A NEW SERVICE ARCHITECTURE FOR IPTV OVER INTERNET

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERVE ÖZKARDEŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2013

Approval of the thesis:

A NEW SERVICE ARCHITECTURE FOR IPTV OVER INTERNET

submitted by **MERVE ÖZKARDEŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** ________________

Assoc. Prof. Dr. Şenan Ece Schmidt
Supervisor, **Electrical and Electronics Engineering Dept., METU** ________________

Examining Committee Members:

Prof. Dr. Semih Bilgen
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Şenan Ece Schmidt
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Gözde Akar
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Cüneyt Bazlamaçcı
Electrical and Electronics Engineering Dept., METU ________________

M.Sc. Soner Çınar
ASELSAN Inc. ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MERVE ÖZKARDEŞ

Signature :

ABSTRACT

A NEW SERVICE ARCHITECTURE FOR IPTV OVER INTERNET

Özkardeş, Merve
M.Sc., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Şenan Ece Schmidt

January 2013, 49 pages

Multimedia applications over the Internet and Internet Protocol Television (IPTV) gain a lot of attention. IPTV has a number of service requirements such as; high bandwidth, scalability, minimum delay, jitter and channel switch time. IP multicast, IMS (IP Multimedia System) Protocol and peer-to-peer approaches are proposed for implementing IPTV. However, IP multicast requires all the routers in the core network to possess multicast capability, IMS does not easily scale and P2P cannot efficiently utilize the network resources because of its completely distributed nature. To this end, we propose new application layer multicast protocol Cluster Based Application Layer Multicast IPTV (CALMTV) which combines application layer multicast, scalable video coding and probing techniques to meet IPTV requirements. We present the components and their relevant algorithms and evaluate the performance of CALMTV with ns2 simulations. Our results compared with the published results of other IPTV architectures show that CALMTV has better performance in end-to-end delay and zapping time.

Keywords: IPTV, Application Layer Multicast, Scalable Video Coding

ÖZ

İNTERNET ÜZERİNDE IPTV İÇİN YENİ BİR MİMARİ

Özkardeş, Merve
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç Dr. Şenan Ece Schmidt

Ocak 2013, 49 sayfa

Son yıllarda Internet üzerinden gönderilen multimedia uygulamaları ve Internet Protokol Televizyonu (IPTV) giderek daha çok ilgi görmeye başlamıştır. IPTV'nin yüksek bant genişliği, ölçeklenebilirlik, minimum gecikme, jitter ve kanal geçiş süresi; gibi kendine özgü gereksinimleri vardır. IP-multicast, IMS (IP Multimedia Sistemi) Protokolü ve P2P yaklaşımlar IPTV uygulanması için önerilmiştir. Ancak, IP-multicast çekirdek ağdaki tüm yönlendiricilerin çok noktaya yayın yeteneğine sahip olmasını gerektirir, IMS'in ölçeklenebilirlik sorunları vardır ve P2P tamamen dağınık yapısı nedeniyle ağ kaynaklarından verimli bir şekilde yararlanamaz. Bu amaçla, yeni bir uygulama katmanı çok noktaya yayın protokolünü (CALMTV) öneriyoruz. Bu protokolde daha etkili sonuçlar elde etmek için çok noktaya gönderim, bant genişliği tahmin etme ve ölçeklenebilir video tekniklerini birleştirdik. CALMTV performansını ns-2 kullanarak benzetimledik. Sonuçlarımıza göre CALMTV uçtan uca gecikme ve zap zamanı açısından diğer IPTV yapılarına göre daha iyi sonuçlar vermektedir.

Anahtar Kelimeler: IPTV, Ölçeklenebilir Video Kodlaması, Uygulama Seviyesinde Birden Çok Kullanıcıya Gönderim

To My Family

ACKNOWLEDGMENTS

I would like to express my special thanks to my supervisor Assoc. Prof. Dr. Şenan Ece Schmidt. My special thanks go to TÜBİTAK. I would like to thank to my employer, ASELSAN. Finally, I would like to thank my parents and friends who supported me throughout the whole time.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

ALM	Application Layer Multicast
CALMTV	Cluster Based ALM IPTV Architecture
CP	Content Provider
DRM	Digital Right Management
FMC	Fixed Mobile Convergence
GT-ITM	Georgia Tech Internetwork Topology Model
HD	High Definition
IGMP	Internet Group Management Protocol
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPTV	Internet Protocol Television
ITU	Internal Telecommunication Union
MPEG	Moving Picture Experts Group
P2P	Peer-to-Peer
PLR	Packet Loss Ratio
PVR	Personal Video Recording
QoE	Quality of Experience
QoS	Quality of Service
RDP	Relative Delay Penalty
RTP	Real-Time Transport Protocol
SD	Standard Definition
SIP	Session Initiation Protocol
SLoPS	Self-Loading Periodic Streams
SP	Service Provider
SVC	Scalable Video Coding
TSTV	Time-Shifted TV
VoD	Video On Demand

CHAPTER 1

INTRODUCTION

In recent years with the help of successful compression techniques and new broadband access networks, the popularity of video transmission over the Internet has grown very fast. Internet Protocol Television (IPTV), which consists of these modern technologies and delivers media content through the network, began to gain more attention from the end users. Currently, IPTV market continues to grow in the sense of both customer size and investments [29]. According to Info Research Group predictions, U.S. IPTV market will grow to 15.5 million in 2013 [1]. ABI Research has reported that IPTV will expand and is expected to have 79 million subscribers by the year 2014 [29]. ABI research shows IPTV is evolving not only in the countries which have the high-speed Internet but also in developing countries, too. The improvements in the network technology and scalable video coding techniques make IPTV as the next killer application over the Internet [33].

IPTV has some specific service requirements in order to race against the cable and satellite TV.

Low transmission delay without reducing the system throughput is crucial in IPTV systems [50], [16]. Transmission delay should be under 2 seconds [50] in order to avoid freezing in a live broadcast. Channel zapping time is a very important metric when more than one video channel are transmitted. When end users change the active channel, they want to watch next channel immediately. Agilent Technologies have reported that the channel zapping time should be under 0.43 seconds to ensure end user requirements [22, 21]. Scalability is another critical issue for IPTV implementations because of the highly dynamic network state and subscribers' expectations about the broadcast [50]. IPTV requires a large amount of bandwidth to transport the media data source to destination. Video transmission data are huge compared to other network services such as mail, surfing; and demands huge bandwidth. For an HD video practically 6 Mbps should be supported [50].

There is a number of different approaches for IPTV Network structure in the literature:

- IP Multicast: It [55] sends a packet to multiple receivers. It sends packets to a group of receivers only once. Since it uses router multicast capability, it has deployment problems.
- Peer-To-Peer Streaming: In a P2P IPTV system, peers to supply their bandwidth in order to send the live video. It has extreme advantages of scalability, development and deployment. However, it has some unsolved problems about delay, jitter guarantee.
- IP Multimedia Subsystem (IMS): IMS is a standardized infrastructure and stands on a horizontal control layer which isolates service from the access network. In IMS-based IPTV structures, signalling is an important problem.

In this thesis, we propose Cluster Based Application Layer Multicast IPTV (CALMTV) which is a hybrid solution; it combines probing (bandwidth estimation), ALM (Application Layer Multicast), P2P (Peer-to-Peer), and Scalable Video Coding (Scalable Video Coding).

The novel features of our structure are as follows:

- Different than the previous works, we combined probing, ALM and SVC ideas.
- We proposed a new “ALM Clustering Algorithm” and a“Video Compression Control Algorithm” in order to ensure IPTV service requirements.
- Our solution utilizes the estimation of the available network bandwidt, so there is no need to learn network topology or leased line information.

The remainder of this thesis is organized as follows. Chapter 2 expresses IPTV overview in the sense of QoS requirements, previous proposals, and related technologies. Chapter 3 presents our proposal for IPTV. In that chapter we explain all components of our software blocks, their usage and necessity in detail. In chapter 4 overall performance evaluations can be found as well as our assumptions and traffic models. Chapter 5 states our conclusions and the future direction of our research.

CHAPTER 2

LITERATURE OVERVIEW

Internet Protocol Television (IPTV) represents the delivery of television content to end users over Internet technology. It differs from normal broadcast TV. IPTV can provide different content to different users. The two main functions of IPTV are; live TV support and video-on-demand service support. It makes it possible to log user behaviors and provides different commercials for different users. In the recent years, the media services over the Internet and IPTV grow rapidly, and this trend continues as seen in Fig.2.1.

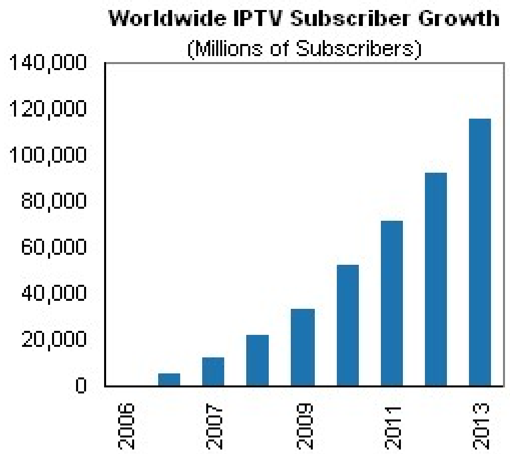


Figure 2.1: IPTV Progression [1]

However, it is not an easy task to manage IPTV. Because the Internet was not designed to provide and support multimedia applications; it represents a shared medium and in order to deliver media applications; it offers a best-effort traffic. However, multimedia services require different demands of the Internet. For example, these kinds of services are more sensitive in the sense of delay, jitter, but they are more tolerable to packet loss. They need stable bandwidth, low latency-jitter in order to meet the user desires. To meet with these requirements, there are plenty of different IPTV architectures and solutions in the literature.

Quality of Service (QoS) is basically about metrics: packet delay, loss and jitter. Certainly, packet delay, packet loss, and jitter are critical issues; but QoS is just the first step. Quality of Experience (QoE) is more about what happens on the customer's side. Better utilization of network sources means nothing if a customer to a service is having a problem. QoE is relevant to but differs from QoS; it is a subjective measure of a customer's experience with a service. In IPTV context, zapping time and experienced video quality are considered to be the most important parameters of QoE metrics.

2.1 IPTV Architecture

Internal Telecommunication Union (ITU) specifies and introduces the basic specifications for IPTV [33]. According to their report in the first stage of IPTV; providers have to support four main service types. These types are live TV, Video on Demand (VoD), TSTV (time-shifted TV) and Personal Video Recording (PVR). For the second stage of IPTV, some other extended services/applications may be requested in the future.

Four different kinds of basic functions have to be assigned to achieve IPTV structure; these functions are Customer, Service Provider (SP), Network Operator, and Content Provider (CP). These functions can act separately and independently to satisfy the requirements of IPTV, so the overall structure has to permit the decomposition of the functional parts. Because there is broadband multicast, security is another key issue about IPTV, service security has to be provided to ensure the interests of CPs and SPs.

To satisfy all the requirements above, IPTV architecture can be divided into five main function sets, which are Customer, Service Operation - Management, Content Operation, Media Distribution - Delivery, and System Management - Security. Each function set can be divided into subsets; their relation can be seen in 2.2.

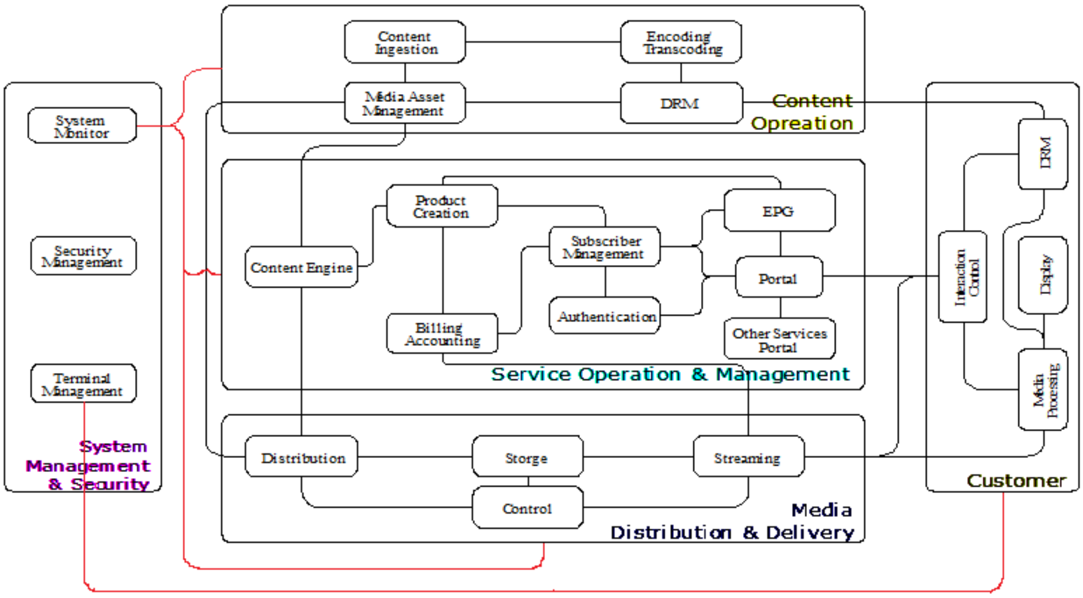


Figure 2.2: IPTV Schema [33]

On the right side of 2.2 there are the end-user functions. These include the hardware and software components such as home gateways, set-top boxes, PC clients, and mobile devices. They allow the subscribers to receive and consume the transmitted video content.

In the middle part of 2.2 content provider operations can be found. In this part, there are provider functions such as, TV channel, music data, encode/decode operations.

On the left side of 2.2 system management operations take place. It is system management's responsibility to handle all delivery processes, multicast operations to deliver video content. Unicast operations have to be handled here, too. These service functions are designed to manage QoS, network sources, packet loss errors.

