IPTV VIDEO STREAMING IN CONTENT DISTRIBUTION NETWORK

Submitted by

Dawood Ahmed

ID: 2013-1-98-001

&

Md. Kaimon Hasan

ID: 2013-1-98-012

A project submitted in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Engineering

Department of Electronics and Communications Engineering

East West University

Dhaka, Bangladesh

April, 2014

Declaration

We, strongly, declare that this project has been done by us and it has not been submitted elsewhere for any degree or diplomas.

Countersigned

Mustafa Mahmud Hussain

Assistant Professor,

Dept. of Electronics and Communications Engineering

East West University

Dhaka, Bangladesh

Signatures of the students

(Dawood Ahmed)

2013-1-98-001

(Md. Kaimon Hasan)

2013-1-98-012

Approval

The project report entitled "IPTV Video Streaming in Content Distribution Network" submitted by Dawood Ahmed, ID No: 2013-1-98-001 & Md. Kaimon Hasan, ID No: 2013-1-98-012, to the Department of Electronics & Communications Engineering, East West University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Engineering and approved as to its style and contents.

Approved by:

(Supervisor)

Mustafa Mahmud Hussain

Assistant Professor

Department of Electronics & Communications Engineering

East West University

Acknowledgement

We would like to take this opportunity to thank our adviser Mustafa Mahmud Hussain , Assistant Professor, Dept. of Electronics and Communications, Engineering, EWU for his guidance, motivation and constant encouragement. We have always been inspired by the simplicity of his logic, advice and supervision throughout the entire course of this work.

Also we are very much grateful to Dr. Gurudas Mandal Chairperson & Associate Professor, Dept. of Electronics and Communications Engineering, EWU for allowing his support and all the facilities to complete the thesis work successfully.

Finally, we would like to express our special thanks and appreciations to all the teachers and all other official s of EWU for their supports. Also our special thanks goes to all of them who have always motivated and appreciated our efforts.

Abstract

Internet Protocol Television (IPTV) is regarded as one of the biggest business opportunities in information and telecommunication industry. It is an emerging technology which is getting popular day by day. However, there are some challenges to deploy successfully the services in terms of technologies, management, and regulation policy. Among the problems, particularly, how to distribute IPTV content efficiently is one of critical problems because traffic overflow could not be dealt with existing IP network capacity when the number of IPTV services and content increase enormously in the near future. It is also important maintain good quality of service. Content Distribution/Content Delivery Network (CDN) is considered as one solution.CDN replicates the same content from the origin server and distributes them to replica servers which are close to end users. IPTV services can be provided fast and reliably using CDN. The main contributions of this thesis includes Study on IPTV, Video streaming and CDN network and including type of service of IPTV. Protocols and encoding for IPTV video streaming. Overview on CDN network and how IPTV can be integrated in CDN. We have also compared CDN with other alternative topologies for IPTV video streaming, network design for practical Implementation and finally the market analysis for CDN network.

Table of Contents

Chapter 2: Overview of Content Delivery Network

Chapter 3: System Architecture and Implementation

Chapter 4: Market

Chapter 5: Conclusion and Future work

List of Figures

Chapter-1

Chapter-2

Chapter-3

Chapter-4

CHAPTER: 1

INTRODUCTION

1.1Introduction:

Recently, Internet Protocol Television (IPTV) is regarded as one of the biggest business opportunities in information and telecommunication industry. However, there are some challenges to deploy successfully the services in terms of technologies, management, and regulation policy. Among the problems, particularly, how to distribute IPTV content efficiently is one of critical problems because traffic overflow could not be dealt with existing IP network capacity when the number of IPTV services and content increase enormously in the near future. In addition, IPTV deployments have higher required quality of services than those of video over the public Internet in terms of latency and packet losses .Content Delivery Network (CDN) is considered as one solution. It replicates the same content from the origin server and distributes them to replica servers which are close to end users. Thus, IPTV services can be provided fast and reliably using CDN. However, telecommunication corporations hesitate about Capital Expenditure (CAPEX) on large scale due to uncertain growth by market saturation.

1.2 Internet Protocol TV (IPTV):

IPTV (Internet Protocol TV) means television that is delivered over IP (Internet Protocol). Digital television service delivered via a broadband IP link using data communications. In this section, IPTV services overview is shown, and then IPTV service architecture focused on content transmission is described.

1.3 Overview of IPTV Services:

According to ITU-T FG IPTV, IPTV is defined as multimedia services such as television/video/audio/text/graphics/data delivered over IP based networks managed to provide the required level of Quality of Service (QoS)/Quality of Experience (QoE), security, interactivity and reliability. Figure 1 shows a conceptual architecture of IPTV.

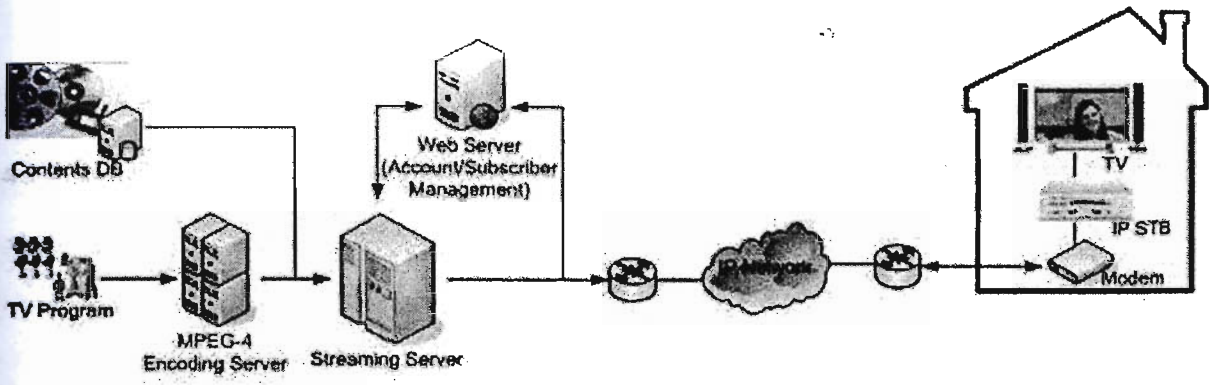


Figure 1: IPTV Conceptual Architecture

IPTV services have both service characteristics of communications and broadcasting due to one of communications-broadcasting convergence services. IPTV services can be classified into 4 major categories: content services, interactive services, commerce services, and communication services.

Content services: These services include not only functionalities to access, and delivery the content, but also functionality to control the delivered content. The content may be broadcasting, multi-media on demand (Video/Audio/Music), and downloadable multi-media content etc.

Interactive services: These services mean the individual service having the independent relation between customer device and the service server. These services may include T-information, T-entertainment, and T-government and so on.

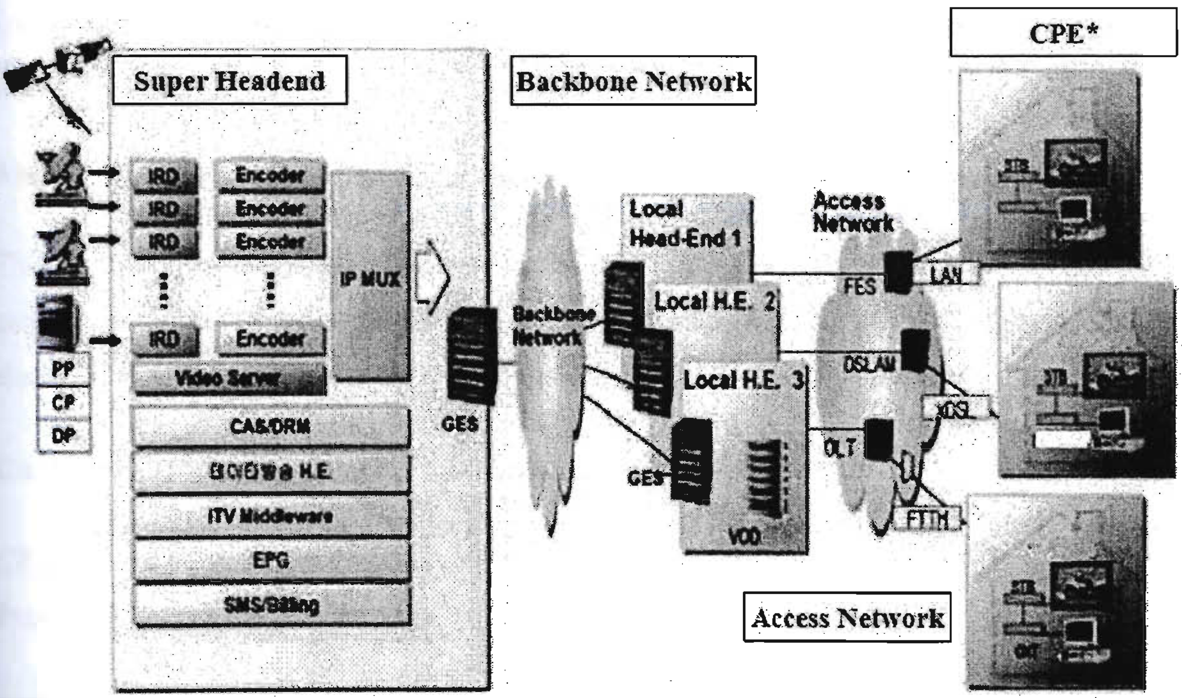
Commerce services: these services provide the user with the interactive marketplace. Also they require high level of security and reliability, by the nature of services.

