Università degli studi di Roma ***"Sapienza"***

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni

Development of the IPTV service. Multicast e Unicast switching in the Fast Channel Change Scenario

Relatore:

Maria Gabriella Di Benedetto

Laureanda:

Carlotta Gessini

Table of Content

Introduction

IPTV (**Internet Protocol Television**) is a system where a digital television service is delivered using Internet Protocol over a network infrastructure, which may include delivery by a broadband connection. A general definition of IPTV is television content that, instead of being delivered through traditional broadcast and cable formats, is received by the viewer through the technologies used for computer networks. The official definition approved by the **International Telecommunication Union focus group on IPTV** (ITU-T FG IPTV) is as follows:

*"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 and experience, security, interactivity and reliability."*

Telco operators are doing huge investments to upgrade their network in order to provide an efficient IPTV service because they consider it as a new revenues opportunity. Since IPTV uses standard networking protocols, it promises lower cost for operators and lower cost for end user. Using set-top-boxes with the broadband internet connection, video can be delivered to household more efficiently than common coaxial cable. Programs are stored on servers and they are viewed with click of IPTV remote control. IPTV is a true interactive television because it allows a personal relationship with a broadcaster via a transactional request/response mechanism, rather than just picking up a broadcast sent to anyone and everyone. It offers an infinite number of channels also with High Definition and it is possible to see anything you want, anytime, anywhere, on any device.

The development of IPTV, in Italy, is a very actual issue also because it is influenced by the digital switch-off from analogical to digital television that is going to be completed in the 2012. In this scenario IPTV is a important competitor for Digital Terrestrial and satellite television because it offers much more advantages for consumers.

IPTV offers both **Live broadcasts** television and **VoD**.

- Live broadcast television it's like watching live TV on your computer screen, it isn't possible to pause, back up or skip through parts of the broadcast that do not interest you.
- VoD arranges movies like a playlist. Episodes or clips are arranged by title or channel or in categories like news, sports or music videos, the user can choose exactly what and when he wants to watch it.

The IP-based platform also include the ability to integrate television with other IP-based services like High Speed Internet (HSI) and VoIP. This kind of opportunity is called **Triple Play**.

Studying the **Alcatel-Lucent Triple Play service (TPSDA 2.0)** I faced the issue of the **Channel Change**. IPTV allows slow channel change. There are many factors that contribute to the Channel Change time some of them are I-Frame delay and **IP Multicast**. When a user changes the channel using the remote control the set-top-box sends a message to the network to request for a channel change. What it usually done it is to change the multicast group. Stopping the reception of a Multicast channel and starting the reception of another one is not instantaneous. Although the IP Multicast represents less than the 10 percent of the overall channel change time, it is a very interest issue for network vendors and operators which efforts are spent to minimize the network contribution to the channel change time. According with the requirements defined by **Open IPTV Forum** the IPTV Solution shall provide a **Fast Channel Change** (**FCC**) mechanism to switch from one channel to another in less than 500 msec.

As the IPTV is still evolving and the number of channels available continues to grow, fast zapping capabilities are a key performance indicator for operators and end users alike, especially when HD video comes into the equation.

Alcatel-Lucent TPSDA 2.0 approach makes channel change time lower using a switch between IP Multicast and Unicast transmission. When the end user change the channel a FCC server sends a burst of **Unicast** video content (beginning with the I-frame) to the

subscriber's set-top-box with enough information to allow an immediate channel change along with several seconds of video information to play out while the STB synchronizes with the new multicast stream. The TPSDA 2.0's implementation of FCC adhere the recent Digital Video Broadcasting (DVB) standard for Retransmission (RET). Since there is not a standard for FCC many corporations and organizations are making efforts to create it.

In this work there is a general **overview of FCC standard development** situation.

Delivering TV services over an IP network allows Service Providers to know more about the end-user. Starting from this consideration the work has also the purpose to understand if it is possible to use the Alcatel-Lucent FCC approach to achieve a better **User Traceability**. The idea is to use the Unicast transmission for get information about the user behavior.

Chapter 1

1 Definition

IPTV (**Internet Protocol Television**) is a system where a digital television service is delivered using Internet Protocol over a network infrastructure, which may include delivery by a broadband connection. A general definition of IPTV is television content that, instead of being delivered through traditional broadcast and cable formats, is received by the viewer through the technologies used for computer networks. The official definition approved by the **International Telecommunication Union focus group on IPTV** (ITU-T FG IPTV) is as follows:

*"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 and experience, security, interactivity and reliability."*

For residential users, IPTV is often provided in conjunction with **Video on Demand** and may be bundled with Internet service such as Web access and **VoIP**. The commercial bundling of IPTV, VoIP and Internet access is referred to as "**Triple Play**" service .

An understanding of IPTV depends on knowledge of basic television concepts and the Internet Protocol (IP) used to transport television as a sequence of packetized frames. That is the reason way in the follow these issues will be considered.

2 Basic television concepts

2.1 Analog Television

The original development of television was based on analog technology. Television

shows were created using an analog video camera to create a video signal that was then formatted and transmitted via a broadcasting station. In the home analog television, receivers would translate the received formatted signal and present the results on the television's display. In the Figure 1 are illustrated the major components of an analog video television system. When we replace an analog video camera with a digital camera and use digital formatting, the components shown in Figure 1 are also applicable to a digital television system.

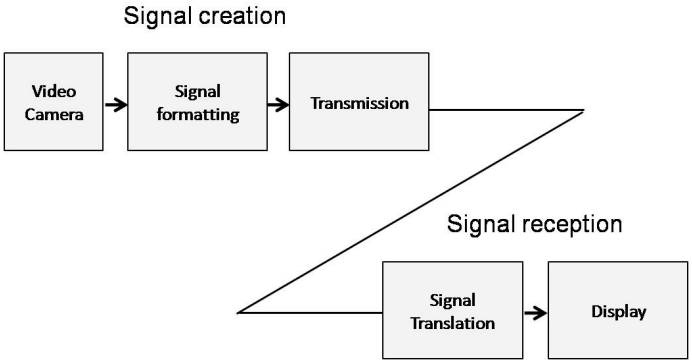


Figure 1: General television system

2.2 Digital Television

On a simplified basis, digital television can be considered to represent a method of transmitting video and audio by turning them into a sequence of 1s and 0s associated with computerized data. Unlike analog television, which uses one UHF or VHF channel to broadcast each television channel, a number of compressed digital television program streams are multiplexed into one transmission stream. This results in a combined transmission stream carrying multiple television channels and greatly increases the transmission of digital TV over analog television. In December 1996 was mandated the conversion of analog television into a digital broadcast TV standard. Broadcasters were initially given a ten-year transition period, which was recently extended to 2012. The conversion of an analog signal to digital enables data to be easily manipulated. This in turn allows the development of compression techniques that minimize both the transmission bandwidth and the storage of information. Thus, the ability to compress digital television permits a number of compressed program streams to be transmitted while minimizing bandwidth requirements. Today most cable television and

satellite operators transmit digital TV to subscribers by flowing content into a set-top-box that unbundles and decodes programs for viewing. Some newer digital televisions have an equivalent set-top box built in, allowing them to directly view digital content. There are several advantages associated with the ability to digitize television. In addition, several key differences exist between conventional analog color television (SDTV) and digital television, resulting in the latter providing both a superior picture and superior sound. Those differences include the resolution, picture scanning, color, and sound transported.

2.3 Component

An IPTV system can be represented by four major elements that are all generic and common to any system provider's infrastructure.

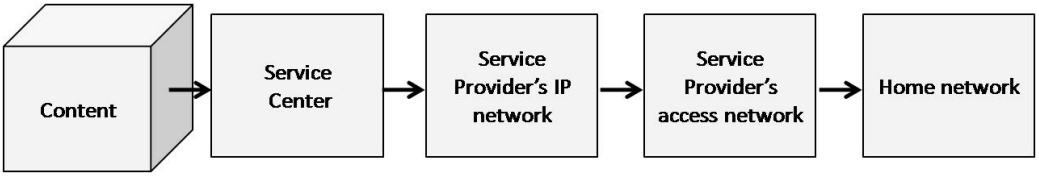


Figure 2: : IPTV infrastructure elements

Those elements include a **Service Center**, a **Service Provider's IP network**, a **Service Provider's access network**, and the **Home network**. Figure 2 illustrates the relationship of the IPTV network elements and the data flow from the content provider to the consumer. The Service Center is composed by the **IPTV Middleware** and the video **Head-end**. In examining the relationship of the four key IPTV network elements shown in the figure, it is important to note that the network elements can be provided by more than a single vendor. For example, if a consumer is using IPTV simply to download a movie or music video via the public Internet, the video head-end could represent one company and the service provider's IP network could consist of a series of IP networks interconnected at a peering point to form the Internet backbone. Then, the service provider's access network could represent an ISP, and the home network could consist of a router and wireless LAN products obtained from one or more manufacturers. In comparison, if the consumer were accessing a movie or TV show via a private IP network, the video head-end, service provider IP network, and service provider access network would be provided by a single company. In fact, it is

quite possible that that company would provide an end-to-end service, including any required home networking equipment. Now that we have an appreciation for how different organizations can provide different elements of an IPTV system, let's focus our attention on each of the elements and their function.

2.3.1 Video Head-end

The video head-end represents the point within a network where content is captured and formatted for distribution over the IP network. The video head-end for an IP network is similar to the head-ends used by cable television and digital satellite systems. That is, the IP network video head-end could be connected to satellite receivers to receive broadcast television and premium television which are broadcast via satellite. Other programming could be received via a terrestrial fiber-based connection or occur via the use of DVD or hard disk servers to provide a content-on-demand service. The head-end takes each data stream and encodes it into a digital video format, such as MPEG-2 or MPEG-4. MPEG is a mnemonic for Motion picture Experts Group, an organization that develops standards for compressing still and moving images and audio. Later we will examine several MPEG standards in some detail. After encoding, each data stream, which can be thought of as representing a conventional TV channel, is encapsulated into an IP data stream and transmitted to a specific IP destination address in response to a customer request for a particular channel. As an alternative to the transmission of TV channels to individual destinations, which is technically referred to as unicast transmission, popular IPTV channels are more than likely transmitted as IP multicast data streams. With multicast transmission, a series of TV channels in the form of data streams is simultaneously broadcast over each link of a network, with a single copy of each data stream flowing over the network. Each data stream is copied only when there is a network branch, so it can flow onto the branch, which minimizes the amount of data that flows over the network. Customers on each network branch then join a multicast group, which enables multiple customers to view a single data stream that flows over a majority of the IP network, which minimizes transmission on the backbone as well as represents a TV channel under an IPTV service.

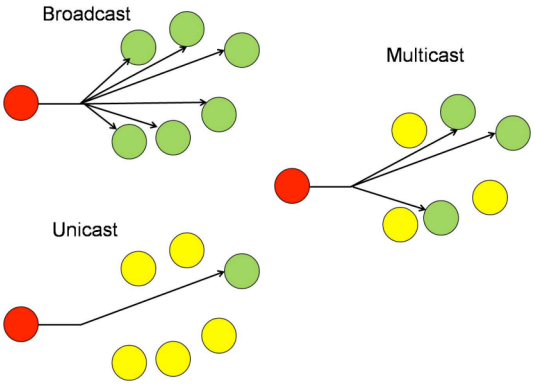


Figure 3: IP addressing methods

Figure 3 compares three popular methods of IP addressing: unicast, broadcast, and multicast. With unicast addressing data is sent to a specific destination, whereas with broadcast addressing data is read by every station. Thus, multicast addressing can be viewed as falling between the two, requiring stations to become a member of a multicast group in order to view an IPTV multicast transmission. Using multicast transmission, a service provider can transmit one IP data stream per broadcast channel from the video head-end through the IP network onto the service provider's access network. Multicast transmission can significantly reduce the flow of data over the network. For example, consider a heavyweight boxing match that tens of thousands of people may wish to view. Instead of having separate data streams of the match sent to each individual subscriber, the IPTV operator could transmit the match as a multicast broadcast. Then, tens of thousands of subscribers could tune into the match by joining the multicast group that carries the match.

2.3.2 The IPTV Middleware

The central role is played by architecture so-called "IPTV middleware that is responsible for:

• ensure integration of the various components functional such as video servers, receivers, encoder, streamer, protection system content;

• support the acquisition, processing and distribution of content processes;

• allow the definition and delivery of IPTV services;

• support integration with external systems such as OSS and CRM systems and billing.