There are plenty of different solutions to realize IPTV. Some of them are based on next-generation networks. Next-generation networks use a single transportation link to carry whole information and services (data, voice, video, ... etc.) via encapsulating these packets. To achieve NGN structure, some architectural changes in the core and access network must be made by service providers. Basic three options to provide IPTV service suggested by ITU-T [54] are listed below:

- Non-NGN based IPTV solutions: It is possible to make some inter working with NGN but generally a separate service control and application layer were developed specially for IPTV services (IPTV middleware).
- NGN based IPTV architecture: It enables interaction and inter working over specified reference

points between IPTV applications and some existing common NGN components. This approach uses a dedicated IPTV subsystem within NGN to provide all necessary IPTV requirements.

- IMS based IPTV architecture: The IP Multimedia Subsystem (IMS) is an architectural framework to deliver multimedia services over the Internet. Recently, some studies showed that it can be used to provide IPTV service. The details of IMS architecture will be discussed in 2.3.3. It specifies IPTV functions supported by the IMS subsystem and employs these functions to allow the reuse of IMS functions and make service initiation and control based on SIP (Session Initiation Protocol).

To realize the second and third solutions, all network components have to operate in NGN mode. So, second choice becomes unusable in the close future, because of its implementation issues. To achieve first and second proposals, there are certain methods in the literature, which are explained and discussed in 2.3 in detail.

2.2 IPTV Challenges and Performance Metrics

To propose a solid IPTV architecture, IPTV requirements have to be found out. In this section challenges and performance metrics to provide IPTV is discussed in detail.

2.2.1 Network Bandwidth

The bandwidth provided over the access network limits the size of transmitted video streaming data. When subscribers change their preferences about video quality, bandwidth demand raises. If that demand exceeds the maximum capacity of the transmission link, then there are packet losses, which cause screen impairment and result in the subscriber complains.

Scalable Video Coding (SVC) is a widely used video compression standard. With SVC a video can be encoded into 2 Mbps standard definition (SD) stream or 6 Mbps high definition (HD) video stream [50, 17]. Hence, to broadcast any video-stream for three hundred subscribers; 10-15 Gbits of traffic has to be handled in the core network per second. According to these results, some kind of multicast algorithm has to be implemented in the network in order to provide IPTV service.

2.2.2 Packet Loss

IP packet loss can trigger screen impairments, wrong image sequences, or unwanted low resolution and unavailable images. It takes place when some data packets fail to reach the destination along the transmission path. Because of the real-time nature of IPTV, undelivered packets are not supposed to be retransmitted.

To understand the packet loss effects on the subscriber side, we will examine an example. Assume that IPTV delivery uses Scalable-Video-Coding. If missing packets have some information about the reconstruction of the frames then there is a strong possibility that video signal may be lost for a short period of time. If the lost packets are related to only video data, the impact is less intense but still video quality is reduced.

IPTV is very sensitive to packet loss. According to [17, 19], when the packet loss ratio (PLR) goes beyond 0.1 percent, it can result in unacceptable video quality. So, error resiliency is very important in this context. Especially SVC and the other video compression techniques make IPTV much more sensitive to packet loss as the packets of the compressed video carry more information compared to the uncompressed video. When PLR exceeds 0.5 percent, continuous picture problems occur, after that point pictures broke up and freezes become unavoidable.

2.2.3 Jitter

Jitter represents the undesired deviations in the frequency of coming video signal. In another way, jitter is a change in end-to-end latency with respect to time. Eventually, this triggers the packet delay, zapping time and packet loss problem. It has a direct outcome of the decoding processes.

According to [20], the jitter for IPTV architecture should not exceed 40 ms in order to achieve requested video quality. The authors in [17, 19], investigated the video-transmission performance for interactive video or video conference. Their results show that the jitter value should not be greater than 30 ms to achieve the demanded video signal quality.

2.2.4 Packet Delay

IPTV is a real-time process, and it works with the Real-Time Transport Protocol (RTP) which adds timestamps to the transmitted packets in order to synchronize the whole process. In IP networks, the transmission route for packets may not be same. It causes different inter-arrival times, accordingly packets arrive with a different order than the order they are generated. RTP puts every packet in the right place as long as the packet delay does not exceed the size of decoded buffer. If the delay exceeds the buffer's limits, then the intermediary packets are considered as lost and dropped eventually. This causes lost video service for a period of time.

To compete with the cable or satellite TV, IPTV has to support comparable or better performance for customers. For example, an HD video has a transmission delay requirement below 2 seconds [50].

2.2.5 Zap Delay

The channel zapping time is the time required to switch channels. It is defined as, the time between leaving a channel and receiving the first data stream from just joined channel is zapping time.

Researchers in [22, 21] suggest that for best satisfaction of QoE, channel switching time should be under 0.43 seconds.

2.2.6 Scalability

IPTV has very dynamic subscriber size; the architecture has to handle variations of network size very well [50].

2.2.7 Error Resilience

IPTV networks are very sensitive to packet loss and errors. Retransmission may be unavoidable for sometime, so error resilience becomes a key component for any IPTV architecture.

2.2.8 Other Performance Metrics

Security, Digital Right Management (DRM) control, and scheduling data packets are some other critical QoE parameters about IPTV, but they are not in the scope this thesis.

2.3 IPTV Proposals

According to section 2.1, there are three main options to provide IPTV. To construct IPTV architecture, we examined two of the three options. We did not take into consideration NGN-based solutions due to their implementation difficulties. IP Multicast and P2P Streaming are basic non-NGN based solutions in literature. Furthermore, there are plenty of studies about IMS-based IPTV solution; we examined these solutions, too.

2.3.1 IP Multicast

The Internet Protocol (IP) is designed with embedded support for multicast delivery (IP Multicast). IP Multicast lowers the network load by canceling the redundant data transfers; it only replicates data in routers only when necessary [26, 27].

In IP Multicast, the source node sends an IP datagram to a group address which is a unique address in the IP address range that is reserved only for multicast purposes. The datagram is transmitted to receivers, which are interested in receiving related data. So, all the receiving hosts have to subscribe to the multicast group address. In the multicast process, an IP packet is forwarded to all subscribers for the group. To achieve this, IP routers must know that this forwarded data is multicast group data, this information is shared through all routers which have a role in this process.

Routing for a multicast group can be demonstrated as a multicast tree. The root of this tree is the sender for multicast packets. Receivers are the leaves, and routers are the non-leaf nodes in the multicast tree.

In multicast, routers replicate IP Multicast packets to their child nodes. As a result of that, IP routers reduce the total traffic compared to unicast where the sender sends the exact same packets for every subscriber.

In Fig.2.3 the schema of the IP Multicast tree can be found. The datagram is replicated at all the routers on the path of transmission.

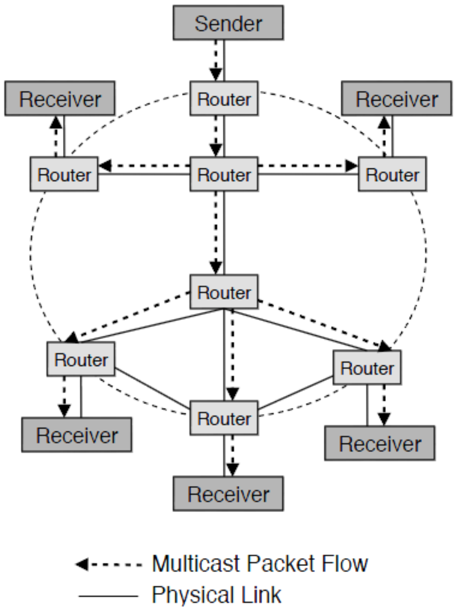


Figure 2.3: IP Multicast Tree / Replication of Datagram.

The source only sends information once, and routers take care of the rest. The redundancy is minimized by this way. However, IP multicast has not been widely used in the network because of the following issues:

- IP Multicast has to be supported by all the routers on the path source to a subscriber.
- Security concerns

When we look at how IP multicast responds to the QoS demands for IPTV:

- Bandwidth: Actually, IP Multicast can realize the IPTV requirement for bandwidth discussed in 2.2.2. Because IP multicast uses the bandwidth more efficiently in the core network, it can carry easily hundreds of channels. One of the successful examples of IP multicast is AT-T U-Verse. AT-T U-Verse [28] can support for four-channel bit rate of 24 Mbps for HD broadcast.
- Packet Loss Ratio: For a payload of 1024 bytes, the PLR starts just below 0.2 percent than decreases and approaches to 0 percent [25]. That result accomplishes the IPTV PLR requirement in 2.2.2.
- Packet Delay: It is reported that AT-T U-Verse [28] can achieve video transmission delay less than 2 seconds.

- Jitter: According to the researchers, the jitter can be eliminated by IP multicast via a playback buffer capable of holding 15 ms of video-stream data [23].

- Zapping Time: Channel switching time can be manageable with IP multicast. It can be decreased to 0.3 sec with some control algorithms [24]

- Error Resilience: If some error occurs in the network like packet loss, then retransmission is needed. Most of the time, IP multicast solves this problem by using redundancy servers (Buffered information held in these servers in case of error). However, these servers have limited capacity, and they can recover up to 1 percent packet loss [28, 50] . According to researchers in [15], error cases are not rare in IP multicast. They claim that, there has to be some error protection and management protocols in IP Multicast.

- Scalability: Scalability is the major problem for IP multicast [50]. Redundancy servers are used against possible errors in IPTV. Because their capacity is limited, more retransmission requests come, and then these become overloaded adversely affecting the network resiliency. Moreover, IP multicast uses IGMP (Internet Group Management Protocol) to handle multicast operations. To provide IPTV service with IP Multicast, all the routers in the network have to support IP multicast and IGMP. On today's internet, there are a few routers, which can manage IP multicast. So, it means there has no chance to provide IPTV using IP multicast soon, unless there is a big alteration in the core network. As a result, IP multicast is an unrealistic proposal when considering the underlying structure.

2.3.1.1 IP Multicast Overall Summary

IP multicast is a very powerful technique to deal with the delay, jitter, and network bandwidth utilization problems of IPTV. Moreover, it has a very good response to zapping time demands. However, in case of error, it becomes very impractical and scalability is a major drawback. To accomplish an IP Multicast solution for IPTV, the underlying network structure has to change, and it is the bottleneck of IP Multicast.

2.3.2 P2P Streaming

In IPTV context, P2P streaming means delivering live video over the Internet. It involves media data, an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content. We will be interested in distribution and delivery part.

A peer-to-peer (P2P) network is a network in which each computer can behave like a provider or subscriber for others. This structure enables several of different data sharing like files, video, audio. P2P networks can be established at home, business or on the Internet.

In P2P systems the peers can change dynamically and hence, in a P2P IPTV system, network size constantly varies. This causes the establishment of new peer relations or some alterations between peers. Peer churn is a collective effect created by departures and arrivals of new peers. Characterizing peer churn is very challenging in practice, because of the large size and dynamic nature of the P2P systems. Hence, in large-scale P2P systems, characteristics of peer churn are not well understood. There are models for peer churn, and some measurement studies show that the effect of the churn is unsteady [9, 10].

In P2P systems, peers offer their bandwidth in order to send the live video. It has significant advantages for scalability, development and deployment. However, its disadvantages for IPTV are transmission delay, jitter and bandwidth. PPLive [42], CoolStreaming [41], PPStream [44] , UUSee [43] are the successful P2P streaming applications.

Now let's analyze the properties of P2P networks in relation to IPTV:

- Bandwidth: To meet the bandwidth demands of IPTV P2P architectures has to reach the certain uplink bandwidth which is limited by the access network bandwidth [50]. In most of the cases, the access network's bandwidth is lower than 5 Mbps. It limits the P2P performance.
- Packet Loss Ratio (PLR): According to researchers, pure P2P network performance is worse than IP multicast in the sense of PLR. In P2P networks, PLR can increase up to 2 percentage [53].
- Packet Delay: Theoretically, P2P architectures can achieve transmission delay under 2 seconds [50], whereas in real-life transmission delays are higher than 2 seconds. There are some cases that delay range changes between 30 seconds to minutes [53].
- Jitter: With some load balance operations and conscious connections between peers, P2P jitter is manageable; which is under 50 ms.
- Zapping Time: When a new channel selected, a starting delay occurs in P2P systems. According to studies in [51] switch to a popular channel takes 10 to 20 seconds; a less popular channel switching time may grow up to 2 minutes. This behaviour can be the downfall of P2P IPTV systems .
- Error Resilience: To cope with the dynamic subscriber behavior P2P systems have some built in functions to support resilience [50].
- Scalability: Because in P2P network peers supply their bandwidth into system, growth in network size is not a big issue. Actually, any increase in the network is a good thing for P2P networks. However, Peer-Churn has some effects on the performance of P2P systems like, old routing tables, inconsistency in stored resources, and inconsistency in overlay topologies [52].

With pure P2P, IPTV cannot be managed. Because there can be thousands of subscribers, to allocate each of them 5 Mbps bandwidth is impossible. There has to be some kind of multicast to solve the bandwidth issue. In literature, there are solid multicast solutions using application layer. Basic two of them is Application Layer Multicast and Overlay Multicast. Now we analyze these proposals.

2.3.2.1 Application Layer Multicast

In IP multicast data transmissions, the source sends to a special destination IP address which indicates a certain receiver group of hosts. The routers are configured accordingly. The main gain is very efficient transmission to preserve bandwidth. However, IP Multicast has a big disadvantage about changing underlying network structure discussed in 2.3.1. One of the proposed solutions to overcome this drawback of IP multicast is Application Layer Multicast (ALM). In ALM, the IP datagrams are replicated at the end users different from IP multicast where datagrams are replicated at the routers. The end users build an overlay network for data transmission. Because ALM replicates packets over the same link (based on overlay topology), it is less efficient than pure IP multicast. In contrast, ALM has no restriction in the sense of service availability and there is no necessity to change underlying network structure.

In Fig.2.4 the schema of ALM structure can be found. The datagram is replicated at end hosts.

ALM has some major advantages over IP Multicast which can be listed as:

- The most important advantage is there is no need for alterations in the network layer infrastructure or allocation of special IP addresses, which is the drawback in IP multicast.
- ALM provides multicast services via IP unicast. Thus, it can use all services' unicast properties, flow control, congestion control, reliability.
- ALM is scalable because it uses P2P streaming to deliver data. As mentioned in sec 2.3.2 in P2P networks, peers supply their bandwidth into the system. Hence, growth in network size is not a big issue for ALM, too.