Communication services: These services provide users the functionality to communicate with others, such as Multimedia Message Service (MMS), e-mail, Voice over IP (VoIP) etc.

Particularly, this study focuses on only Multi-media on Demand of content services which are main killer application of IPTV.

1.4 IPTV network Architecture:

As shown in Figure 2, IPTV network architecture consists of 4 parts: IPTV Super Headend, Backbone core network, Access network, and Customer Premise Equipment (CPE). When the demand of service increases enormously at peak time, critical traffic congestion occurs under backbone core network and it reduce reliability of service and Video server could be broken down. Also, in access network, bottleneck can be happened because of insufficient bandwidth capacities. At this time, reducing video server load by sharing the load with other servers (i.e. CDN Distribution servers) could solve the above problems. That is, the objective to make operation of IPTV services efficient establishing CDN servers in IPTV transport network. Following are brief description of four subsystem of IPTV network.



*CPE: Customer Premise Equipment

Figure 2: IPTV Network Architecture

IPTV Super Headend:

It is the heart of new IPTV Service. Super Headend is responsible for the acquisition, management, encoding, origination and satellite distribution of IPTV content. That is, the IPTV Super Headend aggregates live national content, processes it, encodes the content in MPEG-4/AVC (H.264), and distributes it through an IP core network to religion server.

Backbone network:

Backbone network can be separated into 2 sub networks such as core network and aggregation distribution network. Core network interconnects IPTV Super Headend in national scope to Local Headend in regional scope. Aggregation distribution network is ranged from Local Headend to Video Switching Office (VSO). Backbone network interconnected by dedicated fiber line. Also it is responsible for multicast routing and streaming, QoS under MPLS and Differentiated services.

Access network:

This network is ranged from VSO to subscriber's network. It is responsible for switched channel and QoS per subscriber. It requires the placement of a significant number of new network elements to support the additional bandwidth demands of the IPTV application.

Customer Premise Equipment (CPE):

CPE indicates IP-Set top box including decoder, middleware, and Internet Group Management Protocol IGMP. STB will be extended for mobile IPTV service.

IPTV includes services such as video on demand (VOD), voice over IP (VoIP) and web access services, commonly called Triple Play services. The IP network for IPTV might be a public IP network such as the Internet or private data network such as a LAN-based network. Since IPTV is not a standard, most subscribers are using propriety standards instead of global standards. Therefore, different operators implement IPTV networks in different ways which can create problems during updates and interoperability. An IPTV network should provide the same services and content as cable networks, including broadcast television, channels selection, and anything which can be sent from a single source to many subscribers. There are different types of IPTV, such as Telco IPTV and local or building IPTV. Telco IPTV is designed to deliver TV programs over Telecommunications providers such as the Internet, phone and cable using IP suits. The TV programs are delivered over an IP network at the same time as the Internet or phone services are delivered. Moreover,, the video contents is delivered through the same wire as the TELCO internet service executes the Quality of Service mechanisms on their network to make sure that the delivery of the transmission is reliable. Local IPTV is designed to deliver television programs and video services across buildings and campuses over a Local Area Network (LAN). The video content is directly delivered into the building Local Area Network and the user can watch the channels on a TV or computer screen over the building LAN network. IPTV has the following facilities:

With IPTV, it is possible to pause or stop live transmission. Therefore, the viewer can watch it later without missing any part of the TV program. IPTV records programs that can be played at any suitable time. The user can also record multiple programs at the same time with IPTV system.

IPTV provides the best quality of sound and picture if the Internet connection is fast. Live and prerecorded broadcasting TV channels can be watched through IPTV. Therefore, a viewer does not have to miss any of his/her favorite show or a TV program running on the available channels. Since IPTV service is delivered over the Internet, the system is not affected by weather changes. Therefore, the system can provide normal transmission while the weather is rainy, cloudy or windy.

1.5 Protocols used for IPTV Streaming:

RTP: Real-time Transport Protocol provides packet format for delivering audio and video contents over IP networks. The protocol is used for streaming channels controlled by Real Time Streaming Protocol (RSTP). RTP is commonly used in some streaming media services including telephony and video teleconference applications services. Real-time Transport protocol and RTP Control Protocol (RTCP) work together. In this case, RTP delivers the media streams like audio and video over IP networks, while RTCP refers to monitor transmission statistics and quality of service (QoS) and helps to synchronize multiple streams.

RTSP: Real Time Streaming Protocol or controlling network protocol is used to control streaming media servers. Therefore, RSTP controls the delivery of data while RTP streams the channels. This protocol is used to establish and control media sessions from the server to the user. In order to start and stop data transmission, the media servers issue play and pause commands to easily control the transmission of media files from the server. RTSP is used to control unicast and multicast streams. Streaming of data in RTSP is one-directional. It means that data streams can be sent from the server to the user.

PIM: Protocol-Independent Multicast (PIM) is a set of multicast routing protocols that provide data distribution over IP networks such as the Internet, Wide Area Network (WAN) and Local Area Network (LAN). As it is obvious from its name, protocol independent, PIM is independent and it uses routing information provided by various routing protocols such as BGP or Border Gateway Protocol. In IPTV, Protocol Independent Multicast (PIM) is commonly used to route IPTV multicast streams between networks.

IGMP: The Internet Group Management Protocol (IGMP) is used to manage membership in IP multicast groups. The protocol is widely used in online streaming video and gaming. In IPTV, the Internet Group Management Protocol is a main part of the multicast specification over IP network. Therefore, the protocol is used by IPTV in order to connect to a TV channel and to change from one TV channel to another.

Transport layers protocols include TCP and UDP . Notable differences between the two protocols are highlighted below:

TCP:

Reliable with two way communication with FEC. Data flow is controlled to manage the download rate allowing increasing of the packet rate until congestion is encountered at which point data rate is aggressively slowed down. Allows repeat request for lost or heavily delayed packets.

UDP:

Unreliable, one way packet delivery with no flow control or support for repeat requests. Packets are smaller and thus more efficient. Basic error detection. Packet delivery should exceed encoded stream rate as there is no rate transfer control (for optimum use of the transmission channel) in the UDP protocol.

HTTP:

Hyper text transfer protocol typically used for progressive download. The principal network layer communication protocol (as with all other internet applications) is IP. For RTP/UDP streaming, each IP packet is made up of a header (40 bytes) and a payload.

1.6 Video Streaming:

Video streaming refers to playing video from a source without having to have or download the entire video beforehand. When compared to stored video, streaming video presents a number of unique challenges and is fundamentally different (for example, video is only seekable in one direction). The unique nature of streamed video results in a slightly different encoding method and places greater emphasis on the ratio between encoding speed and output quality of encoded video as well as a number of other trad-offs. This section introduces as well as some basic concepts, encoding considerations and stream deployment scenarios.

1.7 Basic Concepts:

Streaming systems are designed with the eventual implementation network in mind. These systems are designed based on a number of matrics , the most important being available bandwidth for stream utilization. Packet latency, packet loss rate and protocol overhead/efficiency are also important design considerations.

Traditional unicast streaming systems are designed based on the following calculation required server bandwidth:

RSB=Streaming bit rate * Number of clients

For example: RSB=512Kbps*200=100 Mbps to provide each client with an identical stream. It can be seen that a unicast streaming server typically reaches the upper limit in terms of available outgoing bandwidth relatively quickly for a typical internet connection. Packet latency (due to congestion) is key factor due to the effect it has one stream reconstruction and synchronization between clients and the server (especially when the packets are dropped).

1.8 Types of streaming:

There are two distinct ways of implementing video streaming. The first option is progressive download (or pseudo streaming) where is stream is downloaded (not necessarily in play order) into a buffer from beginning to end. Once the buffer is sufficiently full, playback of the stream can begin. The progressive download method has the following characteristics:

- Suitable for small files only
- For longer streams, average data download rate must match or exceed encoding bit rate.
- Uses TCP/IP for data transfer (reliable, retransmission will be sent if required)
- Simple to implement (the streaming server can be a regular HTTP web server)
- Playback only begins when a certain amount of data has been successfully received.
- Streaming media saved to a file on the viewer's PC.

The second approach is known as traditional or regular video streaming. This approach is more suited for streams with a longer lifetime. Typical characteristics of a live stream include:

- Any length stream can be supported
- Higher implementation complexity since a specialized streaming server is required. UDP/IP is used for data transfer (unreliable, no retransmission).
- Playback begins immediately with minimal buffering.
- Streamed data is played back and subsequently discarded (thus not stord to file).

1.9 Encoding for streaming:

Most video coding methods find application in video streaming. However, for optimized video streaming several modifications to the encoding process can be integrated.

Packet loss can often disrupt video streams by degrading I-frames and thus a batch of P-frames and often B-frames. One small error becomes a significantly large disruption to the video stream. Video coding needs to be carried out in such way that the effect of such packet loss are minimized.