The middleware for IPTV is one of the less standardized among technology

component and it is in rapid evolution. The lack of a middleware standard makes the market of applications and services a very fragmented one. The middleware platform implements a client/server with the centralized server component at the head-end and the client component residing on STB. The use of IPTV services occurs through the interaction user, who uses the remote control, with the UI (User Interface) and defined managed by the platform. The client component, using the basic functionality of STB supports the submission Graphical UI, decoding, encryption and display of content. The type of client supported is one of the distinguishing features of a middleware platform: it speaks of "fat client"(“client based solution") when the entire UI is generated and managed through a program Owner run by STB, while the term "thin client" is generally used for browser-based solutions, where the STB, just through the browser, it just make the rendering of HTML pages generated on the server side. A browser based solution facilitates the integration new types of terminals (STB) which is essentially required only at the level of compatibility browser, while the interaction client / server (HTTP and JavaScript) of UI benefits has generally lower compared with a client based solution. A client based solution optimizes the interaction client / server UI, but the integration of new types terminal includes a port of the client on the new terminal (activity that involves is the provider of middleware platform that the supplier of the terminal). The server component is usually implemented, using web technologies, architectures on a three levels: Web Server, Application Server, DB server. The DB is usually kind of centralized, to optimize access can be provided for different level caching. The main components and functionality of a middleware platform are summarized in Figure 4.

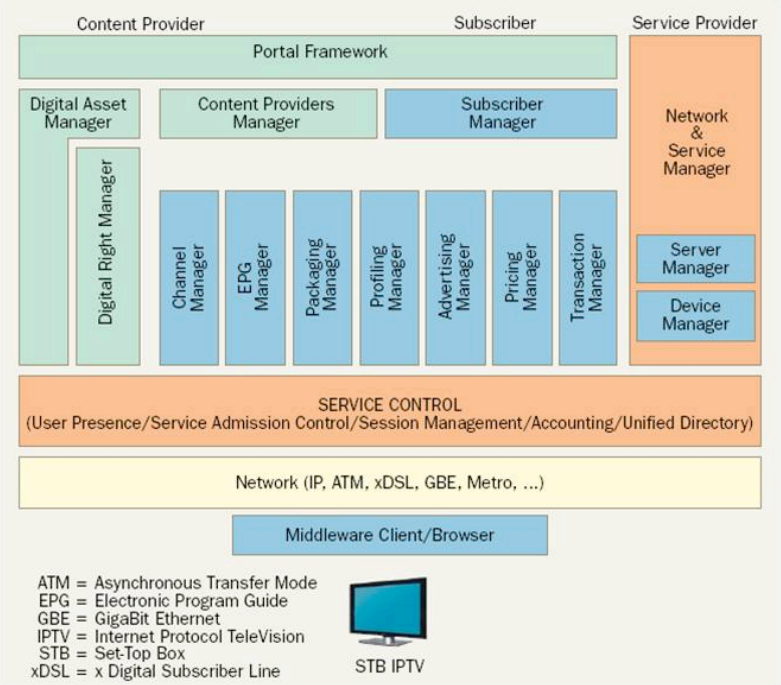


Figure 4: Middleware functionalities

2.3.3 The Service Provider Network

The service provider network can be considered as a delivery system that enables data to flow from the core of the network that is connected to the video head-end to the network edge. Over the service provider network, the channel lineup flows in the form of encoded video streams. Those flows can consist of data transmitted as unicast, multicast, and broadcast transmission. The **TV guide** that flows to each subscriber could be a broadcast transmission. In comparison, a specially requested movie could be transmitted directly to a single subscriber via unicast transmission, whereas the popular channel lineup could flow to all subscribers via multicast transmission. There is a Content Delivery Network used to deliver the content. The contents are usually replicated in the local part of the network in order to have a efficient use of resources. In the figure above (Figure 5) the hierarchical distribution points: *Super Head End*, *Video Serving Office* e *Video Hub Office* are shown.

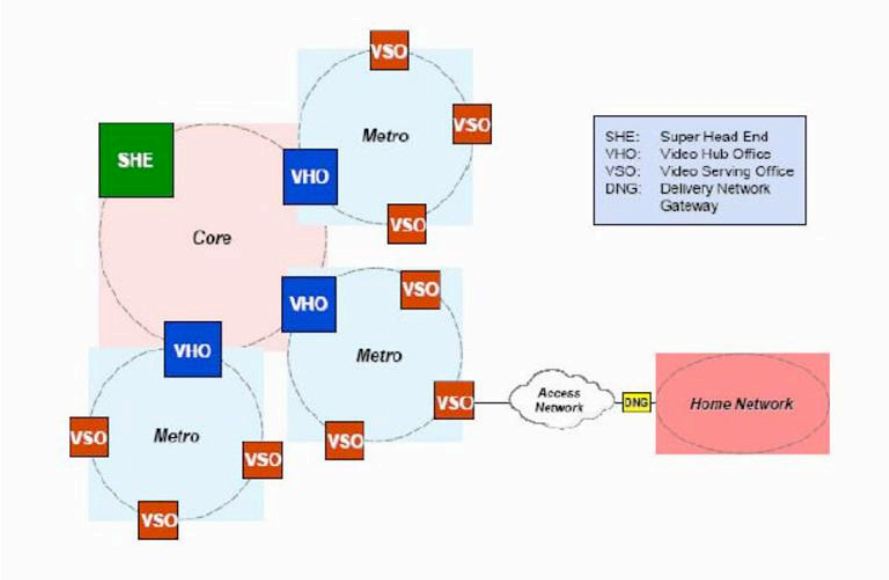


Figure 5

2.3.4 The Access Network

The access network provides connectivity from the customer's premises to the backbone network operated by the service provider. In telephone terminology, the access network is commonly referred to as the "last mile" connection. In the last mile connection several versions of Asymmetrical Digital Subscriber Lines (ADSL), very-high-bit-rate Digital Subscriber Lines (VDSL), and different types of fiber-optic technology, such as passive optical networking (PON) are used. In an IPTV environment, the service provider will use the access network to the subscriber's premises to provide a single high-bandwidth connection. That connection will enable multiple television channels, VoIP, and high-speed Internet access to be provided over a common connection to the service provider's network.

2.3.5 The Home Network

The last major network element in an IPTV environment is the home network. The home network is responsible for distributing IPTV services throughout the home. Currently, the home network is in an evolutionary stage of development, with a transition occurring from wired Ethernet to wireless Ethernet and HomePlug audio-visual (AV) equipment. Wireless Ethernet can provide data rates up to approximately 100 Mbps, and the HomePlug AV

specification enables data rates up to 200 Mbps to be transmitted over the electrical wiring in a home or office. The endpoints in the home network are telephones, the home computer or computers, and the set-top boxes that are required for each television.

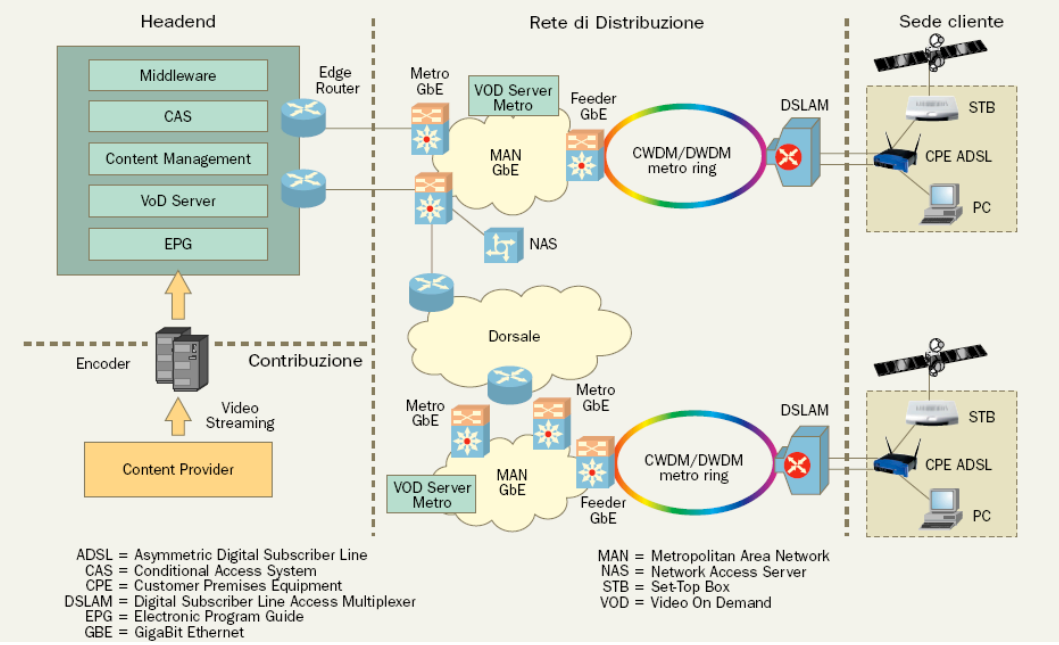


Figure 6: Example of a Network architecture

3 The TCP/IP Protocol Suite

Discussing about the TCP/IP protocol suite and IPTV it is important to note that there are two types of video that can be delivered through the use of the TCP/IP protocol suite. Those types of video can be categorized as **real-time** and **stored** for replay. The first type of video, real-time, requires the use of a **jitter buffer** to smooth out delay variations experienced by packets as they flow through an IP network. In comparison, video that will be stored and later viewed on a PC, video iPod, or other device does not require the use of a jitter buffer.

The TCP/IP protocol suite represents a layered protocol similar to the International Standards Organization (ISO) Open System Interconnection (OSI) seven-layer reference model, but it predates that model and consists of five layers. Figure 7illustrates the five layers of the TCP/IP protocol suite during the formation of a LAN frame as well as the relationship

between the layers in the ISO reference model and the TCP/IP protocol suite.

In examining Figure 7, note that the TCP/IP protocol suite does not define a physical layer (layer 1). Instead, the TCP/IP protocol suite defines a series of address resolution protocols (ARPs) that enable the network layer's addressing to be adapted to operate on the Media Access Control layer (MAC layer) supported by a particular LAN. In addition, layers 5 through 7, which represent the session, presentation, and application layers in the ISO reference model, are a single application layer in the TCP/IP protocol suite. As a LAN frame is formed in the TCP/IP protocol suite, a transport layer header, typically either a TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) header, is prefixed to application data. Both TCP and UDP headers include a source and destination numeric port number identifier, which indicates the type of application data being transported. In actuality, the destination port number indicates the application because a receiving device will "listen" on predefined port numbers to support one or more predefined applications associated with certain port numbers. In comparison, the source port is normally set to either a value of 0 or a randomly selected value.

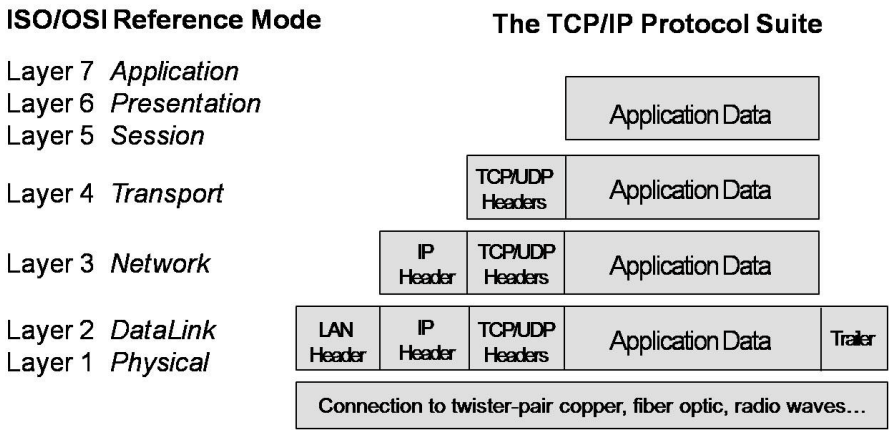


Figure 7: TCP/IP encapsulated in a LAN header

3.1 Segments and Datagrams

Several terms are used to reference headers prefixed onto application data units. First, the prefix of a TCP header to an application data unit is referred to as a TCP segment. In comparison, the prefix of a UDP header to an application data unit results in the formation of

a UDP datagram. Both the formation of TCP segments and UDP datagrams occur at the transport layer of the TCP/IP protocol suite, which represents layer 4 in the ISO reference model. When an IP header is prefixed to a TCP segment or UDP datagram, the result is an IP datagram. As indicated in **Errore. L'origine riferimento non è stata trovata.**, this action occurs at layer 3 in the ISO reference model. In comparison to TCP and UDP headers, which identify the application being transported through the use of destination port numbers, the IP header denotes the sending and receiving interfaces through the use of source and destination address fields. Thus, an IP header as well as a TCP or UDP header is required to identify both the type of data transported by an IP datagram and the originator and receiver of the datagram. As we probe deeper into the relevant fields of the protocol headers that must be considered when transporting video through firewalls and routes, we will note some of the well-known field assignments.

The **physical** and **data link** layers are responsible for transporting raw data in the form of binary 1s and 0s. The physical layer can be twisted-pair, fiber, or a wireless link, whereas the data link layer can be a form of Ethernet or another type of network. The network layer is responsible for delivering data to its destination over one or more router "hops" based on the destination IP address in the IP header. Because data flows through routers, this layer is sometimes referred to as the routing layer.

Moving up the protocol stack, we come to the **transport layer**, which is responsible for the delivery of packets. That delivery can be reliable, in sequence, when **TCP** is used or unreliable and possibly out of sequence when **UDP** is used.

Although many traditional Internet applications use TCP for the transport layer protocol, it is not suitable for digitized voice and data. This is because TCP corrects for lost packets and transmission errors by retransmission, which causes latency that adversely affects real-time applications.