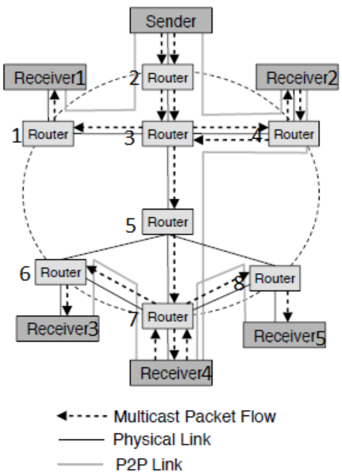


Figure 2.4: ALM over end hosts.

- ALM uses end users to replicate datagram. At first, it seems to be a disadvantage. However, it uses unloaded links to send multicast data. For example in Fig 2.4, assume there is a transmission between sender and receiver 1. To achieve this transmission, the normal unicast IP protocol uses the links between 2.Router - 3.Router and 3.Router - 1.Router. Assume that, there is not enough bandwidth to allocate on the link between 1.Router and 3.Router. So Receiver 1 does not get the datagram in the requested time resulting in not meeting the delay and jitter requirements of IPTV. If ALM is implemented, the sender can transmit the related information to Receiver-3 rather than forwarding the packets from Receiver-3 to Receiver-1, provided that the other links are not heavily loaded. So, uninterrupted video data can be seen in Receiver-1. Consequently, ALM can transmit data efficiently, and it can balance network load. Moreover, because the network uses all the advantages of IP unicast, it is very flexible and scalable.

The significant disadvantages of ALM compared to IP multicast can be listed as follows:

- There are control overhead and some extra traffic because end users control multicast. Overlay nodes may have to replicate some identical packets on same link which increases the load on the network.

- This increased load may lead to extra transmission delays and increased jitter.

2.3.2.2 Overlay Networks

An overlay network consists of a number of participating nodes, which organize themselves to an overlay topology on the actual IP network. When overlay networks are used for IPTV delivery, some special nodes are introduced to the system. These nodes called as multicast service nodes, which can be interpreted as proxies to the end user. These proxies take routing information from underlying network structure.

In today's network conditions due to deployment and applicability problems of IP multicast, it is not considered to be the most feasible way to provide IPTV service. On the one hand, overlay networks do not change the underlying network structure. On the other hand, they require some kind of special nodes to realize hardware.

The mechanism can see on Fig 2.5 the overlay network above the actual IP network, and the proxies are clearly indicated.

Overlay networks bring many advantages in the sense of providing IPTV, which are;

- Because it uses the underlying network as unicast, deployment will not be a problem. However some external proxies have to be deployed.

- Overlay multicast can reduce redundant data transfer compared to pure P2P technology.

- Unlike ALM it has all the route information, so it does not need to apply some probing algorithm to find out network conditions.

- It is less efficient than IP multicast in terms of utilization network conditions.

- The performance of the network is highly dependent on the number of proxies and placement of proxies are crucial.

- Similar to ALM, it has control overheads.

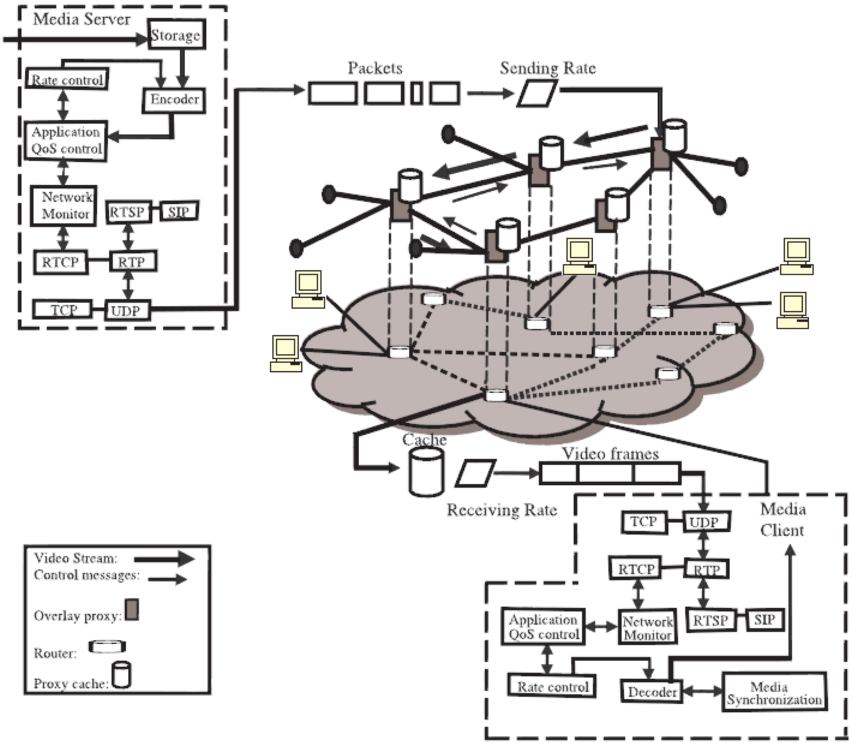


Figure 2.5: Overlay Network Structure.

2.3.3 IMS

IP Multimedia System (IMS) [31] is proposed to provide more flexible service, subscriber control and mobility management. It uses Signal Initiation Protocol (SIP) like VoIP. It is suggested that most of the IPTV demands can be achieved by using IMS's intelligence, such as; user subscription, better QoS control, and session management [2].

IMS protocol can be defined more specifically as: "IMS is a global, access-independent and standard-based IP connectivity and service control architecture that enables various types of multimedia services to end-users using common Internet-based protocols." [7]. In other words, it is like a framework to deliver IP multimedia services. To achieve that instead of using standardized infrastructure, IMS stands on a horizontal control layer which isolates service from the access network. IMS can be defined as a medium for Fixed Mobile Convergence (FMC) as its integrated services.

In Fig 2.6 the basic structure of IMS based IPTV solution can be seen.

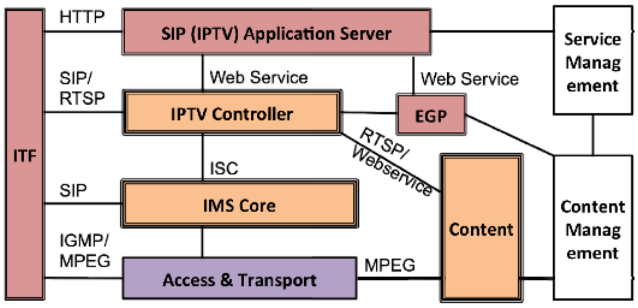


Figure 2.6: IMS Structure [2]

IMS uses Session Initiation Protocol (SIP) for communication. SIP is a signalling protocol, which enables to control multimedia services like, VoIP, video among other services. It is a TCP/IP based Application Layer protocol. It does not depend on the transport layer.

To provide IPTV with IMS architecture the following requirements have to be satisfied in the network [8]:

- Interworking support has to be supplied. Subscribers have to reach other subscribers regardless of terminal properties.
- Roaming support
- Security controls should be done in IMS architecture. IMS takes security into account by including its own authentication and authorization mechanisms among other procedures

Alcatel-Lucent Bell Labs analyzed the IMS-based IPTV. The results indicated that IMS-based resource control mechanism generates 50 percent more signal messages compared with non-IMS based approaches. By control messages of services on demand, it is a three to four increase in the control messages. By linear services (broadcast), this increase is seven to eight times. Network congestion probability with messages by customer Personal Video Recorder (PVR) activities is within IMS-based approach very high. PVR customer sessions will begin simultaneously during the most watched period. Messages of control sources can be in IMS-based approach the reason of increased signal load during the most watched period of up to 66 percent [18].

2.4 Technologies for IPTV

To achieve IPTV services, there are some helper technologies. In this section, the related technologies, Probing and Video Compression are examined.

2.4.1 Probing

For IPTV architectures, available bandwidth in the network is one of the critical parameters. The effects of available bandwidth are discussed 2.2.1, in detail. For ALM applications, if the available bandwidth on the links is known, more intelligent routing mechanisms can be carried out in the application layer. In order to find the available bandwidth some basic techniques are used, one of which is probing. Probing provides a means for end points to make inferences about the properties of a network path. One such property is available bandwidth.

The simplest and the most successful probing technique is active probing. In active probing, a few test packets are sent through the path and estimates the network load by looking end to end delay. End to end estimation has received notable attention and in literature, there are plenty of different active probing tools. Looking at the big picture, these systems infer the available bandwidth of a network path by sending a few packets and analyzing the effects on the probe frames of intermediate nodes and cross-traffic. Examples of probing tools are Pathload[37], IGI/PTR [38], PathChirp [39], and ASSOLO [40]. These tools differ in the size and temporal structure of probe streams, and in the way the available bandwidth is derived from the received packets.

IGI/PTR [38] uses a sequence of about 60 irregularly spaced packets to probe the network, and the gap between two consecutive packets is increased until the average output and initial gaps match. Pathload[37] uses constant bit-rate streams and alters sending rate in each lap. PathChirp[39] uses bit-rate stream packets, which exponentially spaced. ASSOLO [40] is a tool based on the same principle with PathChirp, but it provides a different probing traffic and utilizes a filter to improve the accuracy and stability of results.

In IPTV case, there are plenty cases, which have problems of network impairments, packet losses, significant jitter and delay. Moreover, IP networks have no guarantee on QoS. The popularity of ALM and overlay networks began to rise in the context of IPTV [56], which generates an option that service providers can transmit data packets over several network paths to receiver [57]. Such a kind of path variability gives one more adaptation option, dynamically switch the transmit path according to the observed-estimated network load conditions. This variability can be performed using the probing algorithms mentioned earlier typically using dummy probing packets.

2.4.2 Coding and compression

The most challenging part of IPTV is to meet the requirement for high bandwidth so, bandwidth reducing techniques become substantially important. Among these techniques, coding and compression technology is the most remarkable and effective one. The main purpose of it is to decrease the number of the bits that are required to transmit a video image. As a result of that, bandwidth is used more efficiently, and end users meet with desired service quality. Current coding and compression techniques will be discussed in this subsection.

An uncompressed video bandwidth requirement can be calculated. Suppose that, we have an HD video which has frame size of 1280*720 and frame rate of 24 fps. Assuming that the colour depth is 24 bits, and no compression is used, the required bandwidth is 1,59 Gbps [30]. According to that conspicuous result, some compression is essential to transmit a video over a broadband network. Moving Picture Experts Group (MPEG)[32], International Telecommunication Standard Sector (ITU-T)[33] , and Microsoft Media [34] are very commonly used compression technique for IPTV. The main advantage of this coding and compression technology is it reduced the required bandwidth for, encoded content quite significantly. For example, MPEG-2[35] encoded to broadcast TV in standard definition has dropped the bandwidth around 2 Mbps, which is approximately 270 Mbps when no compression done, and MPEG-4[36] can provide similar quality at 1 Mbps. This coding and compression have not only resulted in lower bandwidth but also advanced functionality, which is very important for IPTV service. Such as MPEG-4 includes media object function, where the part of a video content displayed simultaneously on the screen can be changed.

The following part is about Scalable Video Coding (SVC)/Layered Video Coding [59], which is one of the key parts of this thesis. More specifically, we used SVC to change transmitted data size in order to meet the requirements of IPTV. In H264/SVC standard, there are different layers of video data namely, spatial, quality, and temporal and a new protocol called inter layer prediction, which improves the overall network efficiency. The basic idea is to split the content into two basic parts, base layer and enhancement layers 2.7. The subscribers can decode the content according to their desires, terminal capabilities and received part of the content.

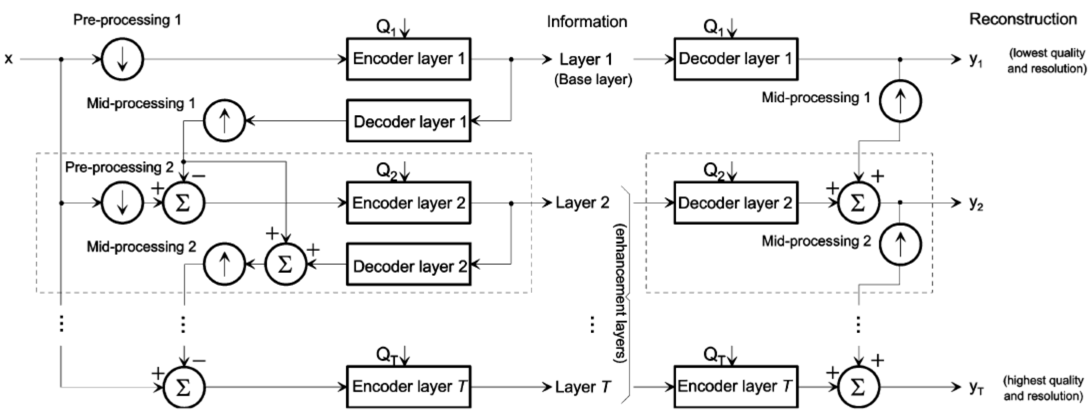


Figure 2.7: Principle of SVC .

The main gain from SVC is its adaptation to network conditions and heterogeneous peers very well. In IPTV concept, there are multiple resolutions of the same video is requested. Without the SVC, for each resolution there has to be different data packets, different routes, much more bandwidth demand. SVC makes the IPTV delivery is less complicated.

2.5 Summary of Literature Search

There are plenty of different solutions in literature to provide IPTV service among whole network elements, namely, NGN-based structures, IP Multicast solutions, P2P structures, IMS-based IPTV architectures. IP Multicast, NGN-based structures and IMS-based IPTV architectures have big advantages about fulfilling IPTV requirements. However, some alterations have to be done in underlying network infrastructure to achieve these solutions.

We observed that in P2P structures, there is no need to make any change in the core network. Hence, we decided to use P2P Streaming to fulfill IPTV requirements. In the scope of this thesis work we designed some control algorithms to alleviate the issues in P2P architectures that are discussed in Section 2.3.2.

We compared CALMTV's results with IP-multicast and P2P streaming in Table 2.1. For this comparison, for CALMTV we used 100 node overlay architecture with 0.5 peers/sec rate. According to these results; although CALMTV can not satisfy zapping delay requirement, it improves current P2P Streaming tools' performance. Moreover, its E2E delay performance is very solid and covers IPTV needs. However, it does a dis-improvement about jitter. In addition to this table, CALMTV's error handling and scalability performances are investigated in Sec 4 in detail.