For video streaming it is essential to support some form of buffering to compensate for an unstable delivery platform. In order to function with a buffer, the GOP and video chunk sizes need to be significantly small. Each chunk (segment of video of a pre-defined size) that is streamed need to be playable as soon as possible without relying on a large amount of subsequent data. Management of the minimum video chunk size is therefore critical consideration in video streaming.

In order to maintain a constant video stream, video needs to be encoded to be adaptable to the delivery system's available bandwidth. This can achieved by encoding at the minimum bit rate or using a coding system that supports dynamic control of the bit rate (such as layered coding). Real time encoding for streaming ideally requires a feedback path as well as a bandwidth sensing system such that small changes in available bandwidth can be accounted for with compensating bit rate changes during adaptive real time encoding.

Reassembly of the received video chunks should be very simple. There should be little or no dependency between video chunks so that while a given video chunk or set thereof is being played, subsequent chunks can be prepared (including error correction or data re-transmission if required) for playback. The process of segmenting initial encoding video (and reassembling before playback) should be computationally simple to avoid any additional processing complexity during both encoding and decoding.

1.10 Real-time considerations:

Encoding of video data in real time is subject to two main factors: encoding complexity and available processing resources. Real time encoding is highly dependent on the implementation platform and available processing power. In general, various adjustments can be made to encoding algorithms to lessen processing load (at the expense of video quality) if required.

Adjustments that can be made for faster encoding speed include actively encoding fewer B-frames and more I-frames (and consequently increasing the overall bit rate requirement).

H.264/AVC has been developed with support for real time applications (such as TV broadcasting) and is designed to fully support real time decoding. Several enhancements can be turned off to enable support in lower grade decoding hardware. The CABAC entropy coding components in H.264/AVC continually updates the statistics of incoming data to the encoder and adaptively adjusts the algorithm for batter performance using a process known as context modeling.

CHAPTER: 2

Overview of Content Distribution Network

2.1 Overview of Content Distribution Network:

A content delivery network or content distribution network (CDN) is a large distributed system of servers deployed in multiple data centers across the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance. CDNs serve a large fraction of the Internet content today, including web objects (text, graphics and scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks.

A CDN operator gets paid by content providers such as media companies and e-commerce vendors for delivering their content to their audience of end-users. In turn, a CDN pays ISPs, carriers, and network operators for hosting its servers in their data centers. Besides better performance and availability, CDNs also offload the traffic served directly from the content provider's origin infrastructure, resulting in possible cost savings for the content provider. In addition, CDNs provide the content provider a degree of protection from DoS attacks by using their large distributed server infrastructure to absorb the attack traffic. While most early CDNs served content using dedicated servers owned and operated by the CDN, there is a recent trend to use a hybrid model that uses P2P technology. In the hybrid model, content is served using both dedicated servers and other peer-user-owned computers as applicable.

Due to the explosive growth of web traffic in the Internet, the efficient distribution of stored content has become a major concern among researchers and in industry. During the 1990s, the focus was on the design and implementation of caching strategies for traditional web content. Considered one of the main techniques to reduce network traffic as well as the latency perceived by the clients, proxy caching, where popular content available at an origin server is independently stored at proxy servers (also called local servers since they are located closer to the clients) has been a very active research area. The key challenge behind the design of caching strategies is how to efficiently determine which content should be replicated at the proxies in order to achieve a predefined goal such as to reduce client latency or the traffic in the Internet backbone. Recently, the Internet has witnessed the emergence of a number of content distribution companies, such as Akamai and Volera , which work directly with the content providers in order to improve the delivery service and ultimately the revenue accrued by the company. The organization and terminology associated with a CDN is illustrated in Figure 3. The CDN consists of an origin server, or simply the origin, and a number of proxy servers. Popular content may be (partially) replicated at the proxy servers, which are located closer to some of the client populations. A proxy server may serve a single university campus, a metropolitan area or a wider region involving several states or countries. We use the term "local" network to represent the network where a proxy server and its clients are located. We point out that a "local" network may span a large region depending on the dispersion of the clients that are served by the proxy server. Similarly, the term "remote" network is used to represent the network that connects the "local"

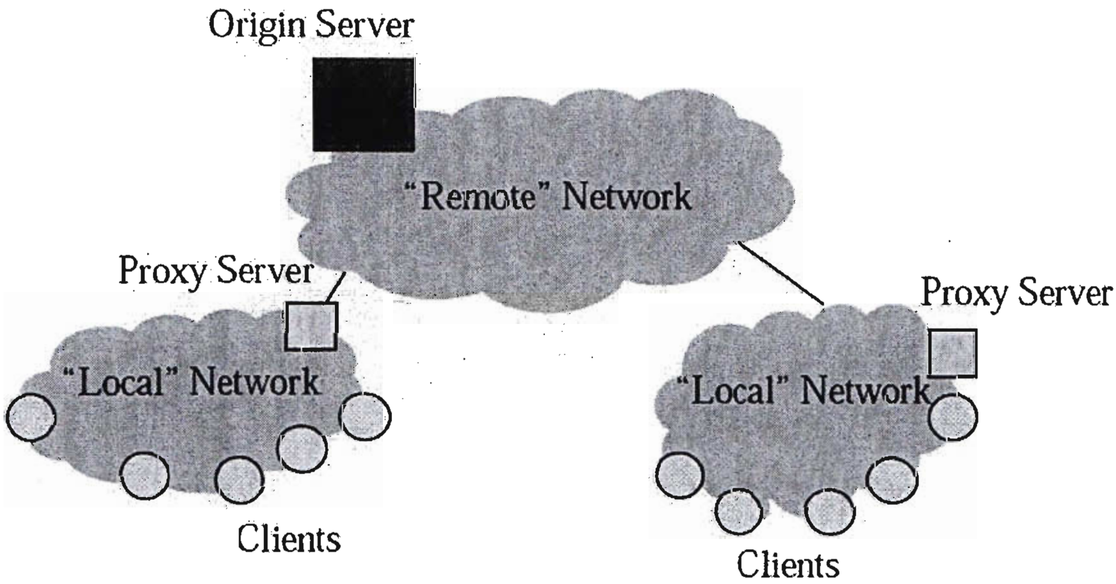


Figure 3: Example of a Content Distribution Network

networks where proxy servers and clients are located to the origin server (e.g., the Internet backbone). In hierarchical CDN architectures, the client sends a request to one of the proxy servers (typically the closest one). If the requested content is not (fully) stored at the proxy, the proxy server forwards the request to the origin server. The origin server either returns the content directly to the client, or, more commonly, sends it to the proxy, which then forwards it to the client. A hierarchical CDN may include multiple levels of proxy servers where requests that miss

at a lower level proxy are forwarded to a proxy server at a higher level in the hierarchy. The origin server is contacted only if the requested content is not found in any of the proxies. A single level hierarchy as shown in Figure 3, which is the most common CDN model. However, we point out that the methods obtained can be extended to CDNs with multiple levels of hierarchy by considering that each proxy at level i is a potential client of a proxy at level i +1 of the hierarchy. A different type of CDN architecture, not addressed in this thesis, is a peer to peer architecture where, when a server receives a request that it cannot (fully)
handle, it searches for another server (any other server) to handle the portion of the request that missed locally. Key questions behind a CDN design are: a) which content should be replicated or cached at the proxy servers, b) how many replicas and where each replica server should be placed in the network, c) which server a client's requests should be sent to and d) how to route the content from
the selected server to the client. Thus, designing a CDN involves the development not only of methods for determining server content (or caching algorithms), but also of methods for determining the placement of servers in the network, server selection and routing. Furthermore, another design problem is to determine the times at which the CDN configuration (e.g., server content and routing) is updated in response to changes in the workload, particularly, in the client request rates. Thus, the design of a CDN should be driven by an understanding of key characteristics of the workload, such as client request rates, object popularity and how these characteristics change over time. These questions are addressed with a predefined performance or cost goal.

A CDN replicates the same content or services from the origin server to replica servers scattered over the globe in order to prevent heavy traffic load on origin server and delivery content to end-users reliably and timely. Thus, Usage of CDN provides several advantages such as reduced origin server load, reduced delay time for end-users, efficient usage of bandwidth, and improvement of scalability through content replication. Due to these benefits, CDN is considered as one of solution to handle traffic congestion and guarantee reliability in IPTV network. Figure 4 shows traditional internet transmission. When end user requires Content Provider (CP) server to transmit a content, the content is transmitted through several networks to end user. On the other hand, in CDN, CP server notifies the location of the closest surrogate server which has the content, instead of providing content itself. Then, end user can receive the content fast and reliably from the surrogate server. Figure 5 represents content transmission in CDN.

