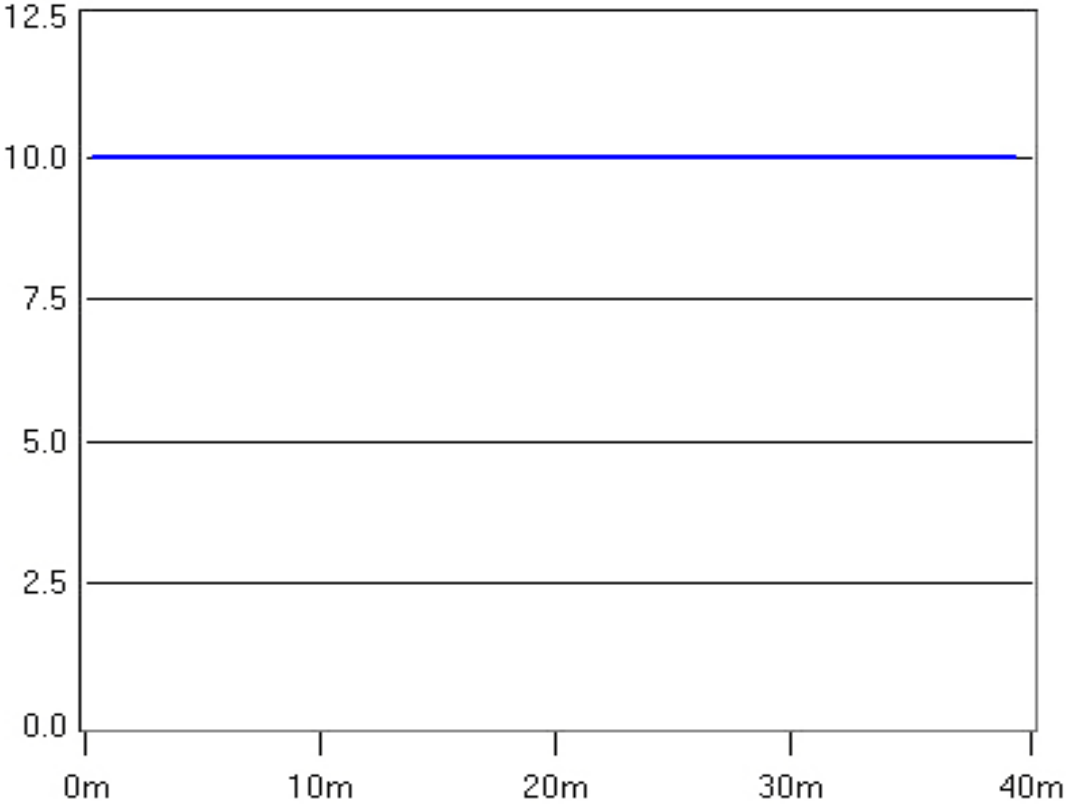



**Figure 38 Session Management Mechanism Receiver Node Joining Time (Seconds)**

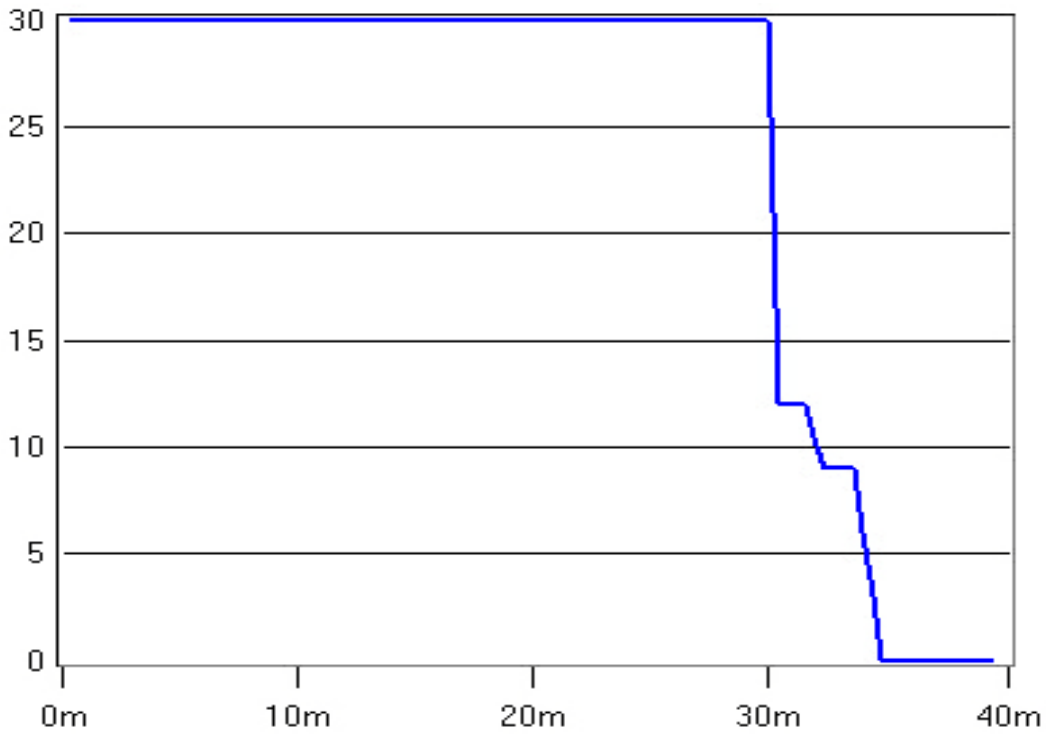



**Figure 39 Member Number in Session Management Mechanism (Number of Hosts)**

In ESM, receiver nodes will need a longer time to find proper parent nodes, and the join time will increase up to over 25 seconds, shown in figure 36, as the more and more receiver nodes join the group, as shown in figure 37. In our session management mechanism, the join time is almost fixed, about 10 seconds, which is shown in a blue line overlapped with the horizontal grid line of 10.0, shown in figure 38.

The main cause for the differences of the receiver node joining times is the differences of joining algorithms used in ESM and our session management mechanism. Each ESM receiver node randomly chooses nodes from a

received node list, sends join requests to these nodes, waits for responses, and reselects nodes and resends join requests if receiving no response. The average joining time will increase, when the session topology gets complicated as more and more receivers join the group. In our session management mechanism, a local service node just needs to join the topology only once when the first downstream node issues the join requests to the local service node, and other receiver nodes only need to bind to the local service node.

The joining time is important for receivers to access multicast services, especially for real time applications, and to recover from failed connections and service interruptions. In our solution, by the support of the session management, the session information is available on the mesh and the local group can obtain the session information easily. To access multiple sessions, a local group will only need to join the topology once. Therefore, our session management mechanism can support multiple multicast sessions with the lowest overhead and the shortest join time. ESM and any other multicast techniques have not such an advantage.

As we can see in the diagram 39 of membership changes, the receivers in

our session management can join the topology at the beginning of the session almost at the same time. However, the receivers in ESM have to join gradually as the topology is extended, shown in figure 37.

This difference is due to the support of session management, support of infrastructure, and difference of topology auto-configuration algorithms. The session management will ‘broadcast’ the session information on the infrastructure, the Mesh, by the session announcement mechanism. With the support of the session announcement and the Mesh, the session information has been placed much closer to the local groups than in any other multicast techniques, which can greatly shorten the join time. In our topology auto-configuration algorithm, a receiver node will first join a local group, and the local group will try to find the shortest path to the root. All these techniques will guarantee that a node will easily find the nearest parent node, and the parent node will easily access session information and accept a child node as fast as possible. In ESM, receiver nodes have to repeat the time-consuming probe-and-wait process to find a proper parent node.

The above comparison and discussion can quantitatively show that our session management mechanism can effectively meet the group management

requirements of dealing with member join and leave, which is an important design goal of our session topology auto-configuration module.