Thus, IPTV primarily uses **UDP** at the transport layer. But UDP can be considered an **unreliable** protocol that depends on the upper-layer application for error detection and correction, sequencing of packets, and other actions that developers may elect to add to the application. Because early trials of IPTV had more than enough bandwidth devoted to the

video streams, the probability of packets becoming lost was virtually zero. However, as more subscribers select this service, the probability of packets being dropped by routers can be expected to increase. Thus, in the future, UDP will more than likely be used in conjunction with the **Real-Time Transport Protocol** (**RTP**), which provides time stamping and sequencing.

3.2 ICMP Messages

Internet Control Message Protocol ICMP messages convey error and control information such that they represent an integral part of the TCP/IP protocol suite. Both routers and hosts use ICMP to transmit reports concerning received datagrams back to the originator. In addition, ICMP is used to generate the well-known and frequently used echo request and echo reply messages that are collectively better known as ping messages. ICMP messages are transported as an IP datagram. This means that an ICMP message is prefixed with an IP header, resulting in the encapsulation of an ICMP message within an IP datagram. As we probe deeper into the use of the TCP/IP protocol suite to convey video, we will note how encapsulation of data through a sequence of headers is used to control the flow of video.

3.3 The Network Layer

We will commence our investigation of the operation of the TCP/IP protocol suite at the network layer. Through the prefix of an IP header, an IP datagram is formed. The IP header includes a series of fields that controls the delivery of data. Because IPv4 is currently used by more than 95 percent of all TCP/IP users, we will focus our attention on the IPv4 header even though the more modern

IPv6 header is considered to represent a replacement for the prior network layer protocol.

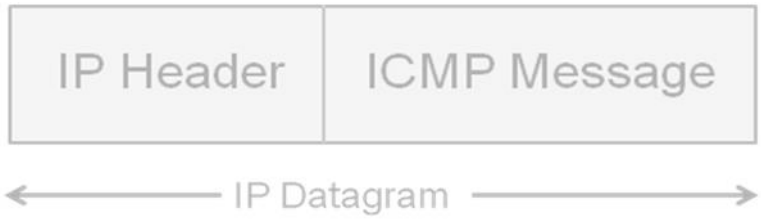


Figure 8

3.3.1 IPv4 Header

The IPv4 header is illustrated in Figure 9. Note that this header consists of 12 fields plus optional options and padding fields. The first field in the header is a 4-bit version field. This field not only specifies the version of the IP in use, but also enables the originator, recipient, and routers between the source and destination to agree on the format of the datagram. For IPv4, the value of the version field is binary 0100 or decimal 4. Although all of the fields in the IPv4 header are important, we will limit our discussion of the header fields to the delivery of video. Thus, in a video operating environment, we need to concentrate on several IP header fields. Those fields include three that provide fragmentation control (identification, flags, and fragment offset), the time-to-live field, the protocol field, and the source and destination address fields. We will also focus on IP addressing and the subnet mask because several methods based on addressing can be used to deliver video over an IP network.

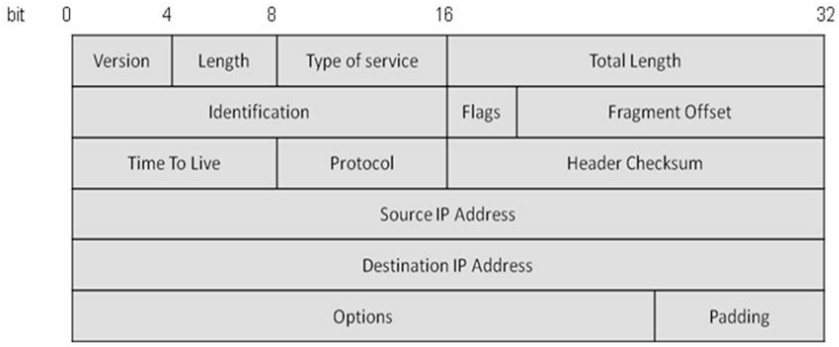


Figure 9:IPv4 Header

Version: 4 bits

The Version field indicates the format of the internet header. This document describes

version 4.

Length: 4 bits

Internet Header Length is the length of the internet header in 32 bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.

Type of Service: 8 bits

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above certain precedence at time of high load). The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput.

Bits 0-2: Precedence

Bit 3: 0 = Normal Delay, 1 = Low Delay.

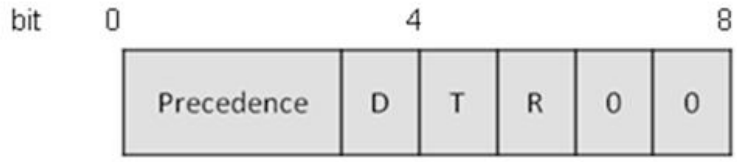
Bits 4: 0 = Normal Throughput,

1=High Throughput.

Bits 5: 0=Normal Reliability,

1 = High Reliability.

Bit 6-7: Reserved for Future Use.



Precedence

The use of the Delay, Throughput, and Reliability indications may increase the cost (in some sense) of the service.

Precedence

111	Network Control
110	Internetwork Control
101	CRITIC/ECP
100	Flash Override
011	Flash
010	Immediate
001	Priority
000	Routine

In many networks better performance for one of these parameters is coupled with worse performance on another. Except for very unusual cases at most two of these three indications should be set.

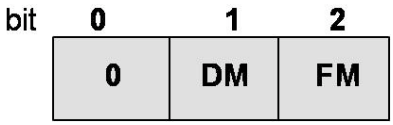
Total Length: 16 bits

Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagram are impractical for most hosts and networks. All hosts must be prepared to accept datagram of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams.

The number 576 is selected to allow a reasonable sized data block to be transmitted in addition to the required header information. For example, this size allows a data block of 512 octets plus 64 header octets to fit in a datagram. The maximal internet header is 60 octets, and a typical internet header is 20 octets, allowing a margin for headers of higher level protocols.

Identification: 16 bits

An identifying value assigned by the sender to aid in assembling the fragments of a datagram.



Flags: 3 bits

Various Control Flags.

Bit 0: reserved, must be zero

Bit 1: (DF) 0 = May Fragment, 1 = Don't Fragment.

Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.

Fragment Offset: 13 bits

This field indicates where in the datagram this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

Time to Live: 8 bits

This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Protocol: 8 bits

This field indicates the next level protocol used in the data portion of the internet datagram. The values for various protocols are specified in "Assigned Numbers".

Header Checksum: 16 bits

A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed.

The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero. This is a simple to compute checksum and experimental evidence indicates it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Source Address: 32 bits

This field represents the packet source Network layer host address

Destination Address: 32 bits

This field represents the packet destination Network layer host address.

Options: variable

The options may appear or not in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams. The option field is variable in length. There may be zero or more options. There are two cases for the format of an option:

Case 1: A single octet of option-type.

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet and the option-length octet as well as the option-data octets.

The option-type octet is viewed as having 3 fields:

1 bit copied flag,

2 bits option class,

5 bits option number.

The copied flag indicates that this option is copied into all fragments on fragmentation.

0 = not copied

1 = copied

The option classes are:

0 = control

1 = reserved for future use

2 = debugging and measurement

3 = reserved for future use

Now we come back to the source and destination addresses. Those IP addresses are further broken down into five classes, referred to as Class A through Class E. Class A, B, and C

addresses are subdivided into network and host address portions and are collectively referred to as classful IPv4 addresses. Class D addresses represent multicast addresses, where source traffic is transmitted to multiple receivers as a bandwidth conservation method. The fifth type of IPv4 address is a Class E address, which is used for experimental purposes.

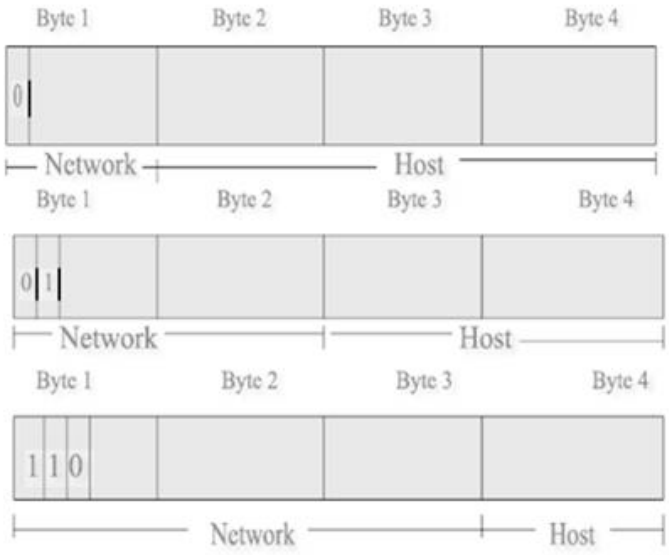


Figure 10: Classful IPv4 address formats

Figure 10 illustrates the three classful IPv4 address formats. Class D

Addresses used for multicast fall into the address block 224.0.0.0 through 239.255.255.255. Thus, the first three bits in a Class D address are set to indicate that the address represents a multicast address. In examining Figure 10, note that a Class A address is identified by a binary 0 in its first bit position. Similarly, a Class B address is identified by a binary 1 followed by a binary 0 in its second bit position, and a Class C address is identified by the bit sequence 110 in its first three bit positions. As we move from a Class A network address to a Class B and then a Class C address, the network portion of the address increases while the host portion of the address decreases. Thus, a Class A address has the smallest number of definable networks but the largest number of definable hosts, whereas a Class C address has the largest number of definable networks but the smallest number of definable hosts. Because the first bit in a Class A network is set to a binary 0, this reduces the number of Class A

networks to a maximum of 127. However, the 127.0.0.0 address represents the well-known loopback address, further reducing available Class A network addresses to a maximum of 126.

3.3.2 General Classful Address Structure

Using N to represent a network byte and H to represent a host byte, we can note the general structure of a Class A address as follows:

(N) (H) (H) (H)

As previously noted, each successive classful address byte increases the number of hosts that can be defined on a network while decreasing the number of unique networks that can be defined. Thus, the general structure of a Class B address becomes

(N) (N) (H) (H)

Similarly, the general structure of a Class C address becomes

(N) (N) (N) (H)

Due to a significant increase in the number of devices being connected to the Internet, classful addresses have become a rare and valuable resource. Although such techniques as network address translation (NAT) and the use of private IP addresses behind a NAT device have extended the useful life of IPv4, the process of subnetting has allowed organizations to considerably conserve on the use of IP addresses, further extending the useful life of IPv4. For example, consider an organization that has two LANs, one with 15 workstations used by accountants and one with 20 workstations used by engineers. Without subnetting, the organization would use two Class C addresses, each permitting 256 unique hosts (0 through 255) to be identified. However, because a host address of 0 could be confused with the basic network address and a host address of 255 is reserved as a broadcast address, the maximum number of distinct hosts that can be supported on a Class C network address is reduced by 2 to 254. Thus, for the previously mentioned organization with two LANs, the use of a single Class C address would accommodate the 35 workstations of the two LANs. This in turn would save one Class C address, which could be used to support up to 254 additional

workstations. The key to the ability to assign a common IP address to multiple networks is the process of subnetting and the use of the subnet mask. Thus, let's turn our attention to these topics.

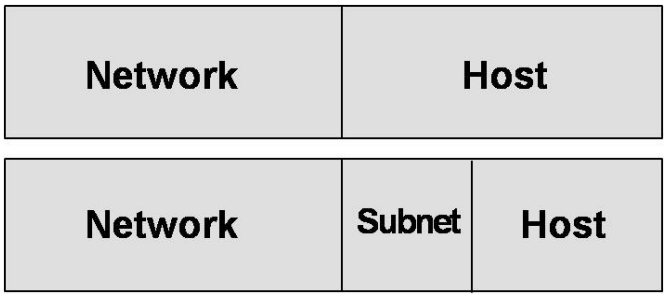


Figure 11: The subnetting process converts a two-level address into a three-level

Network	198.	78.	64.	0
Subnet 0	11000000.	01001110.	01000000.	00xxxxxx
Subnet 1	11000000.	01001110.	01000000.	01xxxxxx
Subnet 2	11000000.	01001110	01000000.	10xxxxxx
Subnet 3	11000000.	01001110.	01000000.	11xxxxxx

Figure 12: An example of the relationships among the Class C network address

3.3.3 Subnetting

Subnetting represents the process of subdividing a classful IPv4 address into two or more separate entities. The subnetting process results in the subdivision of the host portion of a classful IPv4 address into a subnet portion and a host portion, with the network portion of the address remaining as is. Thus, subnetting has no effect on routing on the Internet because the network portion of the address is not modified. **Errore. L'origine riferimento non è stata trovata.** illustrates the subnetting process, which converts a two-level classful address into a three level address. From **Errore. L'origine riferimento non è stata trovata.** it is obvious that as the number of subnets increases the number of hosts that can reside on each subnet decreases. As an example of the use of subnetting, let's assume an organization has four

LANs located in a building, with a maximum of 25 hosts on any network. Let's further assume that your organization can obtain only a single Class C address. Thus, let's focus our attention on how that one Class C address can be subnetted, which would eliminate the need for three additional Class C addresses. Because there are four LANs, we need two bits for the subnet, reducing the number of bits used to represent a host on each subnet to 6 (8 – 2). Thus, we can have up to 2 alla 6 – 2, or 62, distinct hosts on each subnet, which is more than sufficient for each LAN. Assuming the Class C IP address provided to the organization is 198.78.64.0, then the relationships among the network address, subnet, and host portions of the Class C address would appear as illustrated in **Errore. L'origine riferimento non è stata trovata.**.In examining **Errore. L'origine riferimento non è stata trovata.**, note that the host portion of each subnet is represented by six bit positions indicated by Xs, resulting in 64 distinct values that range from 000000 to 111111. Because a subnet is similar to a classful network address in that it cannot have a host address of all 0s or all 1s, this reduces the number of hosts on each subnet to 2alla6– 2, or 62.