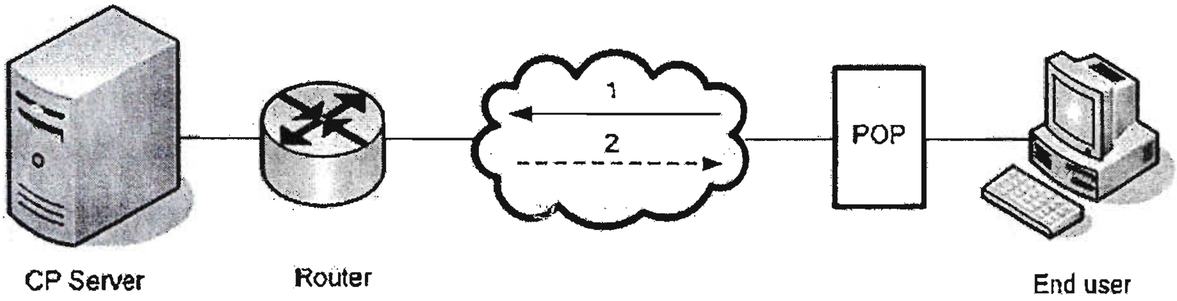


Figure 4: Transmission in Traditional Internet Network

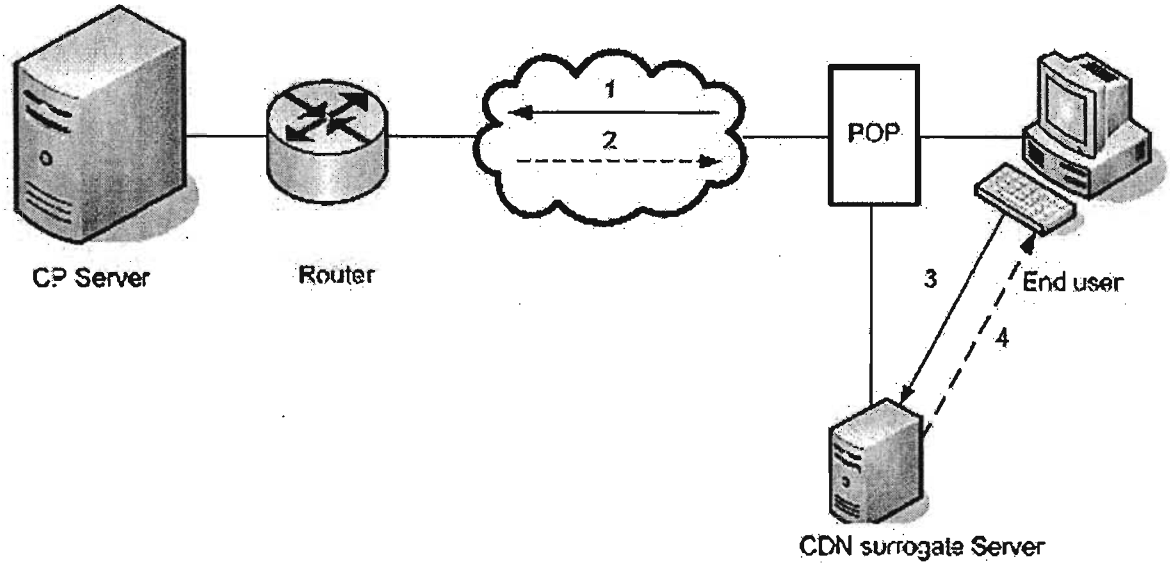


Figure 5: Transmission in Content Distribution Network

2.2 The Special Characteristics of Streaming Media Content:

Motivated by the increasing availability of media content in the Internet, improvements of network bandwidth in the Internet backbone and the availability of faster "last mile" connections, such as cable modems and DSL (Digital Subscriber Lines) services, users are becoming increasingly interested in watching movies, listening to radio or music or viewing lectures over the Internet.

Consequently, streaming media content (i.e., audio and video) is becoming a significant fraction of the total traffic in the Internet.

Streaming media objects and workloads have special characteristics, which distinguish them from traditional web content (i.e., small HTML and image files). These characteristics are: (1) large size, (2) requirement of sustained disk and network bandwidth, (3) possibly unequal access to different portions of the files and (4) several new tradeoffs introduced by the shared delivery of objects if *multicast* delivery is used. The implications of these characteristics on the design of efficient CDNs for streaming media objects are described below.

2.3 Media File Size and Bandwidth Requirements:

A media file may require tens of megabytes to several gigabytes of storage depending on its quality and duration. This implies that most of the cached content must be stored on disk. Furthermore, storing a new media file on the disk requires a significant amount of disk bandwidth, sustained, typically, over a long period. Hence, not only the disk space, but also disk bandwidth and i/o network bandwidth must be carefully managed when designing a streaming CDN. In particular, it

may be important to avoid allocating disk bandwidth to store a requested object in the cache if that object will not be requested again soon. This is particularly true because that bandwidth could otherwise be used to serve client requests for other objects already stored in the cache. Depending on the workload, this may be an important factor to consider in the design of caching strategies for

streaming content. In particular, it motivates the development of new algorithms that do not cache objects without prior knowledge of its client access rates. This implies a significant deviation from traditional web caching algorithms where space is the only resource that is managed and that, typically, write a new object into the cache every time a request misses in the cache.

2.4 Reasons for selecting CDN among alternative network topologies:

In this section, reasons for selecting CDN among alternative network topologies are described. Generally, nowadays 3 alternatives for IPTV network topology are considered. First alternative is a router based topology which provides multicast and unicast services using the existing IP network transport system. If the path and traffic pattern of the network are sufficiently analyzed, this method is expected to perform well, otherwise, router based topologies could cause bottleneck and congestion while unexpected traffic overflow is occurred. In other words, router based topology has low scalability because all traffics converge on specific node, router, regardless of traffic volume. As another alternative, peer to peer topology is considered. The concept of peer to peer method is moving the functionality from single equipment, router to end systems of application level. This method can show theoretically least establishment cost and high efficiency and high scalability but it is under experiment stage and the performance does not be approved. Also, from operators perspective, peer to peer cannot give attractive biz models because network operators will play a role of only pipe line provider under peer to peer network. In the middle of router based and peer to peer, last alternative, CDN topology exists. The concepts of CDN method is making infrastructure-centric architecture which provide value added services at strategic locations. It is different from router based topology in that CDN is additional infrastructure connected by overlay networking. As previously mentioned, CDN balances Server loads, and it guarantees high reliability and scalability. Also network operators can develop various biz models through AAA (Authorization, Accounting, Authentication) function which CDN can give. These are reasons that recently CDN is considered as a practical alternative for IPTV services from network operators perspective. The other side, the weakness of CDN is expensive establishment cost. Thus, cost effective establishment has become a hot issue. Table 2.4.1 compares 3 alternative network topologies for IPTV service deployment.

	Router based	CDN	Peer to Peer
Concepts	– Using Existing IP network based multicast/unicast	– Infrastructure-centric architecture which provide value added services at strategic locations	– Moving functionality from single equipment to application layer level end system
Merits	– Efficient work in well-known network	– Server Load Balancing – Reliability – Scalability – AAA function	– Low establish cost – High Efficiency – High Scalability
Demerits	– Best effort service (Bottleneck, Congestion) – Limited deployment – Low scalability	– Expensive Setup cost	– Few Biz model to operator – Security – Experimental stage

Table 1: Comparisons among alternatives for IPTV network topology

2.5 Difference between p2p and CDN:

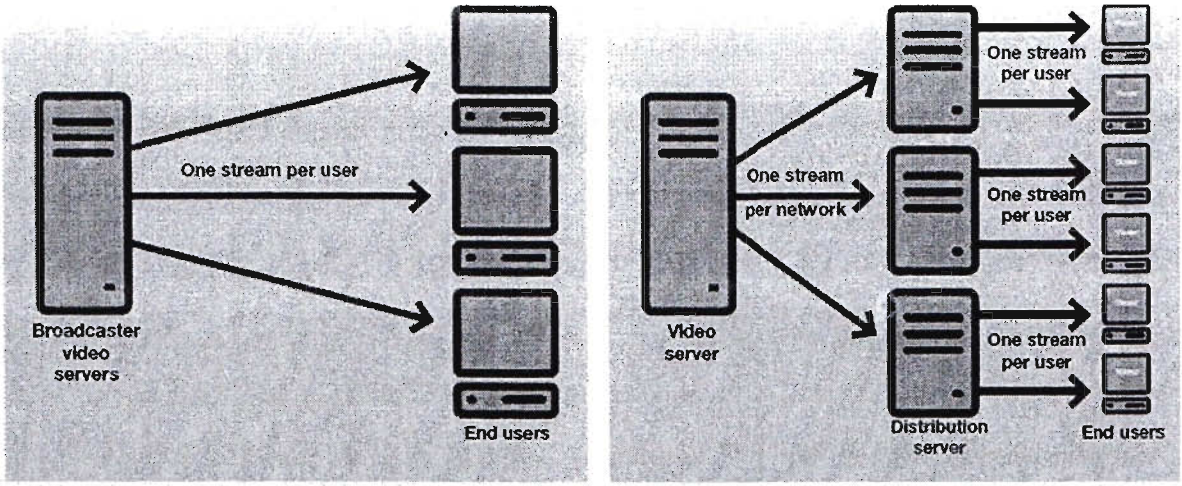
Both peer-to-peer (P2P) and CDN are overlay networks. This is the similarity of them. CDN is a group of servers, or data centers, located in several geographical areas to serve end-users.CDN helps deliver contents to end-user faster. The "content" can be web, video or application. Use web as an example. End-users send their request to web server, and web server redirect this request to CDN. Then CDN makes a decision to choose which server to use to serves this user. One simple way is to choose a server that's geographically close to the user. But usually more complicated scheme are used to make sure the content is available, and can be send to user quickly. P2P is used in file sharing, and content distribution also. In CDN, the traffic is from CDN servers to users (usually user's request goes to web server instead of CDN), while in P2P, every peer is both server and client - that's why it's called peer instead of client or server. So apparently the traffics are different in those two type of networks. Also P2P network can be completely ad hoc, in the sense there's no central server to help find where the desired content is. Such decentralized design is very unlikely to see in CDN since CDN has to promise contents availability. P2P as a content delivery tech is used in some CDNs. A core tech used in P2P, Distributed Hash Table, is also used in some CDNs to help manage, discover and index contents more efficiently. Some CDNs even use P2P to allow end-users to serve end-users, to reduce the load on CDN itself.