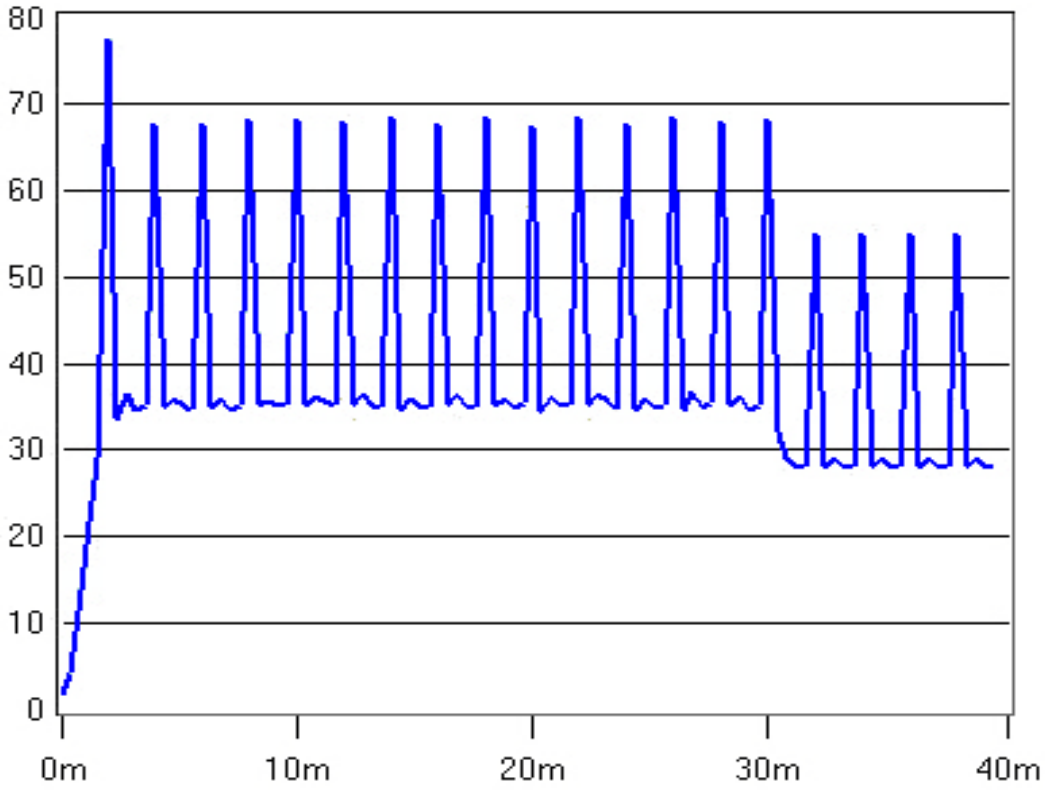



**Figure 40 ESM Total Control Packet Rate (Packets/Second)**

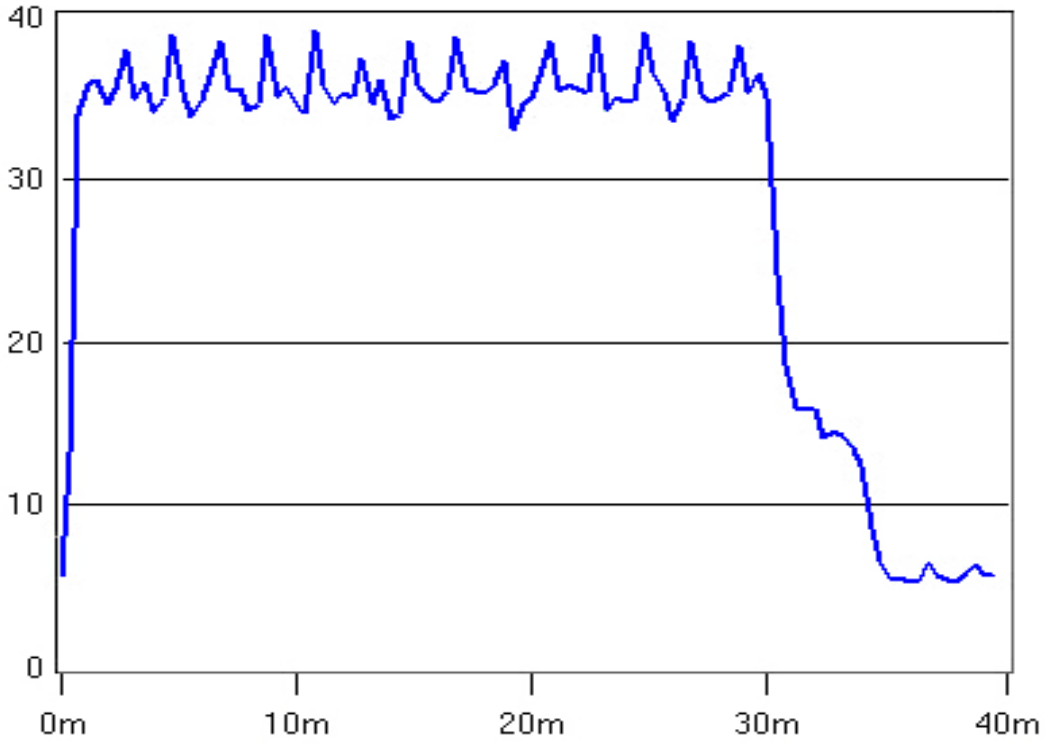



**Figure 41 Session Management Mechanism Total Control Packet Rate (Packets/Second)**

Figures 40 and 41 show the total control packet rate of two techniques. The control packet rate represents the total control packet received by all nodes in each domain at a specific time of simulation. The control packet rate indicates the network overheads to maintain a session in a multicast technique. As we discussed above, the simulation environments and conditions for each multicast techniques are similar. To compare the control packet rates, we use the same timeout value that is used for monitoring and resending all kinds of control packet, and we use the same time intervals that are used for periodically exchanging information with neighbors, e.g., heartbeat packets in our session management mechanism and periodic

membership information exchanging packet in ESM. In figure 40 and 41, the y axes represent the number of control packets per second, and the x axes are the elapsed simulation time.

In the figure 40 and 41, we can see that the ESM needs many more control packets than our session management mechanism needs. The peak values of the control packet rate in ESM are almost twice as high as our session management's peak values. The different control packet overheads can be explained by different session management in two techniques. With the support of session management and hierarchical topology, the membership management and distribution topology in our session management mechanism are more smooth and stable than other techniques. On the contrary, the topology and membership in ESM change more frequently than our session management mechanism. As a result, ESM needs more control packets to maintain and optimize its topology, to keep track of membership, etc. In our solution, with support of session management and hierarchical topology, the probability of recovering lost packets by local groups is much higher. ESM needs more control packets to request lost packets from upstream nodes.