3.3.4 The Subnet Mask

Although the process of subnetting a classful IPv4 address is relatively straightforward, an unanswered question concerns how one determines the subnet within an address. The answer to this question is the use of the subnet mask, which is formed by a sequence of binary 1s to extend the network portion of a classful address through the subnet series of bits. Because the first few bits in a classful IPv4 address identify the address type, this also indicates the initial subdivision of the address between its network portion and its host portion. Then, the subtraction of the number of bits in the network portion of the address from the number of 1s in the subnet mask indicates the number of bit positions in the subnet. For example, consider the previous example in which the network address was 198.78.64.0. Then, the subnet mask required to have a two-position subnet becomes:

11111111.11111111.11111111.11000000

In dotted decimal notation the subnet mask would be entered as 255.255.255.192. The receipts of a datagram with the network address 198.78.64.0 by a router to which the

previously mentioned LANs are connected initiates a series of steps to ensure traffic flows to the correct subnet. First, the router examines the first byte of the network address, noting that the first two bits in the byte are set. This indicates that the address is a Class C address and tells the router that the network portion of the address is contained in the first 24 bits or 3 bytes. Examining the subnet mask, the router notes that it has 26 set 1 bits. By subtracting the number of bits in the network address (24) from the length of the set bits in the subnet mask (26), the router determines that the subnet is 2 bits in length. This enables the router to examine the first two bits in the host portion of the address to determine the subnet onto which the datagram should be routed. Thus, by examining the destination address in the IP datagram in conjunction with the subnet mask, the router obtains the ability to transfer the datagram onto the correct subnet. Now that we understand the use of the IP header, a logical follow-up is to move up the TCP/IP protocol stack to the transport layer. However, prior to doing so, we need to turn our attention to a special type of IP datagram that we briefly discussed previously in this chapter. That IP datagram consists of an IP header used to transport an ICMP message. Because ICMP messages are used to perform a variety of functions ranging from providing the foundation for the well-known ping test to determining the subnet mask, it is important to obtain an appreciation of the capability of ICMP messages. Thus, let's turn our attention to this topic.

3.3.5 Understanding ICMP Messages

Although it is true that some ICMP messages can be used to exploit network defences, it is also true that preventing all ICMP messages from flowing through routers and firewalls can result in lost productivity. In this section we review the operation of 13 actively used ICMP messages. This review will make us aware of the benefits associated with ICMP messages as well as provide us with a better understanding as to why we may wish to allow certain messages to flow through routers and firewalls. Table 1 lists 13 actively used ICMP messages and their type field values. Concerning the latter, each ICMP message commences with the use of three common fields, with the remaining fields in a particular message structured based on the specific message. The three fields common to each ICMP message include an 8-bit type field, which defines the ICMP message, an 8-bit code field, which may provide

additional information about a particular message type, and a 16-bit checksum field, which is used to provide integrity for the message. In the following paragraphs we will discuss the use of ICMP messages

Type Field Value	*Defined ICMP Message Type*
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo request
11	Time exceeded
12	Parameter problem
13	Time-stamp request
14	Time-stamp reply
15	Information request
16	Information reply
17	Address mask request
18	Address mask reply

Table 1: ICMP Type Field Values

Echo Request and Echo Reply

The ICMP echo request (type 8) and echo reply (type 0) messages are used to test if a destination is active and reachable. A host or router will transmit an echo request to a distant device. That device, if both reachable and active, will respond with an echo reply. Both echo request and echo reply messages are used by the well-known ping application.

Destination Unreachable

An ICMP message with a type field value of 3 represents a destination unreachable message. The reason the destination was unreachable is further defined by a numeric entry in the message's code field. Table 2 lists the code field defined values and their meanings for a destination unreachable message. Routers can be configured to transmit a network or host unreachable message when they cannot route or deliver an IP datagram. The type field in the resulting ICMP message identifies the message as a destination unreachable message and the

code field value defines why the datagram could not be delivered.

Code Field Value	Meaning
0	Network unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed
5	Source route failed

Table 2: Destination Unreachable Code Field Values

By default, many organizations configure their routers and firewalls to block all or most ICMP messages. If this situation occurs it will adversely affect your ability to determine certain conditions by which ICMP messages could explain why datagram could not reach their destination cannot be determined. Sometimes a bit of coordination with security personnel can result in the unblocking of one or more ICMP messages, which will enable you to obtain the results you seek from the use of such messages.

Source Quench

An ICMP message type of 4 represents a source quench message. This message is used by routers and hosts to control the flow of data. To understand the use of source quench, note that when datagrams arrive at a device at a rate higher than its processing rate, the device discards them. This explains how packets can be lost. That is, assume a router connects two large domains on the Internet to a third. At various times throughout the day, the packet arrival rate from two domains destined to the third may exceed the packet processing rate of the router. The router then is forced to discard or drop packets. When this situation occurs, the device that discards the datagrams transmits an ICMP source quench message, which informs the source to slow down its datagram transmission rate. Typically, routers and hosts will transmit one source quench message for every datagram they discard.

Redirect

A type field value of 5 in an ICMP message denotes a redirect. When a router detects that it is using a non-optimum route, it will transmit an ICMP redirect message to the host. Because many hackers use this message to play havoc with an organization's network, it is commonly blocked by routers and firewalls.

Time Exceeded

The ICMP time exceeded message is generated by a router when it has to discard a datagram. Because the time-to-live (TTL) field in the IP header is decremented by 1 when a datagram flows through a router and is discarded when the value reaches 0, a router will both discard the datagram and transmit an ICMP time exceeded message back to the source when this situation occurs. A second reason for the transmission of a time exceeded message is when fragment r reassembly time is exceeded. A code field value of 0 indicates a time-to-live count value was exceeded, whereas a value of 1 denotes that the fragment reassembly time was exceeded

Parameter Problem

An ICMP type field value of 12 is used to define a parameter problem. A router or host that encounters a problem in interpreting the fields within an IP header will return an ICMP parameter problem message to the source. This message will include a pointer that identifies the byte in the IPv4 header that caused the problem.

Time-Stamp Request and Reply

From **Errore. L'origine riferimento non è stata trovata.** you will note that ICMP message types 13 and 14 represent time-stamp request and time-stamp response messages, respectively. Both messages are used to synchronize the clocks of two devices. In addition, the fields within these messages can be used to estimate the transit time of datagrams between two devices.

Information Request and Reply

From **Errore. L'origine riferimento non è stata trovata.**, another pair of ICMP messages are types 15 and 16, information request and information reply. The information request message is used to obtain an IP address for a network to which a device is attached. Thus, this

ICMP message serves as an alternative to the use of a reverse ARP. The information reply message functions as a response to the information request.

Address Mask Request and Reply

The last two type field values listed in **Errore. L'origine riferimento non è stata trovata.** represent an address mask request (17) and an address mask reply (18) message. This pair of ICMP messages enables a device to learn its subnet mask. A device first transmits an ICMP address mask request to a router. That transmission can be either as a broadcast if the device was not previously configured with the router's IP address or as a unicast message if it was configured with the address. For either situation the router will respond with the address mask in an ICMP address mask reply message. Now that we have an appreciation for ICMP messages as well as the fields within the IP header, let us move up the protocol stack and turn our attention to the transport layer.

3.4 The Transport Layer

In our prior examination of the fields within the IP header, we noted that the 8-bit protocol field defines the transport layer protocol header that follows the IP header. The transport layer permits multiple applications to flow to a common destination, either from the same source IP address or from different source addresses. To accomplish this task, the transport

Layer protocol includes a destination port field in which a numeric entry defines the application data being transported. Thus, the transport layer resides above the network layer but below the application layer, receiving application data, which is then encapsulated with a transport header that identifies the application. Once encapsulated, the TCP segment or UDP datagram is passed to the network layer, where the IP header is added to form an IP datagram.

3.4.1 TCP vs. UDP

Although the protocol field within the IPv4 header is capable of defining 256 transport layer protocols, two protocols account for the vast majority of transport layer activity: TCP and UDP. TCP is a reliable, connection-oriented protocol that includes a flow control mechanism. In comparison, UDP is an unreliable, best-effort protocol that depends on the application

layer to add such functions as error detection and correction and flow control. As an example of the use of TCP and UDP, we can consider Voice-over-IP (VoIP). For this application, TCP would be used to transport the dialled number and UDP would be used to transport digitized voice as a sequence of small periods of digitized data. Because TCP is a connection-oriented, reliable protocol, a response from the destination is required and the dialled digits arrive error free. In comparison, UDP, which is an unreliable, connectionless protocol, allows digital voice to flow to its destination and small snippets of, say, 20 ms of voice can be lost without adversely affecting the reconstructed conversation at the destination. Now that we have an appreciation for the difference between these two popular transport layer protocols, let's examine the headers of each.

3.4.2 The TCP Header

Figure 13 illustrates the fields within the TCP header. Included in the header are source and destination port fields, with the destination field used to define the application being transported. The sequence number field enables datagrams received out of order to be correctly sequenced, and other fields in the header perform flow control (window and acknowledgement) and data integrity (checksum). For the purpose of this book, which is focused on IPTV, we will limit our discussion of TCP and UDP headers primarily to their source and destination port fields.

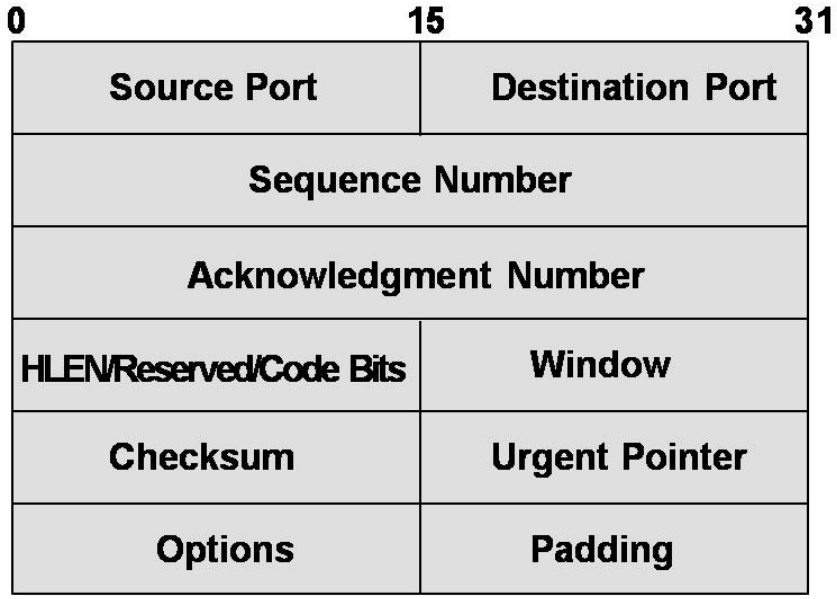


Figure 13: The TCP header

3.4.3 The UDP Header

Previously we noted that the UDP protocol represents a best-effort transport protocol that depends on the application layer for such functions as flow control and error detection and correction. Thus, as you might expect, the UDP header is streamlined in comparison to the TCP header. The Figure 14 illustrates the fields in the TCP header. Similar to the TCP header, the UDP header includes 16-bit source and destination ports that identify the process or application being transported. Thus, let's turn our attention to those two fields, which are common in both headers.

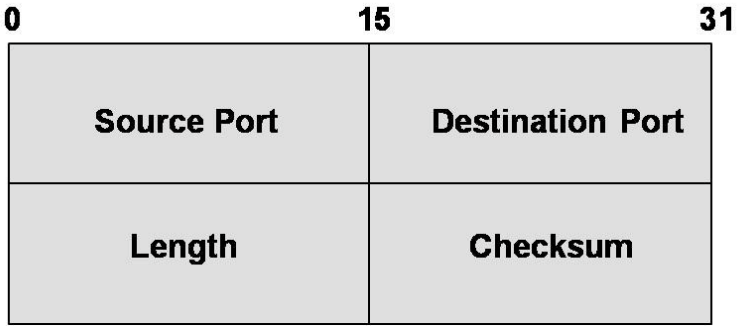


Figure 14: The UDP header

3.4.4 Source and Destination Port Fields

For both TCP and UDP, the source and destination fields are each 16 bits in length. The source port field number is supposed to denote the application associated with the data generated by the originating station. However, most source port field values are either set to 0 if the source port is not used or represent a random number generated by the originator. In comparison, the destination port field contains a value that identifies a user process or application for the receiving station whose IP address is denoted by the destination IP address field value in the IP header. Because a pair of origination and destination addresses data flows can occur on multiple destination port numbers, the use of the port field enables multiple applications to flow to a common destination. For example, when a station initiates an HTTP session, it would place port number 80 in the destination port field. Later, the HTTP session could be followed by a Telnet session, with the originating station placing port number 23 in the destination port field. Because there are three types of port numbers that

can be used in the TCP and UDP port fields, let's examine port numbers in more detail.