2.6 Bypassing the bottlenecks:

The vast majority of transmissions on the Internet are unicast – that is, communication between a single sender and a single receiver. Using unicast for video content consumes a large amount of bandwidth, not only because of the volume of the transmitted data but also because of the number of receivers. This can cause severe network congestion, which spoils content quality. In addition, using a separate stream for each user is very expensive because it requires high server capacity and results in high bandwidth costs.

Multicast – communication between a single sender and multiple receivers – is sometimes used to circumvent the limitations of unicast. This involves streaming content to several distribution servers, usually operated by third-party content distributors. These, in turn, stream the content on to users. Multicast consumes less bandwidth but the consumption is still very high. The multiple distribution servers are also expensive. Multicast transmission has better scalability than unicast, but is still very limited in this respect.

From a content provider's perspective, multicast has serious shortcomings. For example, it allows only live streams, not video on demand (VoD). Few Internet service providers (ISPs) have multicast-enabled networks (meaning that every router must support the multicasting part of the Internet Protocol). A network of multicast-enabled networks, linked together to form a more efficient Internet, does not exist. Therefore, multicast does not enable content providers to guarantee quality of experience (QoE) for users worldwide.



In unicast video distribution, a single video server provides a separate video stream for each end-user. This consumes a lot of bandwidth, and a lot of server capacity per end-user.

Multicast video distribution places a layer of distribution servers between the main video server and the end-users. This uses less bandwidth and has greater scalability, but still suffers from shortcomings in terms of service flexibility, coverage and quality of experience.

Figure 6: Unicast and Multicast video distribution

2.7 Types of CDN:

There's not really a generic CDN, but for our purposes, they fall into three basic categories: those that deliver general content including, but not constrained to, video delivery; those that deliver on-demand or pre-recorded/pre-encoded video; and those that deliver live video.

All the technologies described below, from full download to progressive download to streaming to HTTP delivery, are in use today—in a variety of business models—although some have proven more popular than others.

2.7.1 General Purpose CDNs:

The growth of CDNs started before video, and chances are you've used their services if you've downloaded a software update, bought a song on the iTunes Store, or even just visited a popular website.

These general-purpose CDNs perform what's most frequently referred to as web acceleration. This is generally accomplished best by a CDN that has a number of servers in many locations, ideally close to the large connection points between internet service providers (ISPs) or even within the same data centers as a popular website or gaming/application provider. The CDN caches content, storing a copy of content that will frequently be requested by a high number of internet users.

To understand web acceleration, think about it in terms of the marketing pitch that used to accompany those CDs we got in the mail for a certain dial-up services. The marketing pitch offered potential users the ability to "turbo boost" of their web browsing experience, at much faster speeds than their 56Kpbs modem would allow.

Accomplishing this was fairly simple, since every internet-bound user of the dial-up service provider needed to dial in to one of its points of presence (POPs). When a particular piece of content was request multiple times, the service provider would cache that content at one, several, or all of its POPs, meaning that any request for that content would never go out across the costly internet network lines but would instead be served up from within the service provider's own data cache.

Today, however, web acceleration is a bit more complex, as even the most popular ISP in the United States only provides internet connectivity to only about 22% of the total U.S. online population. The CDN delivering content that can be web accelerated needs to cache content at many locations, resulting in a need for both a national footprint as well as a large quantity of servers at each the majority of US data centers and ISP headends / POPs.

The general-purpose CDN market is highly fragmented outside of the United States, falling along language lines and country boundaries, except in a few parts of Europe and Asia. Even in those markets, the language clusters in a single country can lead to multiple CDNs being able to make inroads into various parts of the market. India is a good example of this, with more than thirteen different distinct official languages, depending on the given state or municipality.

2.7.2 On-Demand Video CDNs:

Some general-purpose CDNs also provide CDN services for on-demand video content. The thinking is that video content is just a large file, like a game or large application download, so the serving of video content should not be much different than other content.

A few years ago, the disconnect between general-purpose and on-demand-video CDNs was quite distinct, as video delivery required the use of a streaming server.

Streaming servers deliver the content at the time of a request, but only deliver the bits requested rather than the entire length of the video clip. This was helpful for the content owner who was paying the CDN for delivery by the bit, since a viewer choosing to abandon viewing of a clip halfway through its duration would not have downloaded the entire clip, regardless of the viewer's internet connection speed.

CDNs didn't necessarily want to spend the extra dollars for hardware and streaming server software, just to limit the number of bits being delivered to the viewer, so a few other ideas were tried.

One was the direct download, where the entire video clip needed to be downloaded to a viewer's computer before it could be viewed. While this is acceptable for downloading a game or computer application, the tolerance level of viewers waiting for content to download—especially something such as a movie, which could be over 1GB in size—was low. No one wanted to wait, upon selecting a movie to view, twenty minutes or more to view the content.

Another option was the progressive download, which is used by a number of online video sites, such as YouTube. In this version, the CDN begins delivering the download, but a viewer can begin watching content within the first 3-5 seconds, on the assumption that the viewer's fast internet connection will continue to download the video clip at a fast enough pace to avoid running out of viewable content. As the speed of internet connections outpaced the bitrates used for standard-definition content, though, many viewers would have the entire video clip downloaded before they were even halfway through viewing the content.

Abandoning the video clip partway through still meant that the content owner was paying for full delivery, since the entire clip had been downloaded. This issue, of course, is exactly what a streaming server was designed to avoid, so the hunt was on for a progressive download solution that could throttle content delivery (curtail the speed at which content was downloaded to a level just a bit faster than content would be viewed) while still avoiding the use of a specialized streaming server for on-demand content.

In recent years, the best solution to emerge is called HTTP streaming, and it is coupled very tightly with adaptive bitrate (ABR) encoding and delivery.

Just like it sounds, HTTP streaming uses generic HTTP servers (often based on Apache or Windows Server) to deliver on-demand video files the same way that other HTTP website content—such as images and text files—is delivered. HTTP delivery has had the capability of throttling for a number of years, even before the advent of online video, and these same techniques can be applied to throttle on-demand video clip delivery.

The most interesting addition to the CDN arsenal is adaptive bitrate (ABR), which converts a video stream into fragments or chunks, often 2-10 seconds in length. ABR creates discrete streams at various bitrates and then uses feedback from the internet user's video player to dynamically detect the optimum network speed for delivery of the video clip.

As network conditions change – for better or worse – the stream with the most appropriate bitrate is served for that give chunk of time. As this is a streaming solution, it keeps the content owner happy by limiting the bitrate and delivery of content to just what the viewer will consume, and it benefits the CDN by limiting the number of specialized streaming servers needed for on-demand video delivery.

Not surprisingly, given the benefits of ABR, there are multiple ABR solutions in the marketplace today, including one each from the Big Three – Adobe, Apple, Microsoft – as well as several proprietary solutions, such as Move Networks. All act in a fairly similar manner, and some even allow delivery via HTTP, either HTTP streaming or progressive download.

2.7.3 Live Video CDNs:

Despite the advent of ABR and HTTP streaming, there still is a need for live video delivery, since live video can't be cached like content that's previously recorded.

This area is probably the least mature of all the CDN models, for three reasons.

First, the vast majority of video content delivered by CDNs is on-demand video: some estimates are as high as 95% of all online video being delivered as on-demand video content.

Second, since live video can't be cached, it's necessary to modify the basic CDN infrastructure to have either very high-bandwidth pipes between a central location and the end-user viewing the content or to have slightly lower-bandwidth pipes that send the live stream to a repeater or reflector that is nearer to the end user.

Third, given the two points above, the cost to build out and maintain a live streaming solution for very popular live events is daunting: building and maintaining a million-plus viewer live streaming solution is quite expensive.

Given the fact that actual streaming servers must be used and that bandwidth delivery costs rise dramatically for each additional simultaneous viewer (on-demand content delivery typically sees simultaneous viewer bandwidth at one-tenth the number of live viewers) the live video delivery solution would need to be in almost constant use at its peak viewing level to reach the economy of scale of over-the-air television and traditional cable live video transmission.

That's not to say that live video CDNs aren't viable: the move by over-the-air broadcasters to use part of their digital frequency spectrum to deliver video to mobile televisions or handsets is just another version of the CDN, albeit one that has a sixty-year head start in terms of placing transmission towers and covering wide geographic areas.

The problem with the blockbuster online live event, however, is that it often takes a number of months of planning to increase the size of a live video delivery network to the level of anticipated audience viewership. If successful, and more viewers attempt to watch than the system is built for, there is a risk of technical failure; if fewer viewers choose to watch the live event, there's the risk of financial failure from overbuilding the CDN for a specific event. The challenge is great, but the industry has shown it can overcome large challenges, so expect to see significant movement in this part of the CDN space over the next two years.