Table 2.1: CALMTV vs. Other IPTV Proposals' Performances

Metrics	IPTV Requirement	IP Multicast	P2P Streaming	CALMTV
Jitter	<30 ms	0	50 ms	143 ms
E2E Delay	<2 sec	<2 sec	30 sec to minutes	98 ms
Zapping Delay	<0.43 sec	0.3 sec	<2 min	<0.57 sec

CHAPTER 3

CALMTV: CLUSTER BASED ALM (APPLICATION LAYER MULTICAST) IPTV ARCHITECTURE

IP Multicast and IMS are not easy to implement. First, there has to be some alteration in current core network structure in order to carry out IPTV. Moreover, due to dynamic condition in the network (high arrival and departure rates) P2P networks are more efficient in the aspect of scalability. Consequently, we decided to design and implement our IPTV architecture with P2P streaming. To handle IPTV requirements, an Application Layer Multicast (ALM) structure is designed on top of P2P streaming. Furthermore, we incorporate probing for measuring delays and scalable video coding (SVC) in our architecture and we propose an algorithm for the control of ALM. To this end, we name our architecture "Cluster Based ALM IPTV Architecture" (CALMTV). Block diagram of CALMTV can be seen on Fig 3.1.

Our design consists of four main sub-blocks namely; clustering/error management, probing, video compression control and data transmission. Clustering Block is responsible for handling new arrivals and departures in the P2P system. Data Transmission Block is in charge of forwarding video streaming data. Clustering Block and Video Compression Blocks are working together in order to stabilize the system quickly, in case of any error. Furthermore, Video Compression Block changes the transmitted video's bandwidth by changing video layers enhancement to base or vice versa. Probing Block is used like a toolbox and serves other blocks when some estimation is required from the underlying network. Through this chapter, all of these blocks and their relationships will be analysed.

3.1 Clustering Block

This block is responsible for controlling arrivals and departures in the network. It divides the network into clusters and assigns a leader to each group. The clustering block admits new comers to a cluster according to the locations of the subscribers, which is estimated by probing.

Clustering Block has three main parts. The first part accepts new peer arrivals and locates the nearest clusters to corresponding arrival. Moreover, it deals with departures. The second part serves the error management block. If the error sensation mechanism detects an error, it may demand some rebuild in clusters. The third part consists of inter-cluster messaging which runs in the background. We designed one control algorithm for each part. In the next sections 3.1.1, 3.1.2, 3.1.3; control algorithms and their logic are explained. Moreover, this block uses Probing Block whenever it demands some information from underlying network. In addition to this, it uses buffering technique for error correction purposes. The detailed schema of Clustering Block can be seen on Fig 3.2.

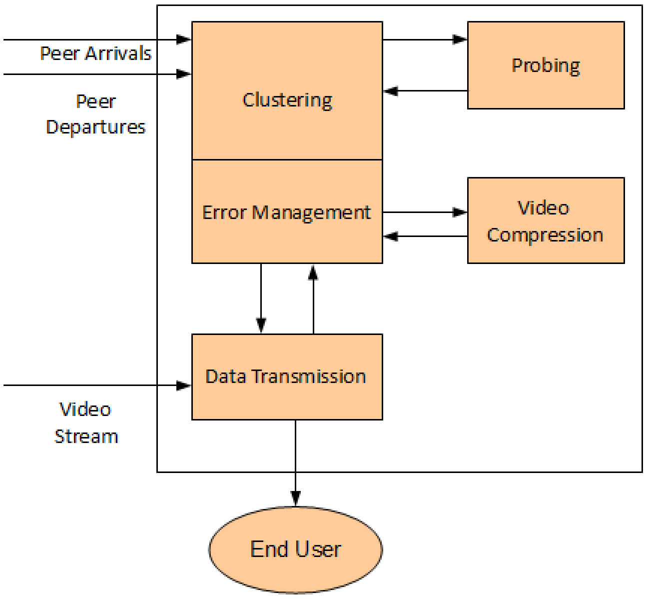


Figure 3.1: Block Diagram for CALMTV.

To understand this cluster structure, let's look at a simple example in Fig 3.3. In Fig 3.3 blue circled node is video server and blue circled node is a cluster leader. Gray circled nodes are subscribers. Assume that, in this case there is only one video channel. First, server sends the video information to the cluster leader. After that, it forwards this video data to subscribers. Our gain is apparent, because if there is no cluster schema, link between node-6 to node-7 may collapse after more subscriber participations.

3.1.1 Handling Peer Arrivals and Departures

A P2P IPTV network is very dynamic. There can be multiples of arrivals and departures of peers in a short period of time. Moreover, multicast tree has to be controlled to optimize the usage of available bandwith in the network. We propose a Location/Heart Beat Algorithm to solve these problems.

When a new peer arrives, first the nearest cluster is found by Location Algorithm. The main reason behind is that; if video data comes as multicast through the edge network, then there will be a few replication of same data. For example, imagine opposite of this situation. Assume that, two adjacent end users are fed by a distant cluster leader, then leader has to transmit same data to each of the subscribers. It will cause unnecessary replication of the same data and it reduces network utilization. In our solution, the nearest cluster leader will work like a IP multicast router. Each cluster leader only transmits one video channel.

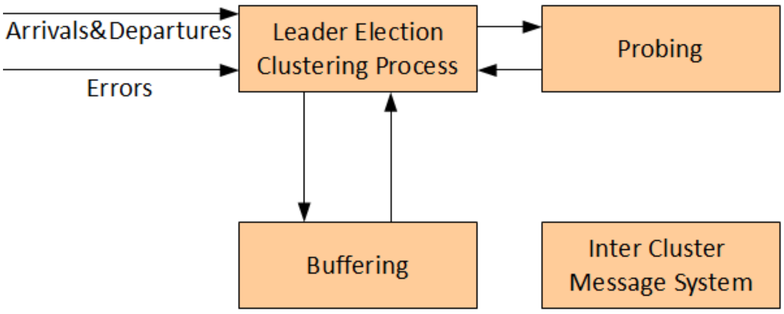


Figure 3.2: Clustering Block Main Structure

When a new subscriber comes, at first it has to send an "Attend Request" message to a random cluster leader. After that, "Clustering Process" begins. This random cluster leader publishes this new arrival information to other leaders. Each leader finds its distance from the new comer with the help of probing. For distance comparison end-to-end delay is used. When probing is completed, results are shared and "Cluster Leader Election Protocol" is initiated. As a result of that protocol, the most appropriate cluster is selected for subscriber. If there is more than one cluster found by the election protocol then the assignment is done randomly. Pseudo code of Cluster Leader Election Algorithm can be seen on Alg 1. If a new comer is accepted by a cluster than cluster leader updates the related subscriber-list information.

Furthermore, to reduce channel switching time we send video base-layer for first 10 frames. After that, we begin to send enhancement layer if requested by subscriber. We choose 10 frames for transient period, because it is a short period that nearly 0.3 sec, subscribers can not recognize the impact of base layer. Moreover, it is a sufficient time to stabilize network conditions; after the new channel assignment.

In steady state all subscribers receive their demanded video data, provided by clusters. If there is a demand for switching channels, the subscriber has to leave one cluster and join another one. Moreover, clusters should not send any information to the peers that left. To this end, a Heart Beat mechanism is created.

.

Heart Beat mechanism uses the basic ping idea. During the subscription period, all the subscribers send "Still Alive" message to cluster leader. If the "Still Alive" message is not received by cluster leader in a certain period of time, the subscriber is deleted from the related cluster leader's subscriber list. This message does three jobs at once. First, it makes sure that the related subscriber still wants the video data. Second, it carries a time stamp. It controls whether it takes to long to deliver the packets. In addition to time stamp information, it carries the last delivered video packet ID. This last feature is very useful for error correction and retransmission cases. When a Heart-Beat message is received by a cluster leader then it sends Heart-Beat message back to the subscriber with time stamp.

If there is a channel switch demand than with the help of "Cluster Leader Election Protocol", new

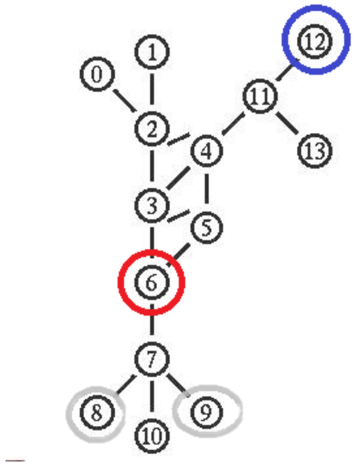


Figure 3.3: Clustering Example

cluster leader is chosen. After that the same procedure is executed. To leave a cluster there is no special protocol. Heart-Beat algorithm takes care of it.

We choose threshold time as 300 ms for the large networks which have more than 30 nodes. In these cases, our E2E delay value never goes beyond 250 ms. Moreover in case of error, E2E delay value have the lower limit of 400 ms, so we can identify error cases as well. For small networks which have less than 30 nodes, have a threshold value of 600 ms due to our inter messaging overhead which will be discussed in proceeding sections.

3.1.2 Error Management - Cluster Rebuild

In a P2P video transmission network, errors may happen. In this section we analyse the cluster errors. We call this error type subscriber sensed errors. If a subscriber node does not get any answer from previously assigned cluster leader after a certain period of time, it concludes that there may be some error in the links or in the related cluster leader. To solve this, subscriber sends another "Attend Request" in order to receive data and the election process repeated. After that, if there is another cluster leader with the same channel information with minimum distance, that cluster leader is selected. To get the video data subscriber node just sends last "Still Alive" message to the new leader.

However, there is another issue here. Because subscriber node had not got the video data during the new cluster election, there was a time difference between broadcast video and subscriber sensed video. To overcome it, we introduce buffering in cluster leader nodes. Buffering is very important to watch a whole video, it helps to handle variations in delay (jitter). Cluster leader nodes have the responsibility of buffering their channel for at least 10 minutes. Buffering may take place in sender, but that means if there is a retransmission or some error, the send request has to be transmitted though the sender. Also, the video stream has to sent through the whole network. To avoid such a traffic, we thought that

Algorithm 1 Pseudo-code for Cluster Leader Election Algorithm

```
INPUT EstimatedDistance
OUTPUT MinimumDistancedClusterLeader'sDistance
CurrentDistance = Infinite
for All Cluster Leaders do
    if LeaderHasRequestedChannelInfo() then
        if CurrentDistance <Estimated Distance (Probing) then
            CurrentDistance =ProbingResult
            TagCurrentLeaderTemporarily
        end if
    end if
end for
for 10 Frames do
    Send Video Base Layer
end for
Send Enhancement Layer if Requested
```

cluster leaders can keep buffered video data with all layer information. The re-election algorithm has two parts subscriber side and cluster leader's side. Pseudo code can be seen on Alg 2 with subscriber perspective. Besides, on Alg 3 the re-election algorithm can be seen from the cluster leaders view.

It is worth to mention that according to our experiments if more than one cluster is constituted for one channel, it is beneficial for error cases as we discuss in the next chapter.

Algorithm 2 Pseudo-code for Re-Election - Subscriber View

```
INPUTS ThresholdTime, LastStillAliveMessageInterval
ThresholdTime = Variable
if LastStillAliveMessageInterval >ThresholdTime then
    Error Occured
    Execute Alg − 1ClusterLeaderElectionAlgorithm
    Send LastStillAliveMessage
end if
```

Algorithm 3 Pseudo-code for Error Correction In New Cluster Leader

```
Get StillAliveMessage
if RelatedDataIsBuffered() then
    Start SendingBufferedData
else
    Start NormalBroadcast
end if
```

3.1.3 Inter Cluster Messaging

There is a messaging system between cluster leaders to deal with errors and cluster election process. In this system, cluster leaders exchange their subscriber lists and probing results for the execution of "Cluster Election Protocol". Moreover, when there is an error, leaders must communicate among themselves. This messaging system is achieved by a mesh-based interconnection which is explained in next section 3.2 .

3.2 Data Transmission Block

In order to forward the video data and send inter cluster messages efficiently, we investigated ALM node connection algorithms. In literature, there are two basic mechanisms for these purposes; mesh-first and tree-first approach.

3.2.1 Mesh-First Signalling

In mesh-first applications, a mesh topology is created among the cluster members. In general, the sender is chosen as the root. After that, a routing algorithm is executed to build the routing tree. At beginning, mesh topology created, so the topology is known. However, resulting tree topology is unknown. So the quality of the tree depends on the quality of the mesh chosen.

The advantage of the mesh-first becomes apparent here as it gives more freedom to refine the tree. It is possible to manipulate the tree topology to a significant extent by selecting mesh neighbours and changing the metrics. A mesh-first approach is therefore more robust and responsive to tree partitions and is more suitable for multi-source applications, at the cost of higher control overhead.

3.2.2 Tree-First Signalling

By contrast, in the tree-first approach, the tree is built directly without any mesh. The members explicitly select their parent from the known members in the tree. This may require running an algorithm to detect and avoid loops, and to ensure that the structure is indeed a tree. There is no intervening mesh topology here. The reason for using the tree-first approach over the mesh-first approach is that the tree-first approach gives direct control over the tree. This control is valuable for different aspects such as maintaining strict control over the fan-out, selecting a best parent neighbour that has enough resources, or responding to the failed members with a minimum impact to the tree. Another advantage of the tree first approach is independent actions from each member. It makes the protocol simple as it has a lower communication overhead. But when a member changes a parent, it drags all of its descendants with it. This is desirable in the sense that the descendants do not need to change their neighbours; in fact, they are indeed unaware of the incident.

3.2.3 Performance Requirements of ALM Messaging Algorithm

We examined some ALM algorithms from the perspective of inter connections. These algorithms are tree-first algorithms, ALMI [12], OMNI [13]; and mesh-first algorithms, NICE [14], NARADA[11].

First we listed our expectations from an ALM inter communication algorithm and we investigated whether these ALM algorithms fulfill our requirements or not in the sense of inter messaging.