Port Numbers

Both TCP and UDP headers, as illustrated in **Errore. L'origine riferimento non è stata trovata.** and **Errore. L'origine riferimento non è stata trovata.**, contain 16-bit source and destination port fields, enabling port numbers to range in value from 0 to 65535. This results in a "universe" of 65536 port numbers, which are subdivided into three ranges referred to as well-known ports, registered ports, and dynamic or private ports.

Well-Known Ports

Well-known ports are also referred to as assigned ports because their assignment is controlled by the Internet Assigned Numbers Authority (IANA). Well-known or assigned ports are in the range of 0 to 1023, providing 1024 possible assignments. Such ports are used to indicate the transportation of standardized process and for the most part have the same assignments for both TCP and UDP. Ports used by TCP typically provide connections that transport relatively long-term connections requiring error detection and correction, such as file transfers (FTP)

and remote access (Telnet).

Registered Ports

Port numbers beyond 1023 can be used by any process or application. However, doing so in a haphazard manner could result in incompatibilities between vendor products. To alleviate this potential problem, the IANA allows vendors to register their use of port numbers, resulting in port Number values from 1024 to 49151 allocated for registered ports. Although a vendor can register an application or process with the IANA and obtain a port number for the use of the process or application, the registration does not carry the weight of law. That is, registered ports primarily allow other vendors to develop compatible products and end users can configure equipment to use such products. For example, when a new application uses a registered port number, it becomes a relatively easy task to both adjust a router access list or firewall configuration to enable the flow of datagrams used by the new application as well as purchase and use other vendor products that perform a similar function through the use of the same registered port.

Dynamic Ports

Dynamic ports begin where registered ports end, resulting in their use of ports 49152 through 65535. Port numbers in this range are commonly used by vendors implementing proprietary network applications. A second common use of dynamic port numbers is for NAT, which we will discuss in the next section because it can adversely affect certain IPTV operations. Table 3 provides a few examples of well-known and registered port numbers. Although some services and applications may be familiar to readers, a few deserve a bit of explanation. Bit Torrent represents an application and peer-to-peer File Transfer Protocol (FTP) that sends portions of files from one client to another. A central server, referred to as a

Tracker coordinates the actions of peers. Because Bit Torrent enables uploads and downloads to occur simultaneously, it makes more efficient use of bandwidth. In addition, because large files, such as videos, are broken into smaller pieces, the use of Bit Torrent enhances the availability of popular files; instead of an “all or nothing” approach to downloading, a file may be split into hundreds of pieces that can be obtained from many sites. A second protocol worth noting is the RTP, which provides end-to-end network transport functions suitable for

applications transmitting real-time data, such as audio and video, over multicast and unicast network services.

Service	Port Type	Port Number
Well-Known Ports		
Remote job entry	TCP	5
Echo	TCP and UDP	7
Quote of the day	TCP	17
File transfer (data)	TCP	20
File transfer (control)	TCP	21
Telnet	TCP	23
Simple Mail Transfer Protocol	TCP	25
Domain Name Server	TCP and UDP	53
Trivial File Transfer Protocol	UDP	69
Finger	TCP	79
Hypertext Transfer Protocol	TCP	80
Secure HTTP	TCP	443
AppleTalk Filing Protocol	TCP and UDP	548
Kazaa	TCP and UDP	1214
Registered Ports		
Lotus Notes	TCP	1352
Novell Group Wise	TCP and UDP	1677
H.323 host call	TCP and UDP	1720
MSN Messenger	TCP	1863
Yahoo Messenger: voice chat	TCP and UDP	5100–5001
Yahoo Messenger	TCP	5050
Yahoo Messenger: Web cams	TCP	5100
AOL Instant Messenger	TCP	5190
Bit Torrent	TCP and UDP	6881–6889, 6969
RTP-QT4 (Apple QuickTime)	UDP	6970–6999
RTP	UDP	16384–32767

Table 3: Examples of Well-Known and Registered TCP and UDP Services

3.4.5 Network Address Translation

Network address translation (NAT) was originally developed as a tool to extend the life of scarce IPv4 addresses. As the use of the Internet expanded, the ability of organizations to obtain IPv4 addresses became more difficult. Because only a portion of an organization's workstations might require access to the Internet at a particular period of time, the use of a classful IP address could result in wasting many host addresses on a network. Recognition of the fact that some organizations would not directly connect their workstations to the Internet resulted in three address blocks being reserved for private Internet use. Those address blocks,

which are listed in Table 4, are also defined in RFC 1918. By combining an address translator to map or translate unregistered, private IPv4 addresses into a registered address, it became possible to conserve IP addresses. For example, suppose your organization has five LANs, each with 200 workstations. Instead of assigning each workstation a scarce IPv4 public address, you could use five private IP Class C network addresses from Table 4. Then, using an address translator, your organization would translate RFC 1918 addresses to a single public IP address. Obviously, without a technique to differentiate one translation from another, havoc would result. Thus, to ensure each translation is unique, the address translator uses or assigns a high port number to the source

Port in the TCP or UDP header and enters the RFC 1918 IP address and the port number into a table.

Address Blocks
10.0.0.0–10.255.255.255
172.16.0.0–172.31.255.255
192.168.0.0–192.168.255.255

Table 4: RFC 1918 Reserved IPv4 Addresses

Then, when a response occurs, the address translator notes the port number returned in the header and uses that number to perform a table lookup, noting the RFC 1918 address associated with the port number. Next, the address translator converts the Class C public address in the IP header's destination IP address field to the recently obtained RFC 1918 address and forwards the datagram to the private IP network. This type of translation is referred to as port-address translation and is performed by many routers and firewalls. Although NAT can considerably economize on the use of scarce IPv4 addresses, it can create problems when some type of tunnelling is employed and the inner IP datagram remains as is, with the outer IP header operated on by NAT. This means that you could not interconnect two networks that use the same RFC 1918 private network addresses, because doing so would result in routing problems on each network. Instead, you would need to change the network address on one of the interconnected networks.

Chapter 2

1 Delivering IPTV

IPTV covers both live TV and stored video. Video content are compressed using a standard codification and then sent in an MPEG transport Stream delivered via **IP Multicast** in the case of live TV or via **IP Unicast** for the VoD.

The protocol used change according to the delivered service:

- Live TV uses **IGMP version 2** or **IGMP version 3** for IPV4 for connecting to a Multicast stream (TV channel) and for changing from a multicast stream to another (TV channel Change)
- VoD is using the Real Time Steaming Protocol (**RTSP**)
- N-PVR (Network-based Personal Video Recorder) is also using the **RTSP**

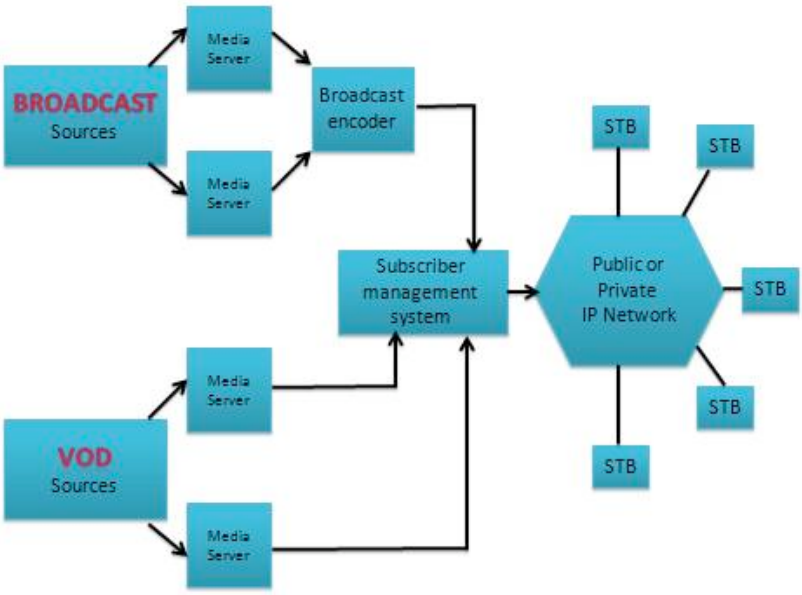


Figure 15: IPTV Services

Live TV is the **Broadcast** TV. The channel producer selects the content to be

broadcast at a given time; all users may select the channel they wish to watch. Commercial offerings may consist of channel packages with each package sold individually, and usually charged on a monthly fee-per-package basis. An electronic program guide provides information on the content broadcast on each channel throughout the day and some details on the program currently on display.

Video on Demand permits a customer to browse an online program or film catalogue, to watch trailers and then to select a selected recording for playback. The play-out of the selected movie starts nearly instantaneously on the customer's TV or PC.

Technically, when the customer selects the movie, a point-to-point unicast connection is set up between the customer's decoder (STB or PC) and the delivering streaming server. The signaling for the trick play functionality (pause, slow-motion, wind/rewind etc.) is assured by RTSP.

In an attempt to avoid content piracy the VoD content is usually encrypted. While encryption of satellite and cable TV broadcasts is an old practice, with IPTV technology it can effectively be thought of as a form of Digital Rights Management. A film that is chosen, for example, may be playable for 24 hours following payment, after which time it becomes unavailable.

Network Personal Video Recording is a consumer service where real-time broadcast television is captured in the network on a server allowing the end user to access the recorded programs on the schedule of their choice, rather than being tied to the broadcast schedule. The NPVR system provides time-shifted viewing of broadcast programs, allowing subscribers to record and watch programs at their convenience, without the requirement of a truly personal PVR device. It could be compared as a "**PVR that is built into the network**" -- however that would be slightly misleading unless the word "Personal" is, of course, changed to "Public" for this context.

2 Standard and Protocols

The role of standards is important with respect to the digitization of the television

signal and its transport on IP: the ten years experience of digital television (via cable, satellite or terrestrial) is ,so, a solid base for the IPTV.

The analog TV signal PAL (Phase Alternate Line) not compressed requires the availability of bandwidth of 270 Mbits/s, that is much more of the transmission capacity of many of resources available for distribution of signals TV over IP networks. Similarly the North American television, which has a lower color rendering, requires 168 Mbits/s. The situation becomes much higher if we take into account high-definition television, so the band demand becomes greater than 1 Gbit/s. That is the reason why we need coding techniques, known as "lossy", which drastically reduce the required bandwidth, while introducing loss of information, maintaining the perceived quality not less than the one the viewer is used with the Analog television. **Errore. L'origine riferimento non è stata trovata.** describes the need of bandwidth for some television systems.

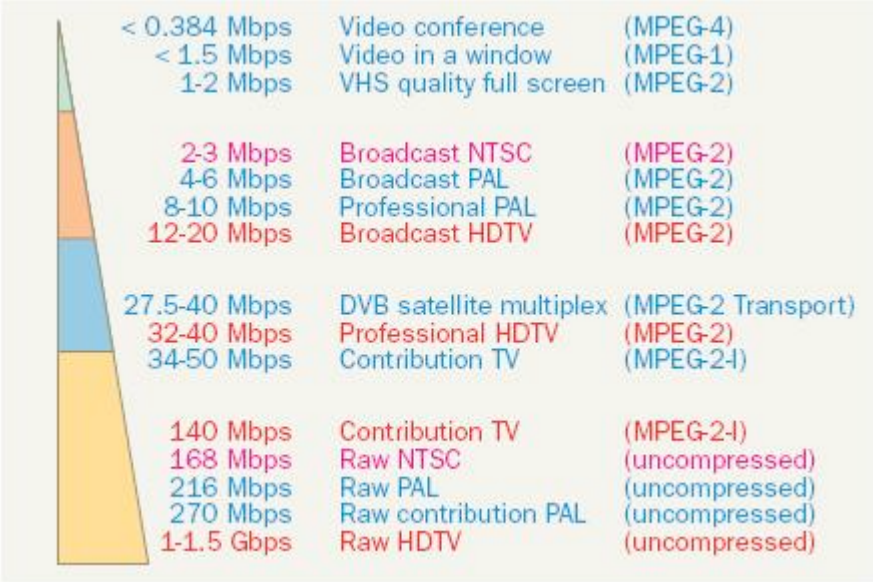


Figure 16: Required Bandwidth for different television systems

One of the key factor for the future development of IPTV services over ADSL is the maturity reached by new generation of encoders.

In standard television, any movie is made up of many photographs called **frames** which are put into rapid sequence. Each image broadcast TV is made up of 30 (or 29.97 for

the devices color) frames per second (fps), if the used standard is NTSC (North America and Japan), or 25 fps if the standard is PAL (Europe, Africa Eastern Europe, India, Australia, China). Every single frame is interlaced, which consists of two 'half-frame' that appear and quickly combine forming the frame, each half frame is made up of alternate lines, one containing the odd lines, the other the even lines.