2.8 Managerial Issues on CDN:

The management issues on CDN can be separated into 4 categories as following .

---Surrogate placement: It is issues that where CDN server should be located in.

---Content selection and delivery: It is issues that which content should be replicated and how many copies of content are optimal.

---Request redirection: it is issues that when content request occurs, which CDN server delivery the requested content is optimal in terms of global servers load balancing.

---Cache Update: it is issues that how frequent synchronization with origin server should be done. Also it is problem related to the level of update (i.e. removing existing and storing new content)

2.9 Focus on QoE:

2.9.1 Quality of Experience (QoE) is a subjective measure of a user's experiences with content delivery, either with the delivery of a piece of content or with all content delivered by a content provider. Unless a content provider delivers optimal QoE, users will defect to the competition. QoE is a function of two factors: quality of content (QoC) and quality of service (QoS).

2.9.2 Quality of content (QoC) is a user's subjective, often unconscious, appraisal of the attractiveness or importance of a piece of content or of a content provider's entire offering. Is it entertaining, exciting, educational or helpful? Is it well produced, designed and written? Is it easy to find and access?

2.9.3 Quality of service (QoS) is an objective measure of the accuracy of content transport from the content provider, over the Internet, to the user's receiving device. It can be quantified in terms of available bandwidth, latency, packet losses, etc., and used to guarantee a certain level of a specified resource to selected traffic on a network.

If users experience freezing in video playback, color blurring, significant delays for startup or other transmission errors, then they may abandon the service, temporarily or permanently. Quality of content is of little value unless delivered intact to the user. Quality of experience is what counts.

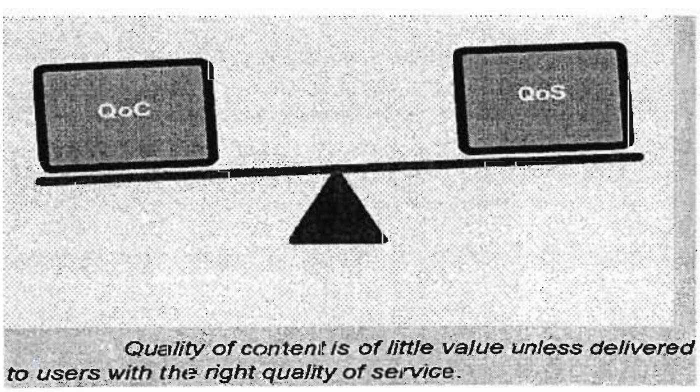


Figure 7: Balance between QoC and QoS

2.10 Latency kills quality content:

Latency refers to user-perceived response time. Slow response is the single greatest contributor to users' abandoning websites and processes. After waiting past a certain attention threshold, users bail out to look for a faster site.

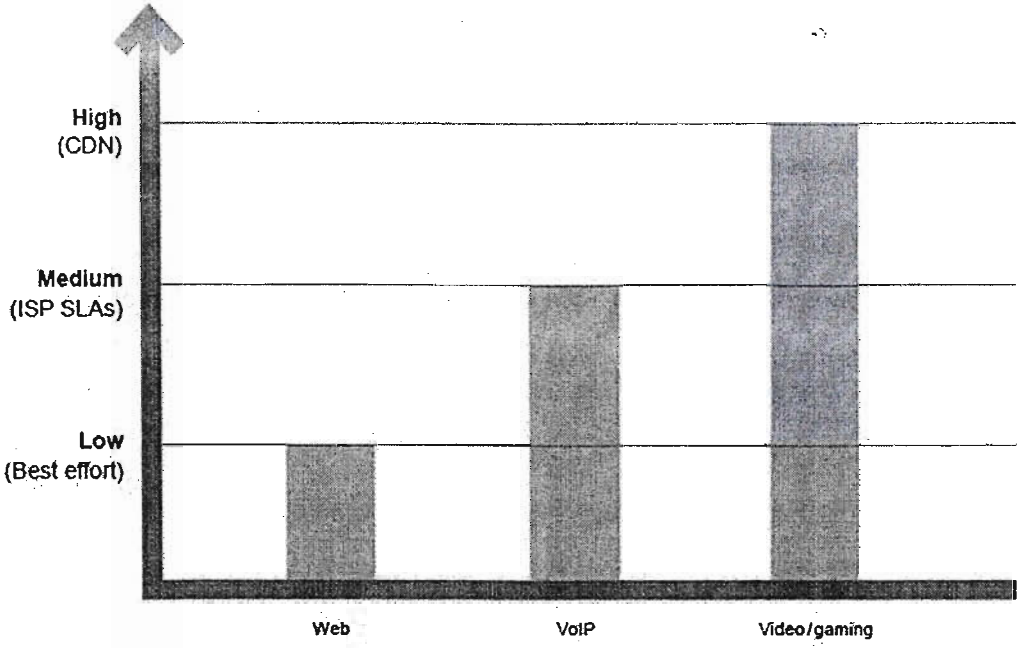
For static web pages, users are unlikely to wait more than four seconds for a page. One second or less is required for users to feel comfortable. For streaming media, latency budgets are much tighter.

Estimates of average Internet latency vary from 125 milliseconds to 500 milliseconds. Internet service providers' (ISPs) service level agreements often stipulate a maximum of 85 milliseconds. This means that most ordinary web applications, including web browsing and email, usually work well over the Internet without the support of a CDN.

Certain applications, such as Voice over IP (VoIP) also work but users may experience minor quality problems. Some applications, such as high-fidelity music broadcasts, live Internet TV and online gaming, do not work well at all over the raw Internet.

Streaming media require high-quality connections. Flaws show up immediately as interruptions or degradations of content quality, resulting in poor user experiences. Online gaming places even stricter demands on Internet connections. Data exchange between gaming servers and users' computers must be exceedingly fast and precise. Even latencies as low as 20 milliseconds may make gaming choppy and unstable, preventing players from performing optimally.

High-quality CDNs make it possible to obtain end-to-end audio and video latency of 5-20 milliseconds, which is similar to the audio time delay between participants in a face-to-face conversation or between musicians on a stage. This may be hard to believe – until you consider that light travels nearly a million times faster than sound (although about 2/3 as fast through glass as through a vacuum).



Different applications demand different latency levels. Only an extremely low latency (5–20 ms) is acceptable to users in video streaming and online gaming.

Figure 8: Different applications demand different latency levels

2.11 System Design Considerations:

streaming CDN architect needs to take into account the following key issues to determine the most cost-effective solution for the four CDN design problems (i.e., server content, server placement, server selection and routing) defined in chapter 1: a) the architecture of the distribution network (e.g., hierarchical or peer to peer), b) the streaming protocol used by each (origin and proxy) server (e.g., unicast, periodic broadcast, bandwidth skimming), c) characteristics of the client workloads such as the client request rates for different objects and how frequently these rates

change and d) the storage capacity, disk bandwidth and network i/o bandwidth available at each server as well as the network bandwidth available on the path(s) between server(s) and clients.

2.12 Further Description of CDN Components:

A CDN consists of a group of clients, a group of (proxy and origin) servers where content is replicated and a network connecting servers and clients. This section elaborates on each of these components and on how they are defined in the context of the solutions developed in this thesis. For instance, our algorithms and CDN design models are solved for a set of client sites, instead of considering each individual client separately, where a client site is defined as a group of clients that are located geographically close to each other and are connected to the same given

access point (e.g., a cable network, a domain in the Internet). In practical terms, this means that our methods consider the statistically aggregated content requests from a group of clients, instead of each individual client request. We consider CDNs where origin and proxy servers use either unicast or a scalable multicast delivery protocol. For scalable delivery, the Hierarchical Streaming Merging Bandwidth Skimming is the streaming protocol assumed, unless otherwise stated. Bandwidth Skimming was chosen because of its scalability properties, superior to other protocols such as patching, and its overall high efficiency for all ranges of request rates. Furthermore, it can be used in interactive environments and has been extended to include techniques for packet loss recovery . Although our strategies assume the basic bandwidth skimming protocol, the approach, main insights and conclusions obtained should also be valid if the reliable bandwidth skimming is used. The underlying multicast protocol used by each server may be either IP multicast, which may become more widely available as new protocols supporting single-source multicast are implemented, or application level multicast client requests are sent to a proxy server, typically the closest one, which may forward them to the origin server, in case the requested content is not (fully) stored locally. We point out that the origin server may stream the content back either to the proxy server, which then forwards it to their clients, or directly to the clients. Appropriate scenarios will be considered in evaluating caching algorithms and designing the minimum delivery optimization cost models. In particular, we point out that it is cost-effective to transmit origin streams through a proxy only if the content is simultaneously being written into the proxy cache or if the origin server uses unicast and the proxy uses multicast, in which case, it results in significant bandwidth cost reductions.

CHAPTER: 3

System Architecture and Implementation

3.1 System Architecture:

System architecture consists of following elements.

3.1.1 Streaming Server:

This is the main server which encodes video from Camera and starts live streaming of a regular course or event.

3.1.2 Content delivery server:

These are content delivery servers which get the direct video stream from streaming server. Regular streaming clients get connected to one of these content distribution servers and get the live stream.