In figure 40 and 41, the result curves show some periodic changes. In our simulation, although the membership changes are supposed to be random configuration due to random changes in network changes, the simulation somehow shows some nearly periodical changes in network changes that trigger the auto-configuration algorithms in ESM and our solution works almost periodically. The causes of this result may come from simulation platform or configuration in our simulation.

The control packet overhead results show that our session management mechanism can effectively control all the phases in the multicast session, except for AAA and key management, which are not covered in this simulation. It shows that our design of session management mechanism in chapter 5 is successful and efficacious. Because of the support of multicast session management, each phase of life cycle model will need a smaller number of control packets. For example, receiver joining process will generate sender control packets in the session topology auto-configuration phase because the session information is located in the nearest service nodes and each local group will need to join the topology once in our design. The flow control can also benefit from the session management mechanism because lost packet retransmission can be done by a near upstream service

node in our design.

In conclusion, according to the results of our simulation, we can find that our session management mechanism has many advantages. The topology auto-configuration algorithm can effectively establish the hierarchical topology and connect domains running different multicast protocols. With the techniques designed to support multiple multicast protocols, the data transfers between domains flow smoothly. The data translation, data forwarding, collaboration of flow control have been proven feasible and efficient. The session management mechanism can effectively control all phases of a multicast session, from the session creation to the session termination. Our session management mechanism is a fast multicast technique, whose speed is close to the IP multicast in network layer and much faster than overlay multicast. It is also an efficient and effective multicast technique that can significantly lighten network bandwidth burdens, whose data load and control packet rate are low. With the session management support, the nodes in our session management mechanism can join the multicast group faster, get stable data transfer, and recover lost packets easily.