- Low Latency: Our first desire is to select a messaging algorithm which has the minimum delay. Experiments show that average streaming delay (network delay + node delay) changes with used mesh/tree structures [5]. Among four algorithms, ALMI has minimum delay because of its tree structure. However, it is worth to mention that if the number of nodes increase than tree-based protocols' delay will increase. NARADA has the smallest delay when the network size is not big, if network size is growing than its performance will not be definite [3]. Fig 3.4 shows delay variations with respect to network size.

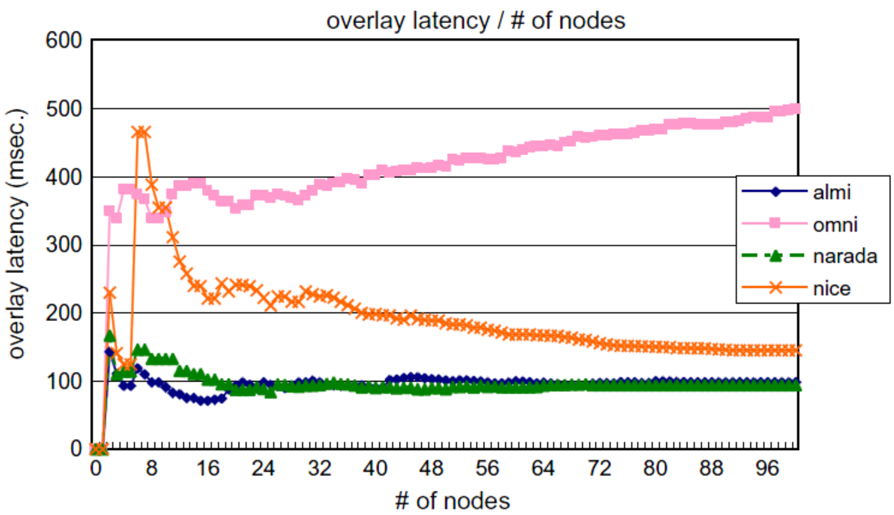


Figure 3.4: Average Delay from Source to End Node [3]

- Low Bandwidth Usage: Bandwidth utilization is another critical factor when doing multicast. Mesh topologies have some advantages when network size tend to grow. On the other hand, tree topologies have some drawbacks, if errors occur. Bandwidth usage of selected algorithms can be seen in Fig 3.5.

- Low Relative Delay Penalty: Relative Delay Penalty (RDP) means that the ratio of the delay from centre to other nodes in the topology with unicast delay [3]. It can be said that if RDP value is less than a certain threshold value, our algorithm becomes efficient [3]. According to our research with the increase in the number of the nodes, ALMI and NARADA have larger RDP values. OMNI has larger RDP values, too. However, NICE has the best resulting RDP for the large networks. Behaviour of the algorithms in the sense of RDP can be seen on Fig 3.6.

According to our investigation although mesh-based protocols have larger delay variations (jitter), their higher scalability is a very big advantage for our case. So, we decided to choose and implement

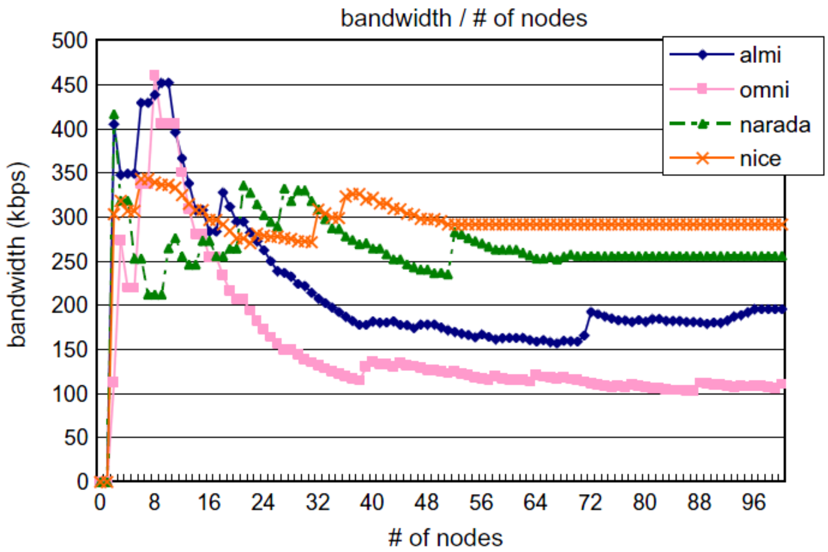


Figure 3.5: Average Bandwidth Usage [3]

mesh-based messaging structure.

3.3 Video Compression Block

This block is responsible for determining which video layer will be downloaded by the subscriber. It is a very critical issue when traffic overloaded or there is an error in transmission process. We assume that all the base and enhancement video layers are distributed among cluster leader nodes. This block mainly uses probing results to decide whether there is a necessity to change the layer that is viewed by the user.

There are no special hardware requirements. However, we assumed that cluster leader nodes keep the full video data with layers. We used this property to avoid jitters, as well. To meet with the jitter requirements of IPTV, we buffered latest one minute of selected channel. In cluster leader nodes, one-minute buffered video data has a reflection in memory consumption. Hence, cluster leader nodes have to separate approximately 100 MB memory to channel-buffers.

First, it is discussed that all cluster leaders gets for the “Still Alive” messages from their subscribers. They are not only get these messages for departure, but also they use it to calculate time intervals. Cluster leaders watch message arrival interval trends and tag the interval as growth or decay. If there is an increase in delay results, leader compares this delay with a threshold value which is set before. If it is above the threshold value it starts the “Layer Switch Algorithm”. This algorithm has two main

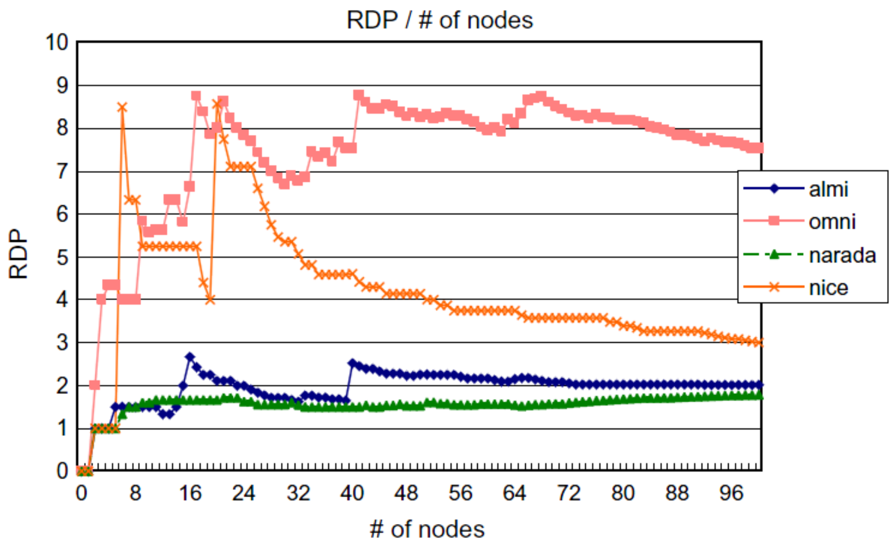


Figure 3.6: Average RDP values [3]

steps. First, it sends a message to other cluster leaders if there is any other cluster with the same channel information and delay is under than a threshold value set before. If there is such a leader then subscriber is registered to new leader. If there is not such a cluster, then we change channel quality in order to avoid any service loss. Our algorithm's pseudo code can be seen on Alg 4.

.

3.4 Probing Block

The main purpose of this block is to provide the bandwidth estimation to other blocks, whenever it is necessary. To select a probing algorithm, the existing algorithms have been researched. Their advantages and disadvantages are investigated in the context of IPTV.

The performance metrics of a a probing algorithm are high estimation accuracy, low bandwidth overhead, and low response time as listed below:

- High Estimation Accuracy :

  High estimation accuracy is very important for our design because all the other blocks use the output of this block. If the estimations have high errors, then we can not identify correctly the low loaded and the high loaded links in the network. As a result of that, error sensation fails, cluster election protocol fails, as well. So, high estimation accuracy is crucial for CALMTV.

Algorithm 4 Pseudo-code for Video Compression Control

```
INPUTS MaxEndToEndDelay, ProbingResult
Set ThresholdTime = MaxEndToEndDelay
if StillAliveMessageInterval > ThresholdTime than then
    for All Clusters do
        if LeaderHasRequestedChannelInfo() then
            if ProbingResult < ThresholdValue then
                RegisterNodeintoCurrentCluster
            end if
        end if
    end for
    if ThereIsNotSuchACluster then
        Change Distributed Layer (Enhancement/Base)
    end if
end if
```

- Low Bandwidth Overhead :

  Because cluster handling, data transmission and error sensation have high loads already, we do not want some other cause to create extra traffic in the network. So, to not complicate the overall structure, selected probing algorithm has to have small bandwidth overhead.

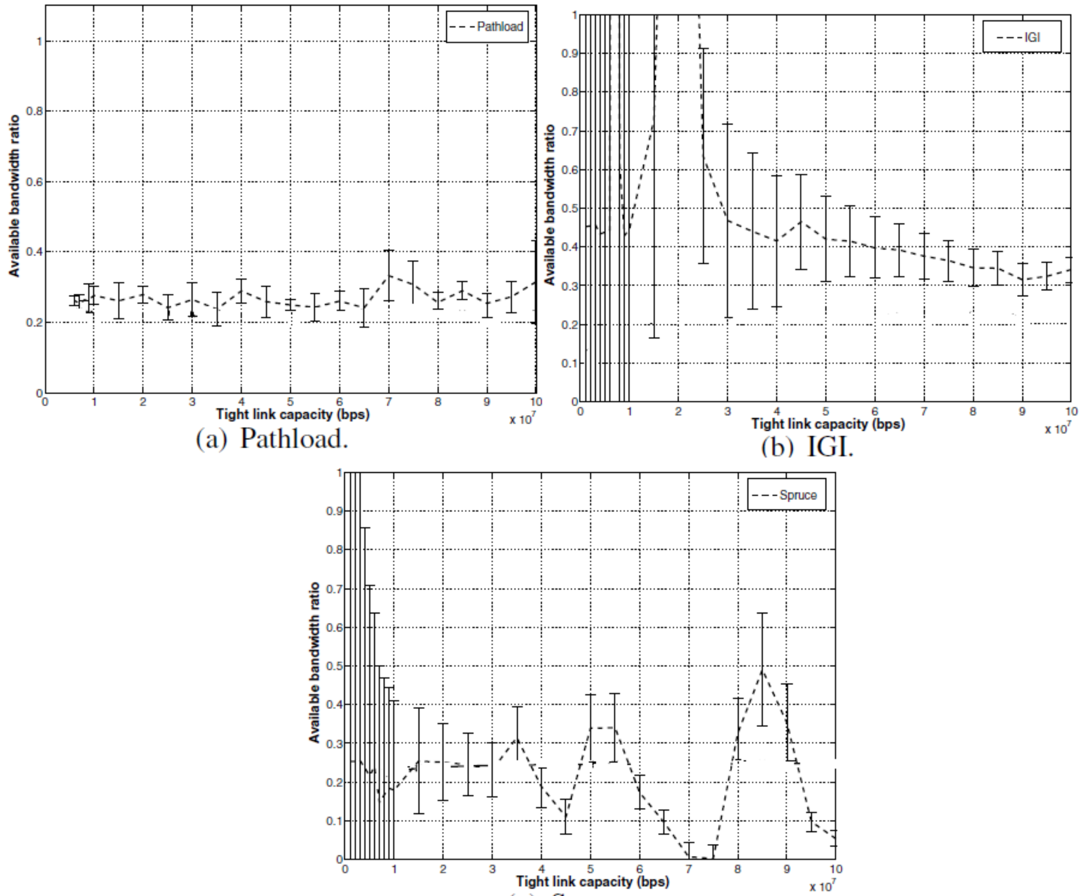
- Low Convergence/Response time :

  Because our design implements a real time application, reaction times are very important. To get instantaneous results from this block, our probing algorithm needs to have low convergence time. However, if the bandwidth overhead is small, we can run probing algorithm in background all the time. So, low convergence time may not be indispensable.

In the literature there are plenty of different probing algorithms. Each of these has standing out capabilities in terms of bandwidth utilization, high estimation accuracy and low convergence time. Our main goal is to select the one which satisfies our demands best.

We investigated the performances of Spruce [49], IGI [38] and Pathload [37] algorithms.

From the perspective of estimation accuracy, researches show that Pathload has the most accurate estimation results among them. It has less than 20 percent error rate in any scenario [47] . However, sometimes it overestimates the network load and calculates for the worst case. Both Spruce and IGI have some accuracy issues. Their error rates are far above than Pathload [47]. Moreover, their predictions are 30 percent wrong in most cases. They present some problem while predicting in highly loaded system, they underestimates network load. Figure 3.7 shows estimation results of the algorithms.