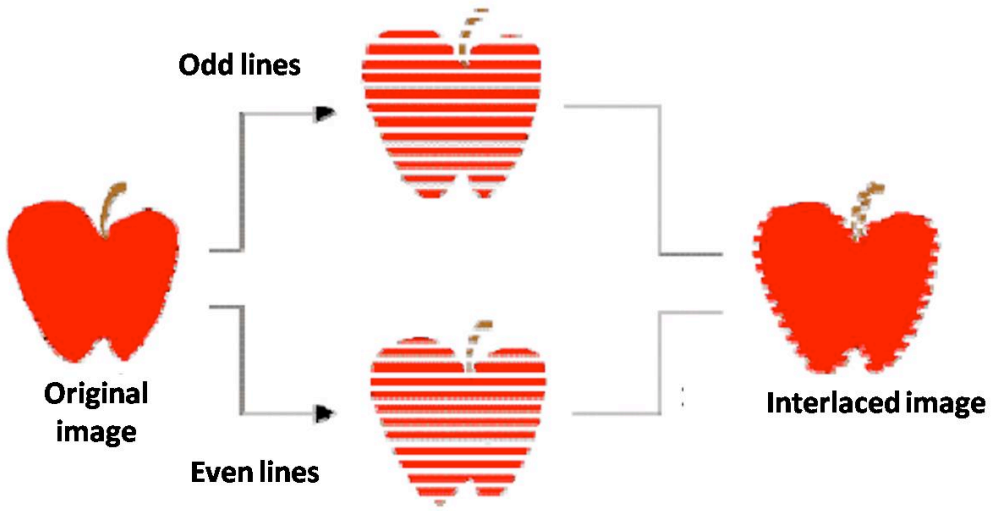


Figure 17: Interlaced frame

Monitors and televisions (and also monitors for PCs) have a fixed relationship width/height that is called **aspect ratio.** For the classic television this is 4:3 while recent panoramic screens have an aspect ratio 16:9.

2.1 MPEG Standard

The video takes an huge amount of space in terms of megabytes (1048576 bits) that the reason why the **Motion Picture Experts Group** (**MPEG**) has established systems of "lossy" video compression to put more and more minutes of movie, quality increasingly, in fixed memories such as CD and DVD. Thus was the MPEG-1, which allowed to put on a CD (650 MB) one hour of video with quality comparable to a VHS videotape. Subsequently was MPEG-2, designed for television broadcasting and used to store movies on DVDs (DVD Video), which offers higher quality and space than its predecessor. Since the MPEG-2 standard provides a good compression using standard algorithms, it was chosen as the coding

technology for digital television.

MPEG is an established standard that enables the exchange of compressed data from different systems. The main difference is detectable in the techniques of data analysis and compression. MPEG analyze the entire sequence of frames (interframe compression). Each frame of these sequences may be of a type **I**, **B** and **P** and are collected into groups called **GOP** (Group of Pictures). A GOP must include at least one frame type I while the length and structure of the sequence can be freely defined by the manufacturer. The **I-frame** are the reference images and are individually compressed. Each area within a frame can be compressed by different factors, for example, the center can be a factor lower than the edges, with a saving of 15% in flow data without visible loss of quality. In the MPEG data stream, the I-frames contain all the information required for decompression and display of the image. The frame B are compressed in two directions and contain only the data for differences between two other frames. The B frames contain a very lower number of information than the frame I. The drawback is that to decompress and display a B-frame it is necessary to refer to the previous and next frame. The P-type frames are called predicted. These are obtained by interpolation from other frames of sequence and contain much less data frame B. The composition of the GOP and the quantity of various types of frames I, B and P depends on manufacturer. The only prerequisite is the presence of, at least, one I-frame.

It was defined a number of levels and profiles for video compression in MPEG-2. The levels keep account the resolution while profiles take the quality of the video. Each of them includes a subset of overall functionality described in MPEG-2. MPEG-2 implementations typically affect a specific couple of profiles and levels. In the IPTV system the chosen couple is the Main Profile Main Level (MP@ML), which covers a range of compressions 1 Mbit/s to 15 Mbit/s. The main systems used for resolutions PAL are:

- 720 X 576 X 25 fps
- 704 X 576 X 25 fps
- 544 X 576 X 25 fps
- 352 X 576 X 25 fps

The audio associated with video content varies according to the source: it goes from content with a mono channel to content for which the audio is a Dolby Digital 5.1. The audio can be encoded in various ways, with increasing efficiency: the most common encodings are MPEG-1 Layer 1, Layer 2 and Layer 3 (MP3), in particular, it often uses MPEG-1 Layer 2. The signal is typically encoded at bit rate between 96 Kbit/s and 384 Kbit/s. The Encoding technique MPEG-2 AAC (Advanced Audio Coding) much more efficient but not compatible with previous ones, is actually used for encode the audio in new technologies, such as MPEG-4 AVC.

MPEG-2 has been the encoding method for digital cable and digital satellite systems for about 15 years. The cost of MPEG-2 encoders, which can be installed in a set-top box or a PC, is thus low due to economies of scale. However, the compression efficiency of MPEG-2 is not sufficient for twisted copper pair loops, whereas cable systems can send all the video channels using the MPEG2 codec since the bandwidth of hybrid fiber coaxial (HFC) cable can be near 4.5Gbps. Moving to H.264 typically provides a 40% saving in bandwidth over MPEG-2 encoded content, enabling IPTV operators to offer High Definition (HD) services to the home. If an IPTV operator does not have sufficient bandwidth and cannot prioritize the video traffic along the IP network end-to-end with quality of service (QoS) tools, it is technically possible that the video traffic may be delayed or fragmented. While the Core network is normally a fiber optic cable the Access network of the local DSL loop from the local office to the customer's set-top box does not have sufficient capacity to stream all the live channels at once. Current ADSL broadband networks can typically support download speeds of up to 8Mbps and upload up to 256 kbps within a 1.5 km distance from the central office (or wherever the DSLAM is located). More advanced access technology, ADSL2+, for example, can provide downstream speed of up to 24Mbps. If a video program is encoded with a MPEG-2 codec, an ADSL loop can accommodate at maximum 2 standard definition (SD) channels (8 Mbps/4Mbps = 2 channels), while an ADSL2+ network can allow up to 6 SD channels (24/4=6) or 1 HD channels and 2 SD channels (15 Mbps x 1 + 4 Mbps x 2 = 23 Mbps) on the condition that the local loop is used only for video delivery.

Encoding	MPEG-2	MPEG-4 part 10 (H.264)	VC-1
Avarage SD	4 Mbits/s	1.5 Mbits/s	1.5 Mbits/s
Avarage HD	15 Mbits/s	8 Mbits/s	8 Mbits/s
DRM	NO	NO	YES

Figure 18: Encoding/Video definition

Given the limitation of the bandwidth of twisted copper wire, telecommunication IPTV operators are providing hundreds of video channels to customers by sending only selected video channels at a time from local offices to set-top boxes, instead of broadcasting all the video channels simultaneously. To do this, operators are using switched digital video technology which switches a video stream to individual set-top box only when the video stream is requested by a viewer. Each subsequent viewer on the node who requests the same channel shares the stream; the operator thereby conserves bandwidth. In a traditional broadcast network (terrestrial TV, CATV, satellite) using broadcast video technology, all the content constantly flows downstream to each customer and the customer switches (tunes) to a different channel using a set-top box. A switched IP network works differently. Content remains in the network, and only the content the customer selects is sent to the customer's home. That frees up bandwidth and the customer's choice is not limited by the bandwidth of the network to the home. The conservation of bandwidth and the capability of sending only selected content to customers who request it enable IPTV operators to provide customers with a large number of video channels and reallocate unused bandwidth to other services. IPTV operators need to upgrade their existing ADSL enabled copper lines by using more advanced transmission technologies, such as ADSL2+, VDSL15, VDSL2,16 while using the same copper lines, or/and by replacing part or the whole of the copper lines with optical fiber. A second requirement is to adopt a video codec with increased compression capability. For example, AT&T.s IPTV service, U-Verse TV, uses MPEG-4 (H.264) encoding rather than MPEG-2 encoding.

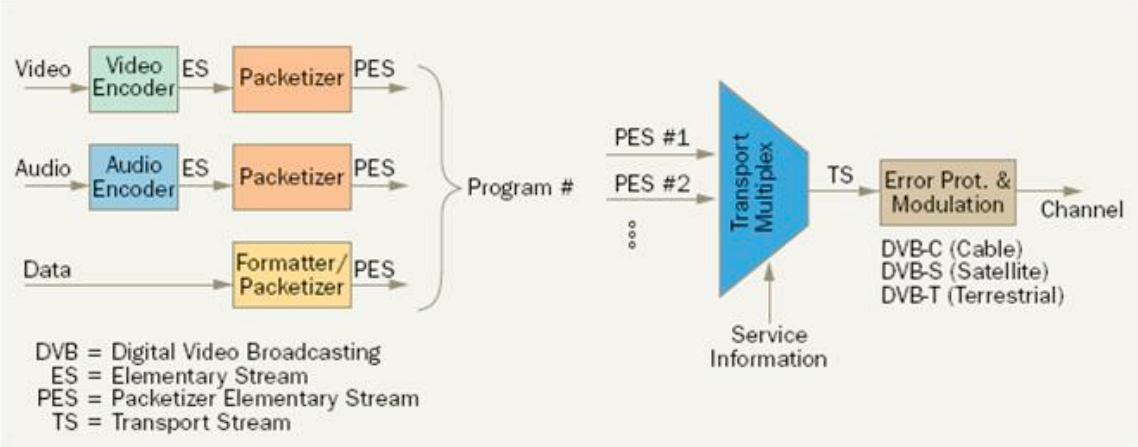
2.2 Streaming and Control protocols

2.2.1 The Application Layer

An MPEG-2 stream is composed of multiple components, called Elementary Stream (ES): a program typically contains an ES for the video, one or more of audio, control data, and data with value information such as subtitles. Each ES is divided into packets PES (Packetised elementary Stream), fixed or variable size, beginning with a header of 6 bytes in which you have the information on the type of ES, which contains, on its length and information control for the synchronization of ES: PTS (Presentation Time Stamp) and DTS (Decode Time Stamp). The standard MPEG-2 allows two types of multiplexing of packets PES:

• MPEG Program Stream: cover the case of packages PES closely related and with the same time basis. It applies to contexts in which both low probability of errors and allows a more easy processing of data. It is the form of multiplexing device used in video storage, typically the DVD;

• MPEG Transport Stream: each PES is broken in "transport packets" of fixed size; the stream may contain packets coming from independent PES. It is the form of multiplexing appropriate to the case of broadcast media on potentially affected by an error and when interested assemble more than one program in the same stream. It is the form of multiplexing adopted by the DVB (Digital Video Broadcasting) the European consortium for the standardization of distribution of signals television, and beyond. DVB deals mainly of Satellite broadcasting digital (DVB-S), but define in general as the MPEG-2 signals are transmitted also on cable (DVBC) or a television frequencies terrestrial (DVB-T) and also how information is managed service, the EPG (guides) as well as any encryption systems. The figure below summarizes what said.



As inputs to the Transport Multiplex, in addition to more than one PES, there are the SI (Service Information), namely that sum of control data used to manage the associations between the ES and describe them. The set of SI derives in part from MPEG-2 and it is partly integrated with fields defined in the DVB. Within a TS is possible to transport both, an individual and a combination of several programs. In the first case we speak of SPTS (Single Program Transport Stream) while in the second we speak of MPTS (Multiple Program Transport Stream).

SPTS configuration is typical in the context IPTV, and can contain just the PES of audio and video transmissions, in the case of transmission regulated by DVB the typical configuration is the MPTS; in this case it is imperative to include the control information inside the SI to identify each single programs.

Transport Stream Structure

Each Elementary Stream (ES) video and audio contained in a TS (Transport Stream) is identified with a specific PID (Packet Identifier), this allows the receiver to identify and filter the packets according to their PID. The description of which ES must be combined to build a program are transported within the Reporting Tables that are Small-scale packages inside the TS, separate and asynchronous compared with PES (Packetised Elementary Stream). Examples of Reporting Tables are:

• PAT (PID 0x000) (Program Association Table): it reports the list of Packet Identifier (PID) associated with individual programs;

• CAT (PID 0x001) (Conditional Access Table): It is sent in case of encrypted ES, it provides information about the type of scrambling and it indicates the PIDs associated with other information on conditional access;

• PMT (Program Map Table): it defines the set of PIDs associated with a single program: the PID of this table for each program is that previously read in the PAT.

The DVB has defined some additional Optional tables in order to describe the transported services (the role of these tables is directly understandable by name):

• NIT (PID 0x010) (Network Information Table): contains information about the physical medium where the information is transported (in the case of satellite: transponders, symbol rate, FEC, ...);

• BAT (Bouquet Association Table);

• SDT (Service Description Table);

• TDT (Time and Date Table);

• RST (Running Status Table);

• EIT (Event Information Table).