3.1.3 Request Redirection Module:

This is the most important module in which clients streaming requests are redirected to content delivery server depending on the specified redirection policy. This system use different algorithms to control traffic. The goal of these algorithms is to intelligently distribute load and/or maximize the utilization of all servers within the cluster. Some examples of these algorithms include:

Round Robin:

Client requests are redirected to one of the content delivery server in round robin fashion. A round-robin algorithm distributes the load equally to each server, regardless of the current number of connections or the response time or any other parameter. Round-robin is suitable when the servers in the cluster have equal

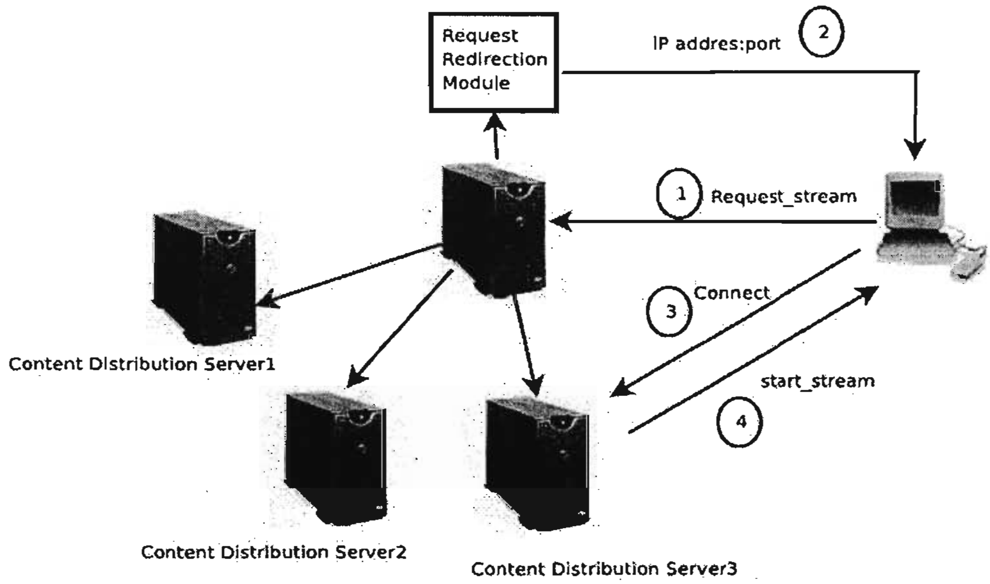


Figure 9: Hierarchical Network Setup

processing capabilities; otherwise, some servers may receive more requests than they can process while others are using only part of their resources. Also as shown in Figure 9 a hierarchical network setup. So client should be redirected to their local (or geographically close network).

Subnet Level Based (2nd Octet Of IP Based):

We can use IP address of the clients to determine from which area they are and according to that we can redirect them to statically allocated server which is local or closer to them. If for clients of an area has the IP address range as 10.8.x.x, and for another area as 10.12.x.x and so on. It can be assumed that same subnet level CDN server can give better performance (low network delay) in streaming video. Also network load will be minimized. Again this technique has disadvantages of being static. As it can happen from some area, number of client requests are higher than the others. In this case a particular server will get most of the requests and if it exceeds its capacity performance will degrade.

Instantaneous Number of Connections:

A least-connection algorithm sends requests to servers in a cluster, based on which server is currently serving the fewest connections. This algorithm will not guarantee the local users will be connecting to local server. Still it will maintain the load balancing at all the servers.

Dynamic Network Utilization Based:

Clients are redirected to a server whose network utilization is least among all servers. It may happen that some servers are running some other applications (or multiple streaming) so only number of connections parameter won't be useful for redirecting and load balancing. Their actual network utilization parameter should be used for fair load balancing. For network utilization we use bandwidth monitor software through which we get 3 sec average of content delivery server's network utilization and we continuously send this average value to main server in 3 sec time interval. If the load on the system are highly dynamic then we can reduce the update interval for better load balancing on the servers.

3.2 Implementation:

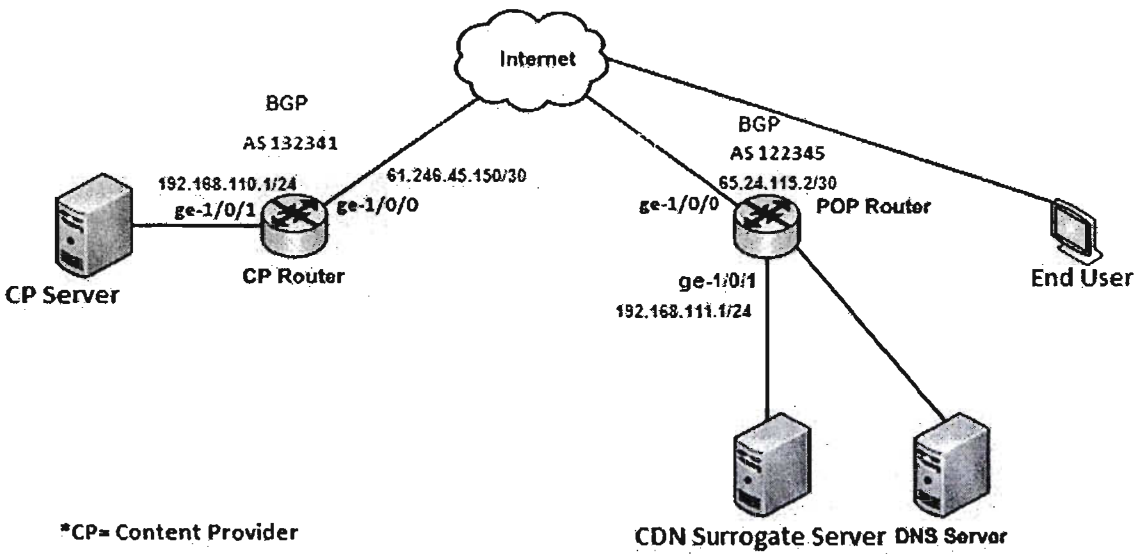


Figure 10: Implementation Network

CP Server:

This is executed on main Streaming server. It listens on TCP port 30000 and accept incoming streaming client requests using VLC player. It uses HTTP streaming protocol. It also contains the Request Redirection module which contains all the algorithm discussed above. Round robin, least-connection based and subnet-based redirection are done by maintaining information of all the distribution servers. Information contains IP address, instantaneous number of connections and instantaneous network utilization.

CP Router:

CP server is connected to one interface of CP router and another interface is connected to ISP. Border Gateway Protocol (BGP) runs on this router for routing.

POP Router:

CDN surrogate server is connected to one interface of POP router and another interface is connected to ISP. Border Gateway Protocol (BGP) also runs on this router for routing.

CDN Surrogate server:

This is executed on distribution server machine which will receive live stream directly from CP server and itself start streaming to other clients. This special
clients can be added or disconnected dynamically.

DNS Server:

DNS server is connected with POP router. It translates domain names to the numerical IP addresses needed for the purpose of locating computer services and device.

End user:

Client machine send request to main server and is redirected to one of the special
clients to receive live streaming. It uses mplayer client to get the live stream from the server.

3.3 Routing Configuration:

CP Router:

```
set interfaces ge-1/0/0 description Connected-CP-to-ISP
set interfaces ge-1/0/0 unit 0 family inet address 61.246.45.150/30
set interfaces ge-1/0/1 description Connected-to-Server
set interfaces ge-1/0/1 unit 0 family inet address 192.168.110.1/24
set routing-options static route 0.0.0.0/0 next-hop 61.246.45.149
set routing-options autonomous-system 132341
set protocols bgp export export-route
set protocols bgp group CP-ISP type external
set protocols bgp group CP-ISP neighbor 61.246.45.149 description "#CP-ISP#"
set protocols bgp group CP-ISP neighbor 61.246.45.149 local-address 61.246.45.150
set protocols bgp group CP-ISP neighbor 61.246.45.149 peer-as 9498
```

POP Router:

```
set interfaces ge-1/0/0 description Connected-POP-to-ISP
set interfaces ge-1/0/0 unit 0 family inet address 65.24.115.2/30
set interfaces ge-1/0/1 description Connected-to-CDN-Server
set interfaces ge-1/0/1 unit 0 family inet address 192.168.111.1/24
set routing-options static route 0.0.0.0/0 next-hop 65.24.115.1
set routing-options autonomous-system 122345
set protocols bgp export export-route
set protocols bgp group CP-ISP type external
set protocols bgp group CP-ISP neighbor 65.24.115.1 description "#POP-ISP#"
set protocols bgp group CP-ISP neighbor 65.24.115.1 local-address 65.24.115.2
set protocols bgp group CP-ISP neighbor 65.24.115.1 peer-as 9285
```

CHAPTER: 4

Market

Market Analysis:

The number of global IPTV subscribers was expected to grow from 28 million in 2009 to 83 million in 2013. Europe and Asia are the leading territories in terms of the over-all number of subscribers. But in terms of service revenues, Europe and North America generate a larger share of global revenue, due to very low average revenue per user (ARPU) in China and India, the fastest growing (and ultimately, the biggest markets) is Asia. The global IPTV market revenues are forecast to grow from US$12 billion in 2009 to US$38 billion in 2013.