To help readers better recall the details and methodology in the project, we summarize the mapping among life cycle phases, requirements, design, and simulations in table 8. In our project, we first summarized the requirements for multicast technology in commercial use and the life cycle model of multicast session. We use a subset of the requirements and life cycle phases as the starting point for this project, and designed a multicast session management mechanism, which can satisfy the subset of requirements. We create the simulation plan that qualitatively and quantitatively checks and validates the requirement satisfaction of our design.

| Life Cycle Phase | Requirements | Design | Simulation |
|---|---|---|---|
| Session announcement and session creation/termination | Group management (dynamically and automatically create/terminate a group) | Session Management Mechanism | Feasibility of topology auto-configuration and session management |
| Session topology auto-configuration | Group management (member join/leave) | Hierarchical topology auto-configuration, Session Management Mechanism | Feasibility of topology auto-configuration and session management, performance comparison of join time |
| | Scalability (large number of receivers, large coverage, inter-domain capability, collaboration with different distribution techniques, support for multiple groups) | Hierarchical topology auto-configuration, Session Management Mechanism | Feasibility of our session management |
| Data forwarding | Data delivery, Scalability (collaboration with different distribution techniques ) | Support for different multicast protocols, Session Management Mechanism | Feasibility of connecting heterogeneous multicast techniques, and performance comparison of data delivery |
| Flow control | Reliability | Session Management Mechanism | Feasibility of our session management |
| Almost every phase | Deployment (works with different underlying hardware and software, and ease of deployment, i.e., incremental deployment) | Session Management Mechanism | Feasibility of transfer data across the domains with different protocol models (PIM-SM and ESM), across various hardware, and even across domains with or without multicast support |

**Table 8 Mapping among Life Cycle Phases, Requirements, Design, and Simulation**

# 7 Conclusion and Future Work

In this chapter, a concluding overview of the Session Management Mechanism is given, summing up its technical and theoretical background. Furthermore, an outlook for future development is provided.

## 7.1 Summary

To provide multicast communication on the Internet, many multicast technologies have been developed in the network layer, e.g., DVRMP and PIM-SM, in the transport layer, e.g., LGMP and RMTP, and in the application layer, e.g., ESM. Because most current technologies cannot satisfy the requirements for multicast in commercial usage, multicast technologies have not been accepted by most ISPs.

In this paper, we summarize the requirements for multicast technologies, including data delivery, scalability, security, group management, reliability, and deployment. In order to understand and meet the requirements, we define a life cycle model that most multicast sessions should follow, from the creation of a session to the termination of a session.

Based on the requirements and the life cycle model we defined, we propose

and design a general solution that can control each phase of a session and satisfy most requirements for multicast technology. This general solution has three parts: hierarchical topology auto-configuration algorithm, Session Management Mechanism, and techniques supporting different multicast protocols.

This proposed general solution is based on a hierarchical topology auto-configuration algorithm, which can automatically establish a hierarchical topology derived from a pre-deployed Mesh and self-organized local groups for multicast sessions.

The Session Management Mechanism is the kernel of our solution. It has a three-layer structure, can be placed on all nodes, and can control every phase of the multicast session life cycle. The Session Management Mechanism can be extended to an excellent infrastructure for supporting different multicast techniques on the Internet.

To coordinate heterogeneous multicast protocols, we propose and design techniques for supporting different multicast protocols based on the topology generated by our hierarchical topology auto-configuration algorithm. Our

solution has three aspects of mapping between different multicast techniques: topology mapping, packet translation, and functionality mapping.

To verify the feasibility of our solution and compare its performance with other multicast techniques, we simulate our solution and compare it with PIM-SM and ESM. The simulation is implemented in Opnet Modeler, which is a commercial tool for network protocol design and simulation. The results of our simulation indicate that our Session Management Mechanism is a good solution for multicasting on the Internet and has excellent performance.

## 7.2 Future Work

Because our simulation is limited by research resources and time, some important requirements and life cycle phases have to be ignored, including security and AAA. To fully verify and validation our design, a more comprehensive and complete simulation should be done. This simulation should cover all phases of life cycle model and some new techniques, e.g., FEC scheme proposed by IETF RMT working group. For example, in our simulation, we only create a session for reliable data transfer that will result in some latency for retransmitting lost data. For live stream, the data latency and performance of our solution could be a little different. New simulations should cover such scenarios.

Our Session Management Mechanism provides a powerful platform that can be used for managing multicast services on the Internet. Therefore, the most important future work is to build a real commercial product, test it with real multicast streams, and deploy it on the Internet.

The research community has already realized that supporting different multicast protocols is an essential task for future development of multicast technology. To build a solution for more general usage, we can standardize the modules, the interfaces between our solution and other multicast techniques, and the procedure of connection between different techniques.

The life cycle module targets a certain class of multicast technology. There may be other classes of multicasting that should be considered by the research community. Their requirements and life cycle models may be significantly different from ours. Investigation of them may help improving our session management mechanism.