Figure 19 shows an example of a piece Transport Stream.

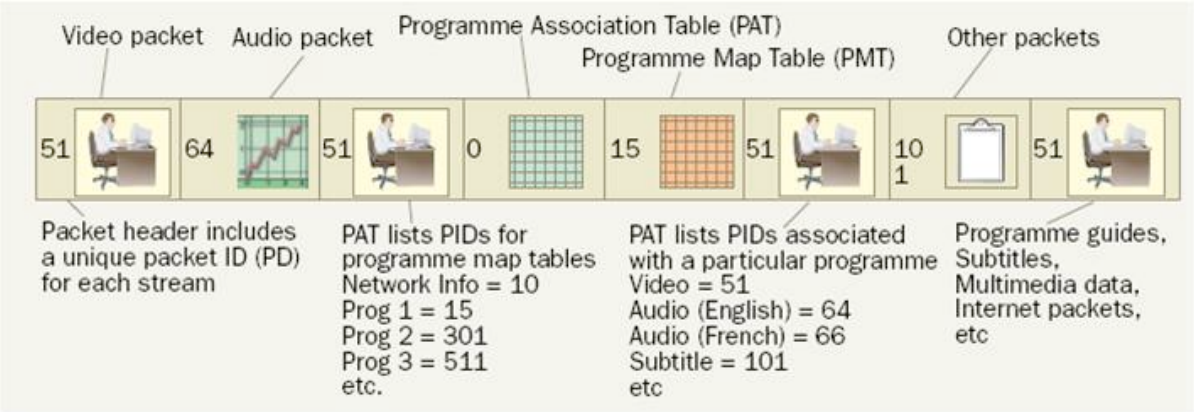


Figure 19: Transport Stream

The synchronization between the ES and the regeneration of the clock at the receiver are carried out using Timing indicators (time stamps) that are transported in the same TS, in particular:

• PCR (Program Clock Reference) is transported in a field of devoted TS or in Video package and it is necessary for the reconstruction of coherent time in both transmission and reception sides. The system clock is 27 Mhz, at the reception side a local clock is generated and from this a counter is extracted and it is compared with the received PCR. The differences are used to correct the frequency of the local oscillator in the reception side;

• DTS (Decoding Time Stamp) and PTS (Presentation Time Stamp): they are transferred in the header of each PES and serve to establish the decoding instants of a single frame and thus the synchronization between ES. The size of a transport packet is 188 bytes, and 4 of them are for the header. In this header the 13 bits of the PID is very important in order to identify each PES of which the transport packet is a party. In the Transport Stream is not guaranteed any order of transmission of packets it means that there is no synchronization between the present PES. To keep the bit rate of multiplex set in the configuration stage may be included stuffing packets (PID = 0x1fff). The structure of a transport packet is shown in Figure 20.

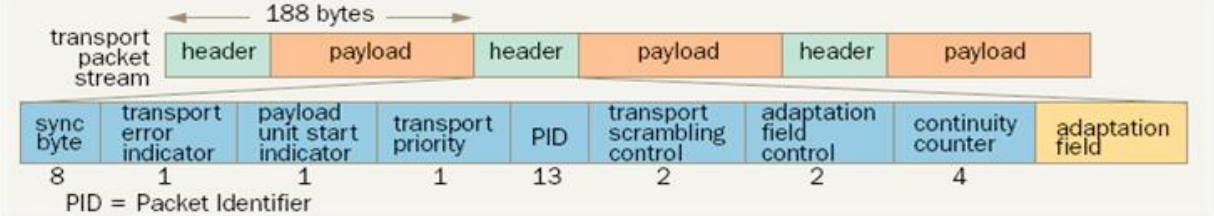


Figure 20: Transport Packet

The meaning of the various fields of the header is as follows:

• Synchronization Byte identifies the beginning of the package. His value, fixed, it's 0x47;

• A set of three flags to indicate how the package must be processed;

• Packet Identifier (PID)

• Two bits used by the Access Control procedure;

• Two bits to indicate the presence of the "adaptation field":

01 - no adaptation field, payload only;
10 - adaptation field only, no payload;

11 - adaptation field followed by payload;

00 - Reserved for future use;

If the adaptation field is present it is located after the header and before the payload. An important content in the "adaptation field" is the PCR (Continuity Counter, 4 bits) that is a very useful counter for evaluating continuity of packets received and to report any losses.

2.2.2 The protocol stack for the live content

Since the most critical network path network is represented by ADSL, it will give an indication of the protocols present at an up to the physical level.

The physical characterization of available bandwidth on the ADSL must be such as to ensure the accessibility of a high proportion of user charges on telephone twisted pair, which introduces attenuation and losses in proportion to its length. The most interesting aspect is the analysis of the downstream component, which pass the contents audio/video as it puts more constraints. The payload to be included in the protocol stack is typically made of the maximum number of transport packet (size 188 bytes each) that can be allocated in an Ethernet plot (1500 bytes). That is the reason why the payload has size of 1316 bytes (7 plots from whole 188 bytes each). This option allows you to make the best use of the information content transportable in a single Ethernet plot, and then to contain the overhead of the protocols. It is not the only possible solution: would be possible to have more number of frames aggregated into a single payload segment, even up to an order of magnitude higher, but it is not recommended over IP networks in order to avoid having recourse to the fragmentation IP packet and the subsequent reassembly with the costs of delay and risk of error.

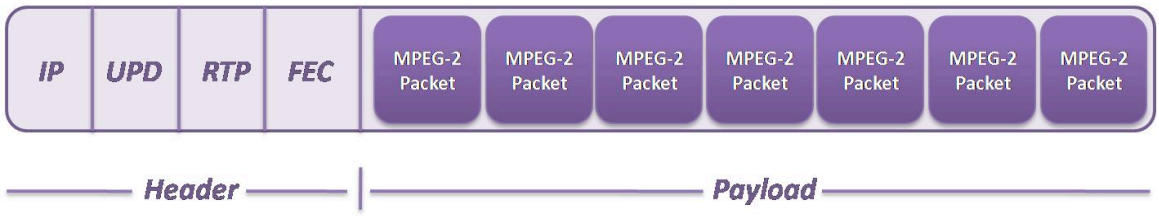


Figure 21: IP Encapsulation (MPEG-2 Transport Stream) of seven MPEG-2 encoded packets

Above the network layer (IP), the transport layer, in this case is always UDP: the advantage is

the highest speed (the type of content and constraints imposed by the streaming real time do not allow the Any delay introduced by retransmission of erroneous packets), the cost is the lack of recovery mechanisms of errors. A wrong UDP packet is simply discarded but in this way there is a loss of no less than 7 frame of content. Between the transport layer and application layer (Transport Stream) to the Protocol RTP (Real Time Protocol) can be added. It is an IETF standard protocol for the controlled transfer of audio and video. Its use is optional and often not used. The STB is able to interpret both, flows encapsulated in RTP and flows in which the plots are directly entered into the payload of UDP packet. Even if there is a more cost of adding an additional overhead and elaboration capacity both on the encoder that on the decoder side, the use of RTP would allow greater efficiency and flexibility in the management of elementary audio and video streams. Today, as said Previously, the sound is normally transported embedded with the video in a single transport stream, with the use of RTP each Elemantary Stream could travel on its own behalf, because RTP protocol has the controls to synchronize flows. For example the use of RTP mode mentioned above would be advantageous in the case of Multilingual audio. If the sound travels embedded with the video there are two alternatives, the fact is that these are both inefficient. As first you can send all audio channels associated with the various languages that the user may choose to listen, with the obvious waste of bandwidth on ADSL and this bandwidth is critical resource. As second you can send a number of separated flows of the same content equal to the number of languages from the Head-end. In this way there is a weight of the occupied band in the backbone and metro network, as well as number of involved encoders. Using RTP there is always the transmission of a single flow of the video Elementary stream and many audio streams as the languages offered. A RTP has an associated control protocol RTCP (Real Time Control Protocol) acting on backward from the receiver (STB) and it provides control information on the instantaneous state of communication to the source. Using this protocol it could be possible to imagine new solutions, for now not commercially implemented, like Automatic band usage.

2.2.3 The role of Multicast

Live contents are broadcast using the IP multicast protocol. Above the network layer protocol stack remains unchanged compared to other types of audio/video: at the transport layer UDP is used and on Application Layer the use of RTP is optional. The choice of multicast is critical to minimize the number of IP flows that are moving on backbone network and the metro. With this choice there is only one stream for each live channel that is provided. As with traditional TV channels, the user began to enjoy the content on the chosen channel at the time of its connection. The user receive what is transmitted at that time and not from the beginning. The IP multicast means to associate each flow with an address in a particular range (from 224.0.0.0 to 239,255,255,255) and a port in the transport layer. The couple address/port uniquely identifies the multicast stream and then the live channel. The rules of the multicast network provide, through the use of protocols for signaling between routers, that the multicast flow circles on a subnet only if its certificated users,(or users certificated to subnets below that one),have applied. In the case of IPTV, the service model assumes that all multicast flows provided by the head-end are received from each DSLAM, which shall replicate in the ports of the users who request it. In this context all multicast flows for all the provided live channels are constantly on the Backbone and Metro network. The protocol with which the user, and so the STB, requires to receive a live channel is the IGMP (Internet Group Management Protocol). Usually the version implemented on commercial systems is the 2nd. The IGMP protocol contains, among others, primitive of Join, Leave the Membership Report for multicast flow. The Join and Leave have, respectively, the role of require/release of a flow. Membership Report primitive informs the above resource (in the case of IPTV it is the DSLAM, in general, the first router) that the current multicast flow continues to be of interest to the device that generates the primitive, in the absence of such feedback, after a timeout, the flow on that door is locked (possible causes: shutdown the STB, a physical disconnect, ...). In order to have a better explanation of the IGMP protocol you will find a devoted section below. Figure 22 is a simplified view of a multicast network used to deliver a *single channel* (using a single multicast address) to a number of group members (or viewers of a particular channel). The flow of data from the Group Source to the Group Members follows the multicast routing tree. A multicast routing protocol such as PIM-SM/DM is used

to create an efficient routing path for the delivery of multicast packets between routers. As said above the communication between the network routers and the STB is generally facilitated using IGMPv2.

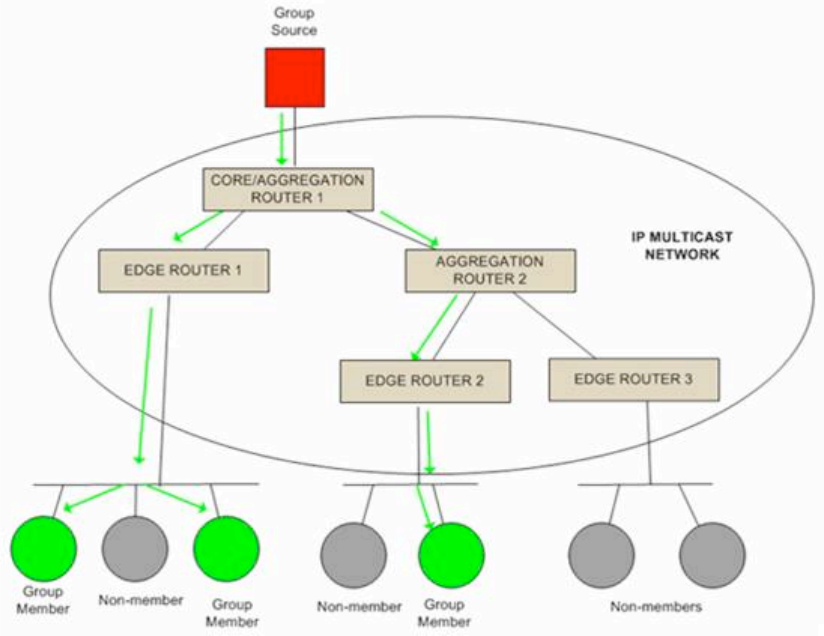


Figure 22: Simple multicast tree used to deliver single channel multicast traffic to hosts

2.3 IGMP protocol

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

IGMP is an asymmetric protocol now we are going to specify it from the point of view of a host, rather than a multicast agent. IGMP is an integral part of IP. It is required to be implemented in full by all hosts conforming to level 2 of the IP multicasting specification. IGMP messages are encapsulated in IP datagrams, with an IP protocol number of 2. All IGMP messages have the following format:

2.3.1 IGMP Version 1

IGMP Version 1 is defined in RFC 1112. In this protocol there are just two different types of IGMP messages:

- Membership query
- Membership report

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

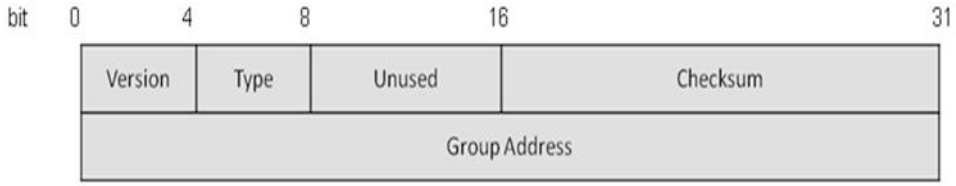


Figure 23: IGMPv1 message format

2.3.2 IGMP Version 2

RFC 2236 defines the specification for IGMP Version 2. In Version 2, there are four types of IGMP messages:

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

IGMP Version 2 works basically the same as Version 1. The main difference is that there is a leave group message. The hosts now can actively communicate to the local multicast router their intention to leave the group. The router then sends out a group-specific

query and determines whether there are any remaining hosts interested in receiving the traffic. If there are no replies, the router times out the group and stops forwarding the traffic. This can greatly reduce the leave latency compared to IGMP Version 1. Unwanted and unnecessary traffic can be stopped much sooner.