Content delivery networks (CDNs) have played a crucial role in the effective operation of the Internet, helping to mitigate problems such as network congestion, packet loss, jitter and delay. Rising numbers of Internet users and increased bandwidth usage have both played a major part in transforming the market for CDN service provision. Just as the CDN landscape has given way to new types of provider – including a diverse range of "pure play" providers and a growing number of network operator CDNs – so too has the ongoing debate about the potential uses of CDNs shifted into new territories, including the delivery of over-the-top (OTT) content, mobile video and cloud services.

There are many ways of measuring the value of the CDN market, and these reflect the diverse range of industry participants. For telecoms service providers and content owners with their own CDN, a CDN's value resides partly in its ability to improve retail service delivery and support their efforts to win and retain customers. For industry manufacturers and providers of managed CDN services, the value of the market is premised on the demand from telecoms operators, content owners and other businesses to have their own CDN. For providers of commercial CDN services, the value of the market depends on the continued need among content owners and online enterprises for a wide range of content management and delivery services. These extend beyond basic per GB "bit delivery" and include premium services such as application acceleration, online security and website optimization.

Finally, for content owners, the value of CDNs lies in the extent to which they provide a reliable and high-quality online experience for the end users of their content. Informa's new CDN forecasts assess the growth potential and specific trends shaping one small, but crucial, part of the burgeoning market for global content delivery. Specifically, they measure the growth of global commercial CDN traffic and revenues through to 2017. By 2017, the market for commercial CDN services is expected to be worth US$4.63 billion, reflecting a threefold increase from 2012. Video will be the largest contributor to commercial CDN revenue growth, accounting for 81% of total revenue in 2017 (fig.4.1).

Although the US will remain the largest market for commercial CDN services, its share of global traffic and revenue will fall. At the same time, China and Western Europe will increase their share of the global "bit delivery" market (fig. 4.2).

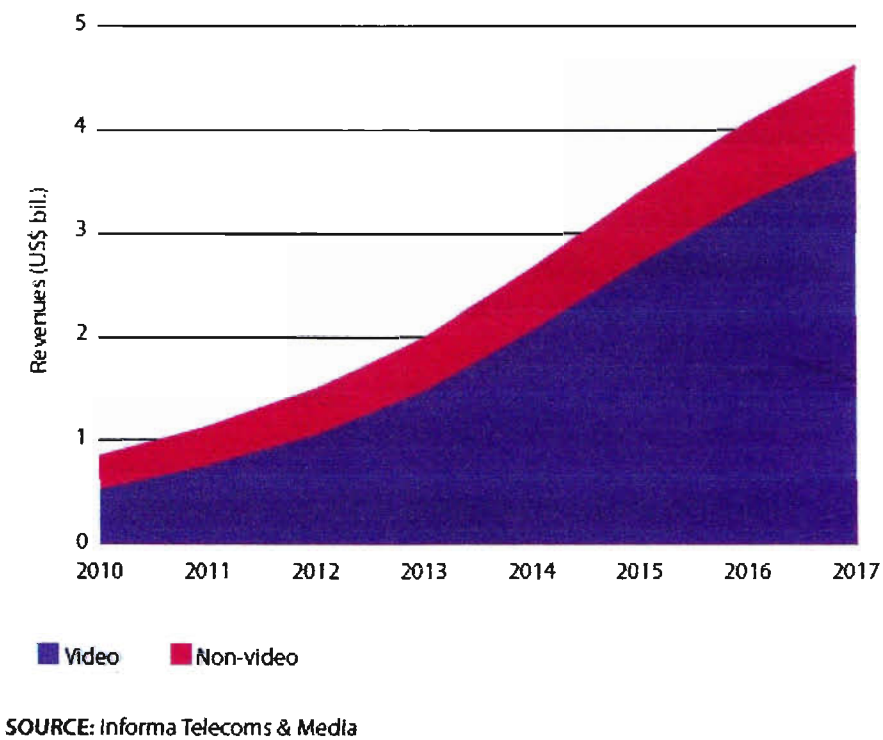


Figure 11: Global, CDN revenues, 2010-2017

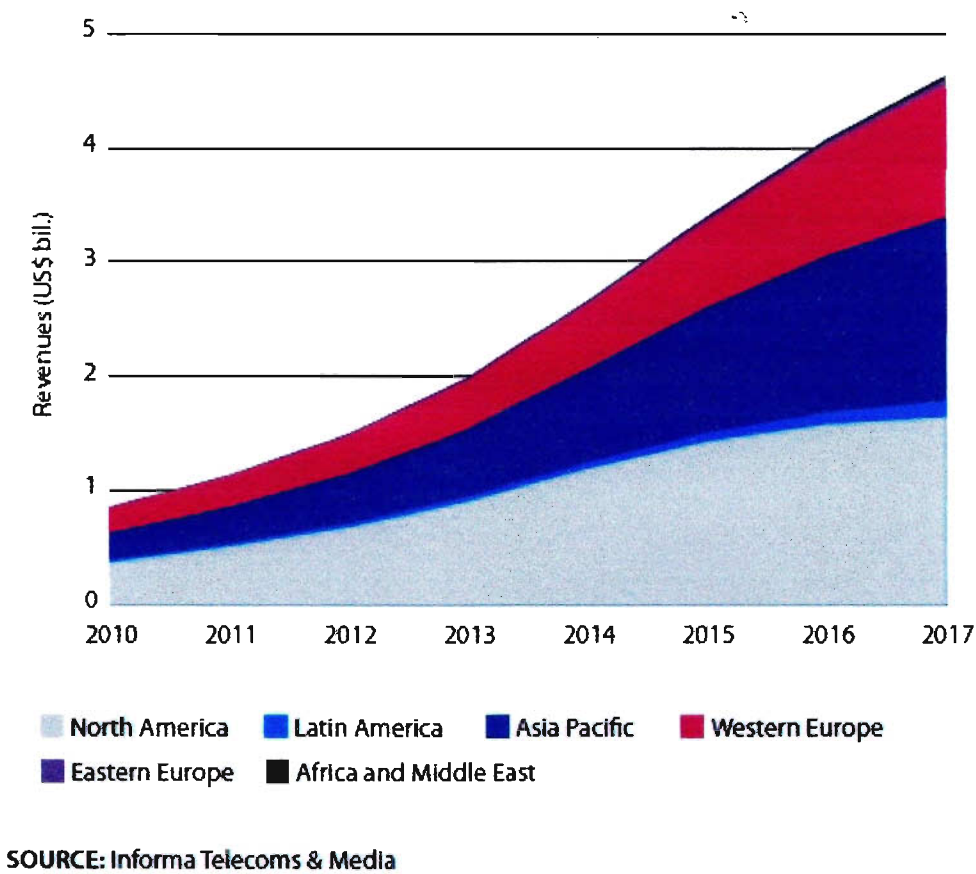


Figure 12: Global, CDN revenues per region, 2010-2017

CHAPTER: 5

Conclusion and Future work

Conclusion:

In summary, this thesis focused on the IPTV video streaming in Content Distribution Network through study on different IPTV services and how it can be integrated in distributed CDN network architecture. We mentioned and discussed various parameters, techniques, Network architectures and protocols related to IPTV streaming over CDN network, such as

- Different IPTV service that can be provided in CDN
- Concepts on video streaming
- Different protocols for streaming
- Encoding for streaming and real-time considerations
- Overview of CND network and reasons for choosing it
- Different types of CDN
- Quality of experience
- System architecture
- Implementation
- Routing configuration
- Market for CDN

Future work:

Some specific areas for future extensions include:

- Cost effective setup of CDN network for IPTV video streaming
- Robust container and video data packaging design to minimize data loss
- Optimization of the data chunk size specially for video streaming
- Mobile application based content delivery in CDN network
- Content delivery through software defined network
- Combination of p2p and CDN network for video streaming
- Improve security of the CDN network
- ISP-CDN collaboration is also an enabler for both ISP and CDN to jointly introduce new applications to end-users

REFERENCES

References:

[1] http:// http://en.wikipedia.org/wiki/IPTV

[2] http://en.wikipedia.org/wiki/Content_delivery_network

[3] http://en.wikipedia.org/wiki/Streaming_media

[4] http://www.cherritech.com/proficiency/video-broadcasting/content-delivery-network.html

[6] http://www.eyepartner.com/hosting/best-cdn-video-streaming-services/

[7] Cisco System, Inc, Cisco Wireline Video/IPTV Solution Design and Implementation Guide, Release 1.1 , 2006

[8] http://www.akamai.com

[9] http://www.juniper.net/techpubs/software/erx/erx41x/swconfig-routing-vol2/html/bgp-config.html

[10] http://www.informatandm.com

[11] http://www.caida.org

[12] http://searchtelecom.techtarget.com/feature/Planning-an-IPTV-strategy

[13] A.K. Pathan and R. Buyya, A Taxonomy and Survey of Content Delivery Networks,Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia. Feb. 12, 2006.

[14] R.Buyya, A.K. Pathan, J. Broberg, and Z. Tari, A case for peering of content Delivery Networks *IEEE Distributed Systems*, Vol. 7 , No.10, Oct., 2006