Because our solution currently is built and tested in IPv4 systems, some design efforts will be needed when we build our session management mechanism in IPv6 systems. For instance, new addressing technique, MLD

modules, and connecting IPv4 and IPv6 systems are critical in new design.

## References

[1] S. Paul, “Multicasting: Empowering the Next-Generation Internet”, IEEE Network, January / February 2000, Vol. 14, No. 1.

[2] Multicast Deployment Made Easy, IP Multicast Planning and Deployment Guide, CISCO, http://www.cisco.com/warp/public/cc/techno/tity/ipmu/tech/ipcas_dg.pdf

[3] “Multicast Status Page”, AmericaFree.TV, http://www.multicasttech.com/status/, 2007

[4] S. Kumar, “Why multicast is irrelevant to the Internet”, http://www.arl.wustl.edu/~jst/reInventTheNet/?p=161, Dec. 13, 2006

[5] K. C. Almeroth, “The Evolution of Multicast”, IEEE Network, January / February 2000, Vol. 14, No. 1.

[6] S. Deering and D. Cheriton, “Multicast Routing in Datagram Internetworks and Extended LANs”, ACM Trans. Comp. Sys., May 1990.

[7] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM)”, IETF RFC 4601, Internet Engineering Task Force, 2006

[8] T. Bates, R. Chandra, D. Katz, Y. Rekhter, “Multiprotocol Extensions for BGP-4”, IETF RFC 2283, Internet Engineering Task Force, 1998.

[9] B. Fenner, D. Meyer, “Multicast Source Discovery Protocol (MSDP)” IETF RFC 3618, Internet Engineering Task Force, 2003.

[10] S. Bhattacharyya, Ed, “An Overview of Source-Specific Multicast (SSM)”, IETF RFC 3569, Internet Engineering Task Force, 2003

[11] B. Adamson, C. Bormann, M. Handley, J. Macker, “NACK-Oriented Reliable Multicast (NORM) Building Blocks”, Internet Draft, Internet Engineering Task Force, 2003.

[12] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft, “Forward Error Correction (FEC) Building Block”, IETF RFC 3452, Internet Engineering Task Force, 2002.

[13] M. Hofmann, M. Rohrmuller, “Impact of Virtual Group Structure on Multicast Performance”, 1997. Lisboa, Portugal, Ed.: A. Danthine, C. Diot: From Multimedia Services to Network Services, Lecture Notes in Computer Science, No. 1356, Page 165-180, Springer Verlag, 1997.

[14] B. Whetten and J. Conlan, “A Rate Based Congestion Control Scheme for Reliable Multicast,” GlobalCast Commun. Tech. White paper, Nov. 1998.

[15] Internet Multicast Addresses, Internet Assigned Number Authority (IANA), http://www.iana.org/assignments/multicast-addresses, February. 2006.

[16] D. Thaler, M. Handley, D. Estrin, “The Internet Multicast Address Allocation Architecture”, IETF RFC 2908, Internet Engineering Task Force, 2000.

[17] S. Hanna, B. Patel, and M. Shah, “Multicast Address Dynamic Client Allocation Protocol (MADCAP)”, IETF RFC 2730, Internet Engineering Task Force, 1999.

[18] D. Estrin, R. Govindan, M. Handley, S. Kumar, P. Radoslavov, and D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", IETF RFC 2909, Internet Engineering Task Force, 2000.

[19] S. Fahmy and M. Kwon, “Characterizing Overlay Multicast Networks”, Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP’03), 2003.

[20] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O’Toole, Jr., “Overcast: Reliable Multicasting with an Overlay Network”, Proceeding of USENIX Symp. on Operation Systems Design and Implementation, 2000

[21] J. Wu, I. Stojmenovic, “Guest Editors' Introduction: Ad Hoc Networks”, IEEE Computer, Vol. 37, No. 2, Page 29-31, February 2004.

[22] C. Gui and P. Mohapatra, “Efficient Overlay Multicast for Mobile Ad Hoc Networks”, Wireless Communications and Networking Conference (WCNC), 2003

[23] K. Chen and K. Nahrstedt, “Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET,” Proc. 21st

Ann. Joint Conf. IEEE Computer and Comm. Societies, vol. 3, IEEE Press, 2002, pp. 1180-1189.