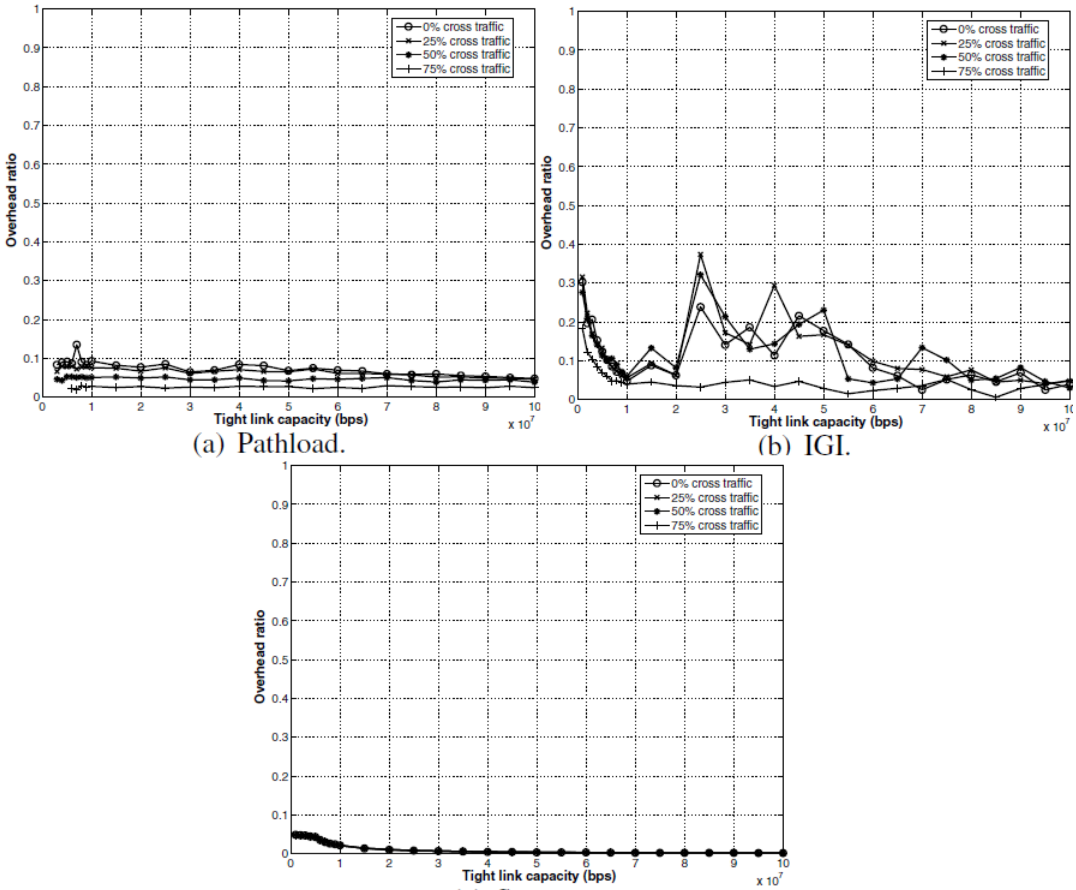
(a) Pathload.

(b) IGI.

(c) Spruce.

Figure 3.7: Estimation Accuracies [47]

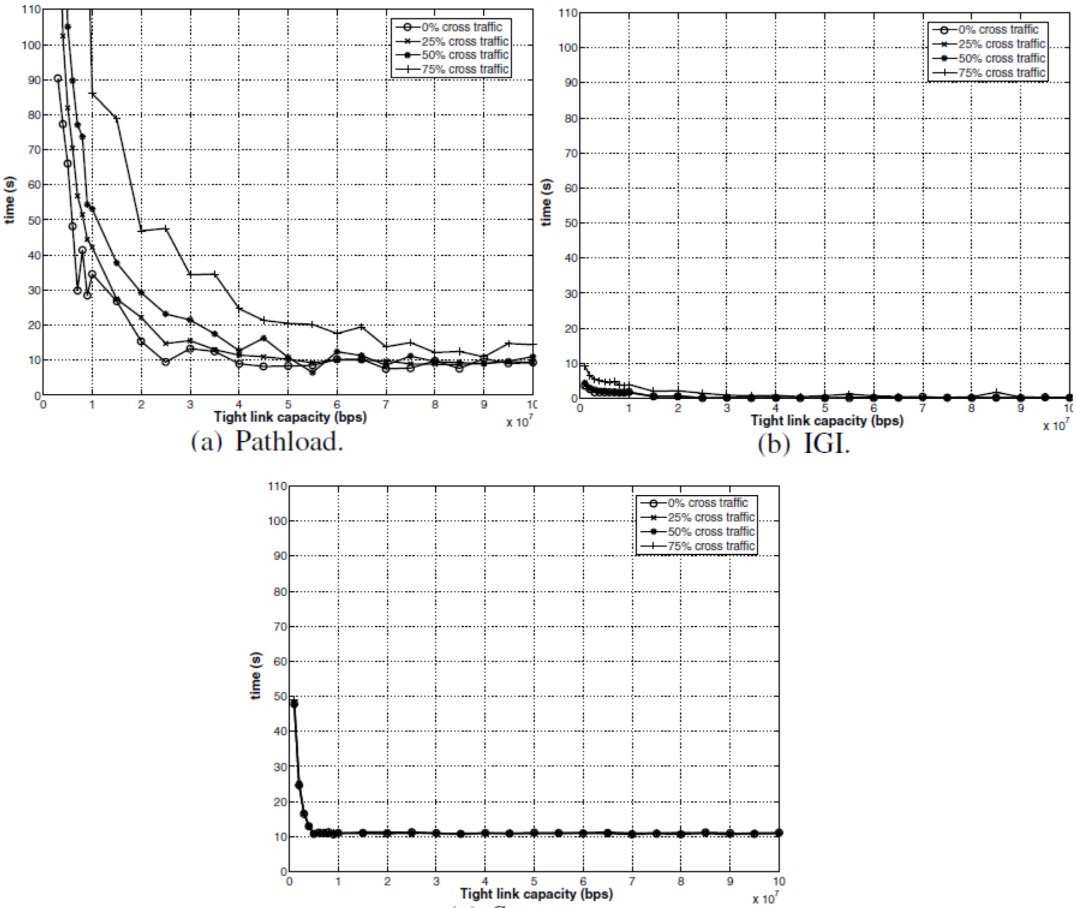
From the aspect of bandwidth overhead, Pathload has respectively small overhead [47]. Its overhead never exceeds 10 percent in tight links. When we look at Spruce's performance, spruce has the smallest estimation overhead [47]. It utilizes bandwidth usage, too. On the other hand, IGI's overhead can grow up to 30 percent and it can be very problematic in our case. Their load overheads can be seen on Fig 3.8.



(a) Pathload. (b) IGI. (c) Spruce.

Figure 3.8: Estimation Overheads [47]

Lastly, their convergence time is varied by link capacities and traffic loads. IGI has the smallest convergence time, it almost never exceeds 10 seconds. Spruce has relatively high convergence time around 10 seconds constantly [47]. On the other hand, Pathload has to have much time to converge [47]. This is not surprising because it uses lower bandwidth and it brings more accurate results than others, it has to do more iterations to get it. Their convergence time comparisons can be seen on Fig 3.9.



(a) Pathload. (b) IGI. (c) Spruce.

Figure 3.9: Convergence Times [47]

According to our investigation, despite of its large convergence time; Pathload's estimation accuracy and low bandwidth overhead assure that Pathload can handle our requirements. We decided to eliminate Spruce and IGI due to their low accuracy,high bandwidth overhead. So, based on all of these we decided to pick Pathload.

Pathload uses Self-Loading Periodic Streams (SLoPS) to estimate bandwidth [48]. Its main principle is to transmit a periodic stream with one way delay. When probing traffic is larger than available traffic, then it increases the delay of periodic data streams. A fixed number of packets are sent and one way delay of each stream are classified as growth or decay in at receiver. Pathload sends these streams by

UDP, while receiver sends trend results back to the sender. If stream rate is R, then Pathload picks an inter-departure time T. After that, it calculates the necessary packet size to make sure R = L/T.

3.5 Novel Features of CALMTV

Our contribution and novel features can be listed as below:

- ALM communication algorithm, probing, and SVC ideas are combined in order to get more effective results. With this combination, we can predict and control available network resources.
- New cluster handling algorithm upon the ALM mesh-based inter connection protocol is designed. Our cluster handling algorithm mainly works like IP multicast. The main difference is that, it runs on the access network.
- According to available bandwidth, a video layer control algorithm is designed. With this algorithm CALMTV can reduce used bandwidth to avoid error cases or enormous delays.

CHAPTER 4

PERFORMANCE EVALUATION OF CALMTV

4.1 Simulation Environment

To test our algorithm, we used ns-2.29 [46] to conduct experiments. For Probing Block 3.4 and Data Transmission Block 3.2.3, open source codes are used. For Cluster Block 3.1 and Video Control Block 3.3, we implemented our algorithms in ns-2.29.

4.2 Traffic Models

We used Georgia Tech Internetwork Topology Model (GT-ITM) [45] for topology generation, our simulations can cover up to 500 node-networks. (GT-ITM) is a C program which allows defining transit-stub model topologies. The variables, number of nodes and edge structure, are read from the configuration file, and then an output topology graph is generated. For more information about the configuration file format and mechanism, please check appendix A.

To construct our overlay model, we constructed a virtual topology on to real network topology. Virtual links are established based on network closeness which is measured as delay by probing block explained in Section 3 in detail. All of CALMTV messages are transmitted through one virtual node to another node by the real network links. For routing protocol, we used Link State Routing Protocol because it uses delay as a metric, it computes shortest-path. In real network we generated a background traffic. For background traffic, we generated TCP and UDP point-to-point traffic with 576 bytes packet size and normal distribution mean 500 kbits/sec.

4.3 Assumptions

Our basic assumptions are:

- We used an arrival-departure rate of 1 peers/minute. In subsection 4.5.2, we increase this rate in order to stress the underlying structure.
- In section 3.1.2, it is explained that buffering is very important to provide non-stop service. We assume that, these buffers have 100 MB capacities which take the memory from RAM. Otherwise, managing memory consumption issues may be very problematic.

4.4 Performance Metrics

We evaluated CALMTV with the following aspects, in order to compare its performance with other IPTV services in the literature.

- Error Resilience : We created basic error cases to appraise this property. We observed CALMTV's performance under different stress conditions.
- Channel Switching Time : Subscriber sensed zapping times were calculated.
- Delay/Jitter : Delay variability (jitter) and end to end delays were measured.
- Delay / Protocol Overheads : We measured our algorithm blocks' bandwidth overheads to see their actual benefit.
- Scalability : We measured end to end delay under high arrival/departure rates to look from the scalability point of the view.

4.5 Scenarios and Experiment Results

We executed plenty of scenarios with different arrival-departure rates, network sizes. Moreover, we created some error cases, which include cluster leader errors. Our topology can be seen on Fig 4.1. For our experiments, we used 10 different clusters and 7 different video channels. In the Fig 4.1, red circled nodes represent cluster leaders and blue circled nodes represent video channel servers. First server, which is numbered as three, supplies first, second, and third channels. Second server, numbered as 52, supplies fourth, fifth, sixth, and seventh channels.

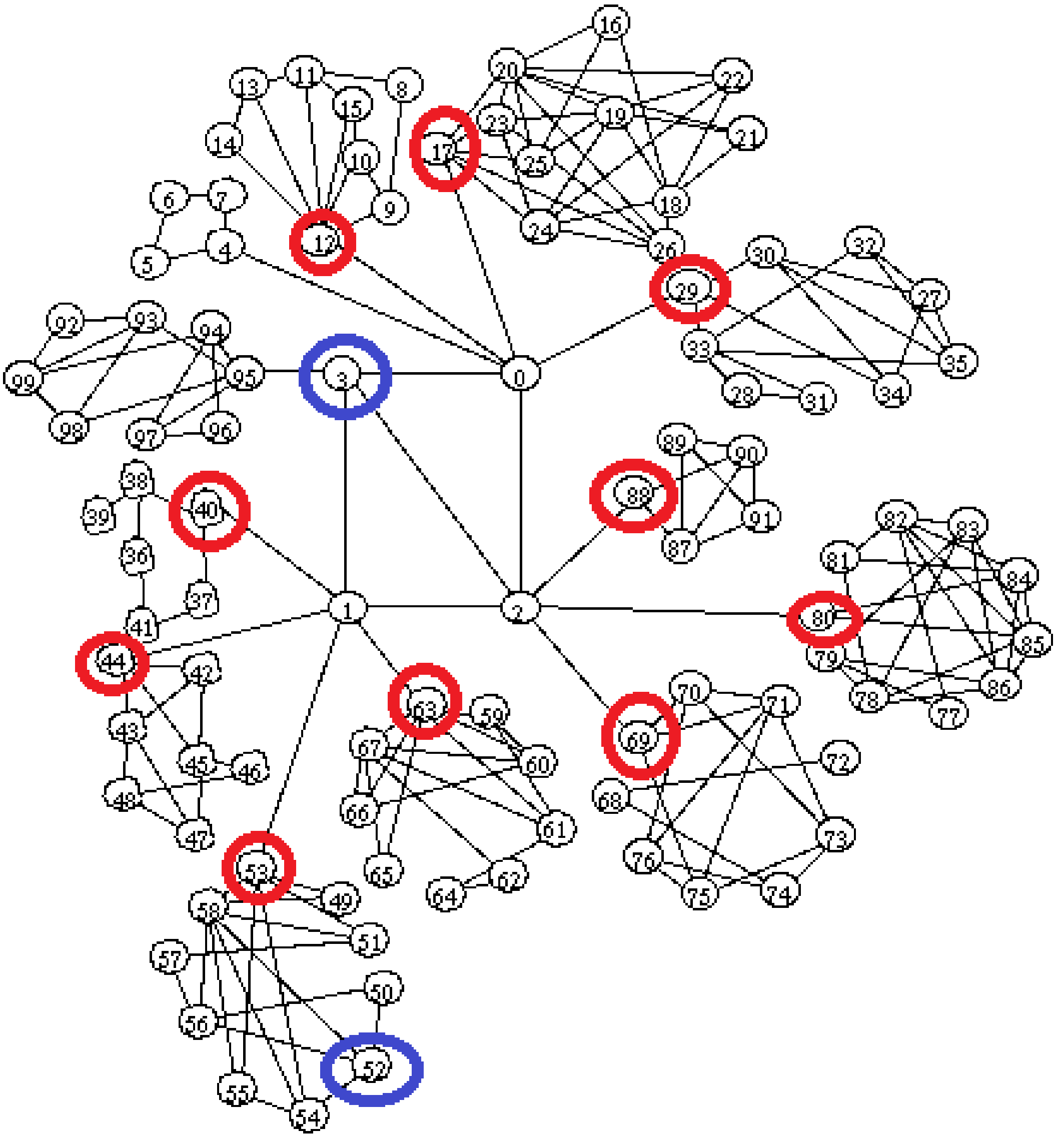


Figure 4.1: Experiments Topology

Cluster leaders are responsible for different channels as well. Cluster leader - video channel assignments can be seen in table 4.1.

Table 4.1: Cluster Leader - Video Channel Assignments

Cluster-1:	1. - 2. channels
Cluster-2:	4. channel
Cluster-3:	7. - 5. channels
Cluster-4:	2. channel
Cluster-5:	3. channel
Cluster-6:	5. - 6. channels
Cluster-7:	2. - 4. channels
Cluster-8:	4. - 5. channels
Cluster-9:	6. channel
Cluster-10:	7. channel

For each channel, we transmitted one base layer and one enhancement layer bit stream. Bit rates of these streams can be seen in table 4.2. We selected these rates because these are a widely used test video sequence's bit rates among developers of video codecs [4].