[24] A. Patil, Y. Liu, L. M. Ni, L. Xiao, A.-H. Esfahanian, “POMA: Prioritized Overlay Multicast in Ad-Hoc Environments”, IEEE Computer, Vol. 37, No. 2, Page 67-74, February 2004.

[25] K. Andreev, B. M. Maggsy, A. Meyersonz, R. K. Sitaraman, “Designing Overlay Multicast Networks For Streaming”, ACM Symposium on Parallel Algorithms and Architectures, Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, Page 149-158, 2003.

[26] Y.Chu, S. G. Rao, S. Seshan, H. Zhang, “A Case for End System Multicast”, Proceedings of ACM Sigmetrics, June 2000

[27] Y.Chu, J. Chuang, H. Zhang, “A Case for Taxation in Peer-to-Peer Streaming Broadcast”, ACM SIGCOMM Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems (PINS), Portland, OR, August, 2004

[28] H. Deshpande, M. Bawa, H. Garcia-Molina,“Streaming Live Media over Peers”, Stanford InfoLab Publication Server.

[29] S. Banerjee, S. Lee, B. Bhattacharjee, A. Srinivasan, “Resilient multicast using overlays”, Joint International Conference on Measurement

and Modeling of Computer Systems, Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Vol. 14, No. 2, Page 237-248, April 2006.

[30] D. Pendarakis, S. Shi, D. Verma, M. Waldvogel, “ALMI: An application level multicast infrastructure”, in Proceeding of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS ’01), 2001

[31] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, J. W. O’Toole, Jr., “Overcast: Reliable Multicasting with an Overlay Network”, Proceeding of USENIX Symp. on Operation Systems Design and Implementation, 2000.

[32] Y. Chawathe, S. McCanne, and E. A. Brewer, “RMX: Reliable Multicast for Heterogeneous Networks”, Proceeding of IEEE INFOCOM, 2000

[33] G. K. John, W. Byers, “ROMA: Reliable Overlay Multicast with Loosely Coupled TCP Connections”, Proc. of IEEE INFOCOM '04

[34] B. Quinn, K. Almeroth, “IP Multicast Applications: Challenges and Solutions” IETF RFC 3170, Internet Engineering Task Force, September 2001.

[35] S. J. Koh, J. Park, E. Kim and S. G. Kang, “Transport Session Management for Multicast Transport Protocol”, ICAST 2001

[36] R. Vida, L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6”, IETF RFC 3810, Internet Engineering Task Force, 2004.

[37] I. Brown, J. Crowcroft, M. Handley, B. Cain, “Internet Multicast Tomorrow”, The Internet Protocol Journal (IPJ), Vol 5, No. 4, December 2002

[38] M. Handley, C. Perkins, E. Whelan, “Session Announcement Protocol”, IETF RFC 2974, Internet Engineering Task Force, October 2000.

[39] M. Kadansky, D. M. Chiu, B. Whetten, G. Taskale, B. N. Levine, Seok J. Koh, “Reliable Multicast Transport Building Block: Tree Auto-Configuration”, IETF Internet-Draft, November 18, 2002. http://www.ietf.org/internet-drafts/.

[40] T. Wang and J. W. Atwood, "Hierarchical Topology for Multicasting", in Proceedings of The IASTED Conference on Computer Systems and Applications (CSA 2004), Banff, Alberta, Canada, 2004 July 8--10, pp. 104--108.

[41] D. Thaler, M. Talwar, A. Aggarwal, L. Vicisano, T. Pusateri, “Automatic IP Multicast Without Explicit Tunnels (AMT)”, IETF Internet-Draft, October 3, 2007, http://www.ietf.org/internet-drafts/.

[42] S. Islam, J. W. Atwood, “Sender Access Control in IP Multicast”, The 33rd IEEE Conference on Local Computer Networks (LCN), Montreal, Canada, 20-23 October 2008.

[43] Ritesh Mukherjee, “Secure Group Communication”, Ph.D. Thesis, Department of Computer Science and Software Engineering, Concordia University, September 2005.

[44] Salekul Islam, “Participant Access Control in IP Multicasting”, Ph.D. Thesis, Department of Computer Science and Software Engineering, Concordia University, July 2008.