Table 4.2: Bit Rates For Layers

Bit rate BL(kbit/s)	350
Bit rate EL(kbit/s)	2100

We conducted four different experiments onto this structure. These scenarios are explained in following sections.

4.5.1 Scenario One - Normal Case

For the first scenario, we conducted our experiments under normal conditions. We used 1 peers/minute arrival-departure rate. In this case, we investigated effects of different network sizes. To achieve that, we used same base topology in Fig 4.1 and same channel assignments. But to increase the size, we added new nodes at end parts. For example, for 110 node case we used the topology in 4.2. Moreover, to decrease the size we cut some end users without touching the base network.

Furthermore, only for this scenario; we calculated E2E delay for different base layer bit rates which will be identified in the next subsection. Also, we calculated E2E delay when there is only base layer stream exists.

In this case, we assumed that there are no errors and nothing special happens in the network.

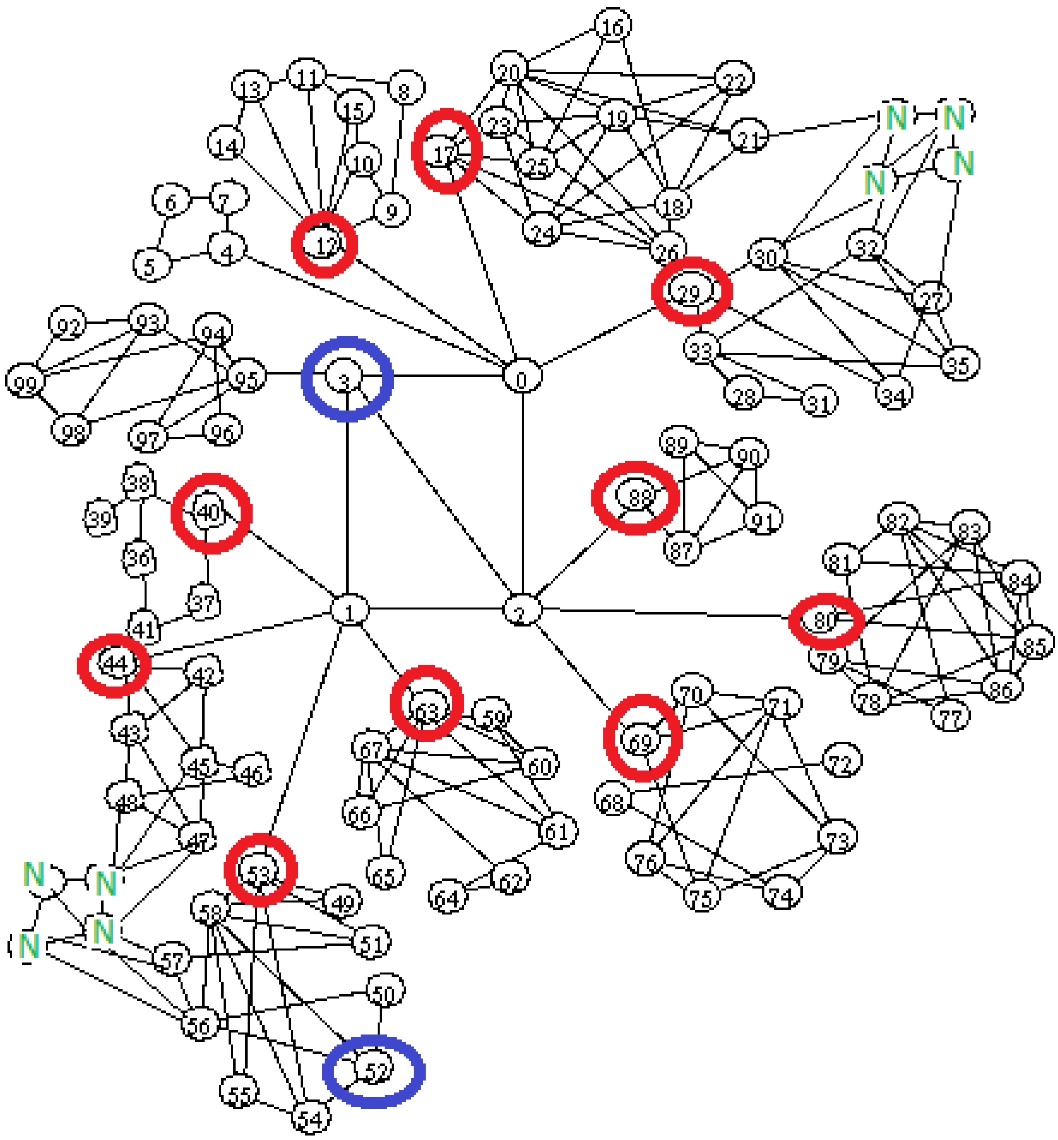


Figure 4.2: Experiment Topology for 110 Nodes

4.5.1.1 End To End Delay

We examined observed average E2E delay, which is a very important performance metric of IPTV. During the transient period, E2E delay is higher than in small networks because our cluster based architecture assumes relatively larger networks. When the number of nodes is small, the overhead of cluster routing makes the average delay larger. However, as the number of nodes increases, more effective clusters are constructed and the average delay becomes smaller like NICE. After that, performance is stable. When we compare this result with other ALM algorithms and our ALM algorithm's performance in Table 4.4, in this comparison overlay node numbers are fixed. CALMTV's average E2E delay is less than other ALM algorithms. Since NICE uses more complex cluster control algorithm, this performance is expected. We will investigate whether this complexity comes in handy in error cases in scenario three. Because ALMI has tree-based structure, its E2E delays are lower in small networks. CALMTV's E2E delay performance meets with IPTV requirements specified in Section 2 which has to be lower than 2 seconds.

We investigated effects of different bit rates in layers. For this purpose, we designed 3 test cases. In first case, we used our original bit rates. For second case, we used tripled base layer bit rate. For the third case, we removed enhancement layer information from video stream. The exact numbers for bit streams can be seen in 4.3.

Table 4.3: CALMTV's Performance: Bit Rates For Layers

Bit Rates	Case - 1	Case - 2	Case - 3
Bit rate BL(kbit/s)	350	1050	350
Bit rate EL(kbit/s)	2100	2100	-

Table 4.4: E2E Delays in Different ALM Algorithms

Number of Nodes	NICE	ALMI	OMNI	NARADA	CALMTV - Case 1
25	220 ms	100 ms	380 ms	100 ms	431 ms
50	170 ms	100 ms	420 ms	100 ms	200 ms
75	150 ms	100 ms	470 ms	100 ms	91 ms
100	140 ms	100 ms	500 ms	100 ms	89 ms

In table 4.5, the effects of different bit streams can be seen. Case - 2 delays are higher than Case - 1 delays. Since their enhancement layer rates are same, the delay increase shows that CALMTV overlay nodes do not distribute enhancement layers sometimes. From Case - 3, it can be concluded that, enhancement layer streams affects the delay values, they are distributed by CALMTV overlay.

Table 4.5: CALMTV E2E Delay Performance in Different Bit Rates

Number of Nodes	CALMTV - Case 1	CALMTV - Case 2	CALMTV - Case 3
25	431 ms	508 ms	317 ms
50	200 ms	301 ms	173 ms
75	91 ms	221 ms	87 ms
100	89 ms	218 ms	69 ms

4.5.1.2 Jitter

In our experiments, we measured average jitter as the change in delay latency with respect to time. We analysed CALMTV's performance in the aspect of jitter, as well. According to our results, in steady state our jitter values are stable lower than other ALM structures [14]. Moreover, changes in network size do not cause any significant effect on jitter performance. Because of our buffering strategy in cluster nodes, CALMTV's performance becomes very stable under variable network sizes. The comparison between other ALM algorithms and CALMTV can be seen in Table 4.6, for this comparison overlay node numbers are fixed. CALMTV's E2E delay performance does not support IPTV requirements specified which has to be lower than 30 ms. However, PPLive's may grow up to minutes. CALMTV's jitter is stable and lower than that value.

Table 4.6: Jitter Values in Different ALM Algorithms

Number of Nodes	NICE	ALMI	OMNI	NARADA	CALMTV
25	650 ms	100 ms	100 ms	620 ms	89 ms
50	680 ms	190 ms	100 ms	720 ms	98 ms
75	780 ms	180 ms	100 ms	720 ms	101 ms
100	783 ms	200 ms	100 ms	720 ms	121 ms

4.5.2 Scenario Two - High Arrival/Departure Rate Case

For this scenario, we want to stress the network. We changed 1 peers/minute arrival-departure rate to 0.25 peers/sec, 0.5 peers/sec, 1 peers/sec, and 10 peers/sec arrival-departure rate. To achieve that, we used same base topology in Fig 4.1 and same channel assignments. We calculated average delay and jitter values for 100 nodes. As seen in Table 4.7 and 4.8 our jitter and delay values are stable up to 0.5 peers/sec rate. Although we expected a fall in delay and jitter performances in high arrival-departure rates, this performance shows that if rate exceeds 1 peers/sec we have serious peer churn problems. Our control algorithms on clusters and reassignments have higher overheads under these conditions.

Table 4.7: Average Jitter Values Under Various Arrival-Departure Rates

Rate	Jitter
1 peers/min	121 ms
0.25 peers/sec	131 ms
0.5 peers/sec	143 ms
1 peers/sec	156 ms
10 peers/sec	267 ms

4.5.3 Scenario Three - Cluster Leader Error Case

In order to evaluate our performance, we included error cases into our analyses. We used the same topology and channel assignments in Scenario One. Our topology consists of 100 nodes in these cases. We created cluster leader errors which stop those cluster leaders' operations. We investigated

Table 4.8: Average E2E Delays Under Various Arrival-Departure Rates

Rate	E2E Delay
1 peers/min	89 ms
0.25 peers/sec	93 ms
0.5 peers/sec	98 ms
1 peers/sec	230 ms
10 peers/sec	803 ms

CALMTV's E2E delay performance and recovery times under these stress cases. We examined recovery times in the states 1 Cluster Leader Failure, 2 Cluster Leader Failures and 3 Cluster Leader Failures. We designed our experiments that failures happen in 5 seconds after starting time. Recovery time stands for completion of full new leader assignments process for end-users.

Recovery times can be seen on Table 4.9. It shows that if there is more than one leader failure recovery times are really high. Actually, it is understandable because there must be new leader assignments for the end-user. We observed a similar response to Scenario Two. In Scenario Two, there are a lot of new channel assignments, too.

Table 4.9: CALMTV's Recovery Times Under Error Cases

Error Cases	Recovery Time
1 Leader Failure	256 ms
2 Leader Failures	493 ms
3 Leader Failures	1700 ms

It can be observed that in all cases after the errors, there is a huge jump on delays. However after recovery times, delays become stabilized.

4.5.4 Scenario Four - Effects of SVC on Channel Switching Times

We investigated how SVC effected on CALMTV's performance. In this case, we used the same topology and channel assignments as in Scenario One. We measured channel switching times. Simply, we created 1 to 10 consecutive channel switch requests. For each case, we found the time in which all new channel assignments are done and all subscribers get requested channel's video information. The results in Table 4.10 show that our SVC control algorithm is very helpful to control channel switching times. As expected, without SVC control, switching times are increased. CALMTV's performance meets with the IPTV requirements on zapping delay as defined in sec 2.2.5 which is 0.43 seconds. However, if there are too many simultaneous channel switching requests, it exceeds the limit.

4.5.5 Protocol Overheads

We analysed our protocols' bandwidth usage to see whether our protocols serve the purpose or not. We conducted our experiments under the same conditions as in Scenario One. However, in this case we increased number of nodes up to 300 nodes.

Table 4.10: Channel Switching Times

Channel Switch Requests	with SVC Control	Without SVC Control
1	0.17 sec	0.21 sec
2	0.19 sec	0.28 sec
3	0.25 sec	0.29 sec
4	0.29 sec	0.38 sec
5	0.33 sec	0.49 sec
6	0.37 sec	0.59 sec
7	0.48 sec	0.79 sec
8	0.57 sec	0.89 sec
9	0.58 sec	0.99 sec
10	0.89 sec	1.5 sec

4.5.5.1 Probing Overheads

We calculated Pathload's performance in the sense of bandwidth overhead. Our experimental results can be seen in table 4.11. We took these results when there is 50 percent cross traffic. Our links have 5 Mbps capacity and Pathload uses 8 percent of the available bandwidth approximately. The results are consistent with the literature [37], [47].

Table 4.11: Probing Overhead

Number of Nodes	Probing Overhead
25	155 kbps
50	283 kbps
75	327 kbps
100	346 kbps

4.5.5.2 Cluster Messaging Overheads

In the beginning phase, bandwidth usage increases very fast due to mesh learning state discussed in section 3.2.1. After that its performance is stable and is shown in Table 4.12. When we compare this result in Table 4.12 with other algorithms, CALMTV's bandwidth usage is lower than NICE and NARADA which are mesh-based, as well. It is expected because CALMTV uses less control messages over the network. Because OMNI and ALMI have tree-based structures, their messaging overheads are smaller than CALMTV's.

Table 4.12: CALMTV and Other ALM Algorithms' Cluster Inter-Messaging Overheads

Number of Nodes	NICE	ALMI	OMNI	NARADA	CALMTV
25	260 kbps	250 kbps	160 kbps	280 kbps	150 kbps
50	300 kbps	180 kbps	130 kbps	240 kbps	250 kbps
75	290 kbps	180 kbps	110 kbps	260 kbps	201 kbps
100	290 kbps	200 kbps	100 kbps	260 kbps	243 kbps

4.5.5.3 Total Protocol Overhead

Finally, we combined probing and mesh-based messaging overheads to see the overall results. Overall bandwidth usage is stable and consistent with the literature search in Table 4.13. When we compare these findings with NICE [14], CALMTV's performance is still better than NICE even when combined with probing.

Table 4.13: Total Protocol Overhead

Number of Nodes	Total Protocol Overhead
25	201 kbps
50	278 kbps
75	302 kbps
100	307 kbps

CHAPTER 5

CONCLUSION

The goal of this thesis is to design and implement a successful IPTV architecture over the current Internet. To achieve that, we designed CALMTV architecture with the following features:

- Probing Algorithm and Analyses
- Cluster Handling Algorithms
- Application Layer Multicast(ALM) Messaging Protocols
- Scalable Video Coding

Our implementations show that with basic control algorithms, in the sense of error management and bandwidth control, IPTV service can be succeeded by using P2P networks. Although, CALMTV's performance could not handle jitter, it assures service in error cases and highly-loaded network conditions. Moreover, in stable cases CALMTV can provide the requested zapping time for IPTV. However, when there are too many channel switching requests, zapping times exceed IPTV limits.

In our experimental structure, we picked cluster leaders at the beginning. This may be a handicap for CALMTV because we give a special meaning to these leaders. If these cluster leaders fail or log out from the network, we have to execute error handling algorithms which have a big recovery times.

In our results, we found that SVC is a very useful tool when controlling available bandwidth in the network. It is beneficial for realizing end-user demands, as well. Moreover, our "Cluster Leader Election" and "Heart Beat" algorithms show with some intelligent control in Application Layer; multicast can be manageable over end-users.

In our design, there is no information collection from routers. To guess network conditions, we used a probing algorithm. We saw that with the intelligent use of probing tools (eliminating redundant growth or decays, etc.) and with a heuristic algorithm, network conditions can be predicted.

In this thesis to simplify the problem, we assumed 100 MB buffers in cluster nodes. For future studies, more realistic memory capabilities may be investigated. Furthermore, to deal with error cases more effectively, our "Video Compression Control" algorithm can be redesigned with some heuristic rate algorithms in the future. Also, we calculated our results on the same base topology. For the future, these tests have to be done on several different topologies with different network sizes.

REFERENCES

[1] Infonetics Research[Online], Available: (http://www.infonetics.com/pr/2009/4q08-iptv-market-research-highlights.asp). (last accessed on 15/01/2013)

[2] Thu-Huong Truong,Tai-Hung Nguyen, Huu-Thanh Nguyen, *On Relationship Between Quality of Experience and Quality of Service Metrics for IMS-Based IPTV Networks*. IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, IEEE RIVF 2012

[3] Kazushi Ikeda, Thilmee M. Baduge, Takaaki Umedu, Hirozumi Yamaguchi, Teruo Higashino *ALMware: A middleware for application layer multicast protocols* Computer Communications, 2011

[4] Mikko Uitto, Janne Vehkapera *Spatial Enhancement Layer Utilisation for SVC in Base Layer Error Concealment* Mobimedia Proceedings of the 5th International ICST Mobile Multimedia Communications Conference, 2009

[5] Carl Livadas *An Evaluation of Three Application-Layer Multicast Protocols* Laboratory for Computer Science, MIT,2002

[6] Video Bitrate Calculator[Online], Available: (http://web.forret.com/tools/videofps.asp). (last accessed on 15/01/2013)

[7] Poikselka, Miikka; Mayer, Georg. *The IMS : IP Multimedia Concepts and Services. Third Edition* United Kingdom: John Wiley - Sons Ltd. 2009. [Consulted on November 2009].[Online], Available: (http://books.google.com/books?id=vxRNUBV4uoMCpg =PP1dq=imshl=esv=onepageq=f=false). (last accessed on 15/01/2013)

[8] Camarillo, Gonzalo; GarcÃa-MartÃn, Miguel A. *The 3G Multimedia Subsystem : Merging the Internet and the Cellular World* United Kingdom: John Wiley - Sons Ltd. 2004. [Consulted on November 2009][Online], Available: (http://books.google.com/books?id=qOSxiMMrA4Cprintsec=frontcoverdq=imshl= esv=onepageq=f=false) (last accessed on 15/01/2013)

[9] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. *Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload* In SOSP, 2003.

[10] S. Rhea, D. Geels, and J. Kubiatowicz. *Handling Churn in a DHT* In USENIX, 2004.

[11] Chu, Y.-H., Rao, S. G., and Zhang, H. *A Case for End System Multicast. In Proc. International Conference on Measurement and Modelling of Computer Systems* ACM Special Interest Group on Measurement and Evaluation ACM/SIGMETRICS 2000, Santa Clara, CA, June 2000

[12] Dimitrios E. Pendarakis, Sherlia Shi, Dinesh C. Verma, Marcel Waldvogel *ALMI: An Application Level Multicast Infrastructure* USENIX Symposium on Internet Technologies and Systems - USITS , pp. 49-60, 2001

[13] Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, Samir Khuller *OMNI: An Efficient Overlay Multicast Infrastructure for Real-time Applications* 22nd Annual Joint Conference on the IEEE Computer and Communications Societies;San Francisco, CA;30 March 2003

[14] Suman Banerjee, Bobby Bhattacharjee *Analysis of the NICE Application Layer Multicast Protocol* UMIACS TR 2002-60 and CS-TR 4380,June 2002

[15] Zhi Li, Xiaoqing Zhu, Member, IEEE, Ali C. Begen, Member, IEEE, and Bernd Girod, Fellow, IEEE *IPTV Multicast with Peer-Assisted Lossy Error Control* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 3, MARCH 2012

[16] Sherali Zeadally, Hassnaa Moustafa, Member, IEEE, and Farhan Siddiqui, *Internet Protocol Television (IPTV): Architecture,Trends, and Challenges* IEEE SYSTEMS JOURNAL, VOL. 5, NO. 4, DECEMBER 2011.

[17] Gang Su, Pham Thuy Oanh, Shuo He, *QoS Guarantee for IPTV Using Low Latency Queuing with Various Dropping Schemes* International Conference on Systems and Informatics (ICSAI 2012),2012

[18] Ivan Baronak,Ladislav Kockovic *Alternatives of Providing IPTV Using IMS* International Journal of Computers and Technology ISSN: 2277-3061 Volume 3. No. 2, OCT, 2012

[19] K. Kerpez, D. Waring, G. Lapiotis, J. Lyles, and R. Vaidyanathan, *IPTV Service Assurance.* IEEE Communications Magazine, September 2006.

[20] Yuehui Jin, Yidong Cui, Jun Shan, Zhi Jiang, Lunyong Zhang, Hongqi Liu, *An Experimental Study On Measurement and Evaluation of IPTV Video Quality* 3rd IEEE International Conference on Broadband Network and Multimedia Technology, 2010

[21] R. Kooij, K. Ahmed, and K. Brunnstróm, *Perceived quality of channel zapping* IAESTED Int. Conf. Commun. Syst. Netw. (CSN), Aug. 2006

[22] Glenn VanWallendael, Wim Van Lancker, Jan De Cock, Peter Lambert, Jean-FranÃ§ois Macq, and Rik Van deWalle *Fast Channel Switching Based on SVC in IPTV Environments* IEEE TRANSACTIONS ON BROADCASTING, VOL. 58, NO. 1, MARCH 2012

[23] Ted H. Szymanski, Member, IEEE, and Dave Gilbert, *Internet Multicasting of IPTV With Essentially-Zero Delay Jitter* IEEE TRANSACTIONS ON BROADCASTING, VOL. 55, NO. 1, MARCH 2009

[24] Hyunchul Joo, Hwangjun Song, Dai-Boong Lee, and Inkyu Lee *An Effective IPTV Channel Control Algorithm Considering Channel Zapping Time and Network Utilization* IEEE TRANSACTIONS ON BROADCASTING, VOL. 54, NO. 2, JUNE 2008

[25] Marc Brogle, Dragan Milic, Luca Bettosini, Torsten Braun *A Performance Comparison of Native IP Multicast and IP Multicast Tunneled through a Peer-to-Peer Overlay Network* IEEE, 2009

[26] S. Deering, *Host extensions for IP multicasting* RFC1112, August 1989.

[27] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, *Internet Group Management Protocol, Version 3, RFC3376, October 2002*

[28] ATT U-Verse [Online], Available: (http://www.att.com/u-verse/). (last accessed on 15/01/2013) .

[29] ABI Research[Online], Available: (http://www.abiresearch.com). (last accessed on 15/01/2013)

[30] Video Bitrate Calculator[Online], Available: (http://web.forret.com/tools/videofps.asp). (last accessed on 15/01/2013)

[31] Open IMS core[Online], Available: (http://www.openimscore.org/) (last accessed on 15/01/2013)

[32] Moving Picture Experts Group[Online], Available: (http://en.wikipedia.org/wiki/MovingPicture ExpertsGroup). (last accessed on 15/01/2013)

[33] ITU-T[Online], Available: (http://www.itu.int/en/ITU-T/Pages/default.aspx). (last accessed on 15/01/2013)

[34] Microsoft Media[Online], Available: (http://www.microsoft.com/en-us/mediaplatform/). (last accessed on 15/01/2013)

[35] MPEG-2[Online], Available: (http://en.wikipedia.org/wiki/MPEG-2). (last accessed on 15/01/2013)

[36] MPEG-4[Online], Available: (http://en.wikipedia.org/wiki/MPEG-4). (last accessed on 15/01/2013)

[37] PathLoad[Online], Available: (http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw-est/pathload.html). (last accessed on 15/01/2013)

[38] IGI-PTR[Online], Available: (http://www.cs.cmu.edu/ hnn/igi/). (last accessed on 15/01/2013)

[39] PathChrip[Online], Available: (http://www.spin.rice.edu/Software/pathChirp/). (last accessed on 15/01/2013)

[40] Assolo[Online], Available: (http://netlab-mn.unipv.it/assolo/README.txt). (last accessed on 15/01/2013)

[41] CoolStreaming[Online], Available: (http://www.coolstreaming.us/). (last accessed on 15/01/2013)

[42] PPLive[Online], Available: (http://www.pptv.com/). (last accessed on 15/01/2013)

[43] UUSeee[Online], Available: (http://www.uusee.com/).(last accessed on 15/01/2013)

[44] PPStream[Online], Available: (http://www.pps.tv/). (last accessed on 15/01/2013)

[45] GT-ITM[Online], Available: (http://www.cc.gatech.edu/projects/gtitm/) (last accessed on 15/01/2013)

[46] The Network Simulator - ns-2[Online], Available (http://www.isi.edu/nsnam/ns/) (last accessed on 15/01/2013)

[47] Cesar D. Guerrero, Miguel A. Labrador *Experimental and Analytical Evaluation of Available Bandwidth Estimation Tools* IEEE, 2006

[48] Manish Jain , Constantinos Dovrolis *Pathload: A Measurement Tool for End-to-End Available Bandwidth* In Proceedings of Passive and Active Measurements (PAM) Workshop,2002

[49] Jacob Strauss, Dina Katabi, Frans Kaashoek *A Measurement Study of Available Bandwidth Estimation Tools* IMC03, October, 2003

[50] Yali Liu, Zhengye Liu, Xidong Wu, Jin Wang, Charlie Chen-Yui Yang, *IPTV System Design: An ISP's Perspective,* International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2011

[51] X. Hei, C. , J. Liang, Y. Liu, and K. W. Ross. *A measurement study of a large-scale p2p iptv system,* IEEE Transactions on Multimedia, Dec. 2007.

[52] Zhonghong Ou, Erkki Harjula, Mika Ylianttila *Effects of different churn models on the performance of structured peer-to-peer networks* IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC , 2009

[53] Majed Alhaisoni, Antonio Liotta, Mohammed Ghanbari *Improving P2P Streaming Methods for IPTV* International Journal on Advances in Intelligent Systems, vol 2, year 2009

[54] ITU-T Focus Group[Online], Available: (http://www.itu.int/en/ITU-T/gsi/iptv/Documents/tech/1106-Hanoi-IPTV-BB-Intro-V3.pdf). (last accessed on 15/01/2013)

[55] S. Deering and D. Cheriton, *Multicasting in datagram internetworks and extended lan,* ACM Trans. on Computer Systems, vol. 18, May 1990.

[56] Zhu, Y., Dovrolis, C., and Ammar, M., *Dynamic Overlay Routing Based on Available Bandwidth Estimation: A Simulation Study* Computer Networks Journal,2006.

[57] Tao, S. and Guerin, R., *Application-specific Path Switching: A Case Study for Streaming Video* in Proceedings of ACM International Conference on Multimedia, 2004.

[58] Business-Wire Research and Markets: Digital Media - IPTV and Smart TV Insights - 2012 (http://www.businesswire.com/news/home/20120223005970/en/Research-Markets-Digital-Media—IPTV-Smart,). (last accessed on 15/01/2013)

[59] Glenn VanWallendael, Wim Van Lancker, Jan De Cock, Peter Lambert, Jean-FranÃ§ois Macq, and Rik Van deWalle, *Fast Channel Switching Based on SVCin IPTV Environments,* IEEE TRANSACTIONS ON BROADCASTING, VOL. 58, NO. 1, MARCH 2012.

[60] Heiko Schwarz, Detlev Marpe, Member, IEEE, and Thomas Wiegand, Member, IEEE, *Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 17, NO. 9, SEPTEMBER 2007.

[61] R. Kumar, Y. Liu, and K. Ross, *Stochastic fluid theory for P2P streaming systems,* in Proceedings of IEEE INFOCOM, 2007.

APPENDIX A

GT-ITM CONFIGURATION FILE FORMAT

Type and number of graphs to be generated
ts 10 47
Average number of stub domains for each transit domain
3 0 0
Average number of transit domains
1 20 3 1.0
Average number of nodes in a transit domain
4 20 3 0.6
Average number of nodes in a stub domain
8 10 3 0.42

With this file, GT-ITM generates ten transit-stub graphs, using as random seed 47. Each graph will have three stub domains per transit node (line 2), with no extra transit-stub or stub-stub edges. Next line (line 3) creates one transit domain. Line 4 specifies that transit domains have (on average) four nodes and an edge between each pair of nodes with probability 0.6. The last line says that each stub domain will have (on average) eight nodes, and edge probability 0.42. Resuming this configuration file generates a graph with one transit domain composed of four transit nodes, each one with three stub domains of eighth nodes. The total number of nodes in the graph is 1 Â· 4 Â· (1 + 3 Â· 8) = 100.