IGMPv2 Message format

All IGMP messages of concern to hosts have the following format:

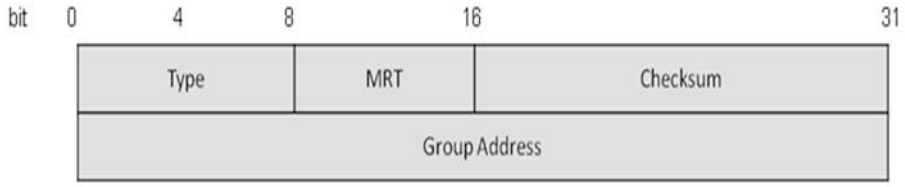


Figure 24: IGMPv2 message format

Type

There are three types of IGMP messages of concern to the host-router interaction:

0x11 = Membership Query

There are two sub-types of Membership Query messages:

General Query used to learn which groups have members on an attached network.

Group-Specific Query, used to learn if a particular group has any members on an attached network.

These two messages are differentiated by the Group Address. Membership Query messages are referred to simply as "Query" messages.

0x16 = Version 2 Membership Report

0x17 = Leave Group

There is an additional type of message, for backwards-compatibility with IGMPv1:

0x12 = Version 1 Membership Report

From now in the document we will refer to Membership Reports simply as "Reports". When no version is specified, the statement applies equally to both versions. Unrecognized message types should be silently ignored. New message types may be used by newer versions

of IGMP, by multicast routing protocols, or other uses.

Max Response Time

The Max Response Time field is meaningful only in Membership Query messages, and specifies the maximum allowed time before sending a responding report in units of 1/10 second. In all other messages, it is set to zero by the sender and ignored by receivers. Varying this setting allows IGMPv2 routers to tune the "leave latency" (the time between the moment the last host leaves a group and when the routing protocol is notified that there are no more members). It also allows knowing IGMP traffic on a subnet.

Checksum

The checksum is the 16-bit one's complement of the one's complement sum of the whole IGMP message (the entire IP payload). For computing the checksum, the checksum field is set to zero. When transmitting packets, the checksum MUST be computed and inserted into this field. When receiving packets, the Checksum MUST be verified before processing a packet.

Group Address

In a Membership Query message, the group address field is set to zero when sending a General Query, and set to the group address being queried when sending a Group-Specific Query. In a Membership Report or Leave Group message, the group address field holds the IP multicast group address of the group being reported or left.

Other fields

Note that IGMP messages may be longer than 8 octets, especially future backwards-compatible versions of IGMP. As long as the Type is one that is recognized, an IGMPv2 implementation MUST ignore anything past the first 8 octets while processing the packet. However, the IGMP checksum is always computed over the whole IP payload, not just over the first 8 octets.

Protocol Description

The term "interface" is sometimes used in this document to mean "the primary

interface on an attached network"; if a router has multiple physical interfaces on a single network this protocol need only run on one of them. Hosts, on the other hand, need to perform their actions on all interfaces that have memberships associated with them. The protocol uses some timer and counter to manage the actions, in the follow the names of them will appear in square brackets. (To know the default values look RFC 2236)

Multicast routers use IGMP to learn which groups have members on each of their attached physical networks. A multicast router keeps a list of multicast group memberships for each attached network, and a timer for each membership. "Multicast group memberships" means the presence of at least one member of a multicast group on a given attached network, not a list of all of the members. With respect to each of its attached networks, a multicast router may assume one of two roles: **Querier** or **Non-Querier**. There is normally only one Querier per physical network and it is elected by an automatic procedure. The procedure works in this way: at the beginning all multicast routers start up as a Querier on each attached network. If a multicast router hears a Query message from a router with a lower IP address, it MUST become a Non-Querier on that network. If a router has not heard a Query message from another router for [Other Querier Present Interval], it resumes the role of Querier. Routers periodically [Query Interval] send a General Query on each attached network for which this router is the Querier, to solicit membership information. On startup, a router SHOULD send [Startup Query Count] General Queries spaced closely together [Startup Query Interval] in order to quickly and reliably determine membership information. A General Query is addressed to the all-systems multicast group (224.0.0.1), has a Group Address field of 0, and has a Max Response Time of [Query Response Interval](default value is 10 sec).

When a host receives a **General Query**, it sets delay timers for each group (excluding the all-systems group) of which it is a member on the interface from which it received the query. Each timer is set to a different random value, using the highest clock granularity available on the host, selected from the range (0, Max Response Time] with Max Response Time as specified in the Query packet.

When a host receives a **Group-Specific Query**, it sets a **delay timer** to a random

value selected from the range (0, Max Response Time] as above for the group being queried if it is a member on the interface from which it received the query. If a timer for the group is already running, it is reset to the random value only if the requested Max Response Time is less than the remaining value of the running timer. When a group's timer expires, the host multicasts a Version 2 Membership Report to the group, with IP TTL of 1. If the host receives another host's Report (version 1 or 2) while it has a timer running, it stops its timer for the specified group and does not send a Report, in order to suppress duplicate Reports. When a router receives a Report, it adds the group being reported to the list of multicast group memberships on the network on which it received the Report and sets the timer for the membership to the [Group Membership Interval]. Repeated Reports refresh the timer.

If no Reports are received for a particular group before this timer has expired, the router assumes that the group has no local members and that it need not forward remotely-originated multicasts for that group onto the attached network.

When a host joins a multicast group, it should immediately transmit an unsolicited Version 2 **Membership Report** for that group, in case it is the first member of that group on the network. To cover the possibility of the initial Membership Report being lost or damaged, it is recommended that it be repeated once or twice after short delays [Unsolicited Report Interval]. (A simple way to accomplish this is to send the initial Version 2 Membership Report and then act as if a Group-Specific Query was received for that group, and set a timer appropriately).

When a host leaves a multicast group, if it was the last host to reply to a Query with a Membership Report for that group, it SHOULD send a **Leave Group message** to the all-routers multicast group (224.0.0.2). If it was not the last host to reply to a Query, it MAY send nothing as there must be another member on the subnet. This is an optimization to reduce traffic; a host without sufficient storage to remember whether or not it was the last host to reply MAY always send a Leave Group message when it leaves a group. Routers SHOULD accept a Leave Group message addressed to the group being left, in order to accommodate implementations of an earlier version of this standard. Leave Group messages are addressed to the all-routers group because other group members have no need to know

that a host has left the group, but it does no harm to address the message to the group.

When a **Querier** receives a Leave Group message for a group that has group members on the reception interface, it sends [Last Member Query Count] Group-Specific Queries every [Last Member Query Interval] to the group being left. These Group-Specific Queries have their Max Response time set to [Last Member Query Interval]. If no Reports are received after the response time of the last query expires, the routers assume that the group has no local members, as above. Any Querier to non-Querier transition is ignored during this time; the same router keeps sending the Group-Specific Queries. Non-Queriers MUST ignore Leave Group messages, and Queriers SHOULD ignore Leave Group messages for which there are no group members on the reception interface. When a non-Querier receives a Group-Specific Query message, if its existing group membership timer is greater than [Last Member Query Count] times the Max Response Time specified in the message, it sets its group membership timer to that value.

Host State Diagram

Host behavior is more formally specified by the state transition diagram below. A host may be in one of **three possible states** with respect to any single IP multicast group on any single network interface:

"**Non-Member**" state, when the host does not belong to the group on the interface. This is the initial state for all memberships on all network interfaces; it requires no storage in the host.

"**Delaying Member**" state, when the host belongs to the group on the interface and has a report delay timer running for that membership.

"**Idle Member**" state, when the host belongs to the group on the interface and does not have a report delay timer running for that membership.

There are **five significant events** that can cause IGMP state transitions:

"**join group**" occurs when the host decides to join the group on the interface. It may occur only in the Non-Member state.

"**leave group**" occurs when the host decides to leave the group on the interface. It may occur only in the Delaying Member and Idle Member states.

"**query received**" occurs when the host receives either a valid General Membership Query message, or a valid Group-Specific Membership Query message. To be valid, the Query message must be at least 8 octets long, and have a correct IGMP checksum. The group address in the IGMP header must either be zero (a General Query) or a valid multicast group address (a Group-Specific Query). A General Query applies to all memberships on the interface from which the Query is received. A Group-Specific Query applies to membership in a single group on the interface from which the Query is received. Queries are ignored for memberships in the Non-Member state.

"**report received**" occurs when the host receives a valid IGMP Membership Report message (Version 1 or Version 2). To be valid, the Report message must be at least 8 octets long and have a correct IGMP checksum. A Membership Report applies only to the membership in the group identified by the Membership Report, on the interface from which the Membership Report is received. It is ignored for memberships in the Non-Member or Idle Member state.

"**timer expired**" occurs when the report delay timer for the group on the interface expires. It may occur only in the Delaying Member state.

All other events, such as receiving invalid IGMP messages, or IGMP messages other than Query or Report, are ignored in all states.

There are **seven possible actions** that may be taken in response to the above events:

"**send report**" for the group on the interface. The type of report is determined by the state of the interface. The Report Message is sent to the group being reported.

"**send leave**" for the group on the interface. If the interface state says the Querier is running IGMPv1, this action SHOULD be skipped. If the flag saying we were the last host to report is cleared, this action MAY be skipped. The Leave Message is sent to the ALL-ROUTERS group (224.0.0.2).

"**set flag**" that we were the last host to send a report for this group.

"**clear flag**" since we were not the last host to send a report for this group.

"**start timer**" for the group on the interface, using a delay value chosen uniformly from the interval (0, Max Response Time], where Max Response time is specified in the Query. If this is an unsolicited Report, the timer is set to a delay value chosen uniformly from the interval (0, [Unsolicited Report Interval]].

"**reset timer**" for the group on the interface to a new value, using a delay value chosen uniformly from the interval (0, Max ResponseTime], as described in "start timer".

"**stop timer**" for the group on the interface.

In the following state diagrams, each state transition arc is labeled with the event that causes the transition, and, in parentheses, any actions taken during the transition. Note that the transition is always triggered by the event; even if the action is conditional, the transition still occurs.

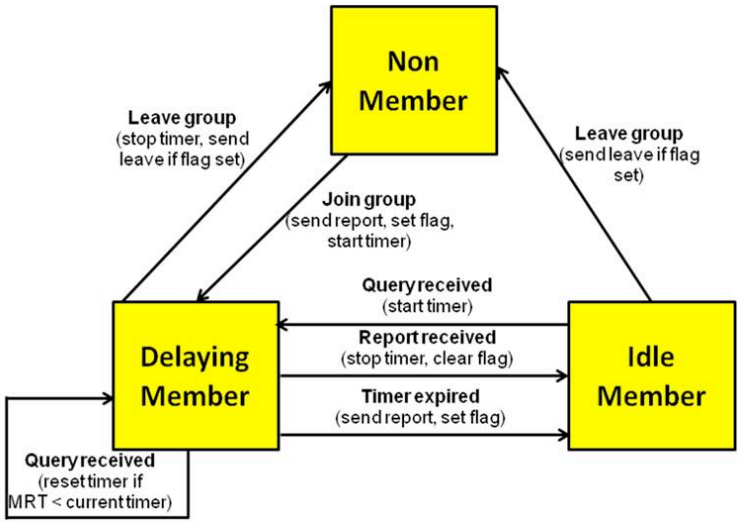


Figure 25: Host-state diagram

The all-systems group (address 224.0.0.1) is handled as a special case. The host starts in Idle Member state for that group on every interface, never transitions to another state, and never sends a report for that group.

Router State Diagram

Router behavior is more formally specified by the state transition diagrams below. A

router may be in one of **two possible states** with respect to any single attached network:

"**Querier**", when this router is designated to transmit IGMP Membership Queries on this network.

"**Non-Querier**", when there is another router designated to transmit IGMP membership Queries on this network.

The following **three events** can cause the router to change states:

"**query timer expired**" occurs when the timer set for query transmission expires.

"**query received from a router with a lower IP address**" occurs when an IGMP Membership Query is received from a router on the same network with a lower IP address.

"**other querier present timer expired**" occurs when the timer set to note the presence of another querier with a lower IP address on the network expires.

There are **three actions** that may be taken in response to the above events:

"**start general query timer**" for the attached network.

"**start other querier present timer**" for the attached network [Other Querier Present Interval].

"**send general query**" on the attached network. The General Query is sent to the all-systems group (224.0.0.1), and has a Max Response Time of [Query Response Interval].

A router should start in the Initial state on all attached networks, and immediately move to Querier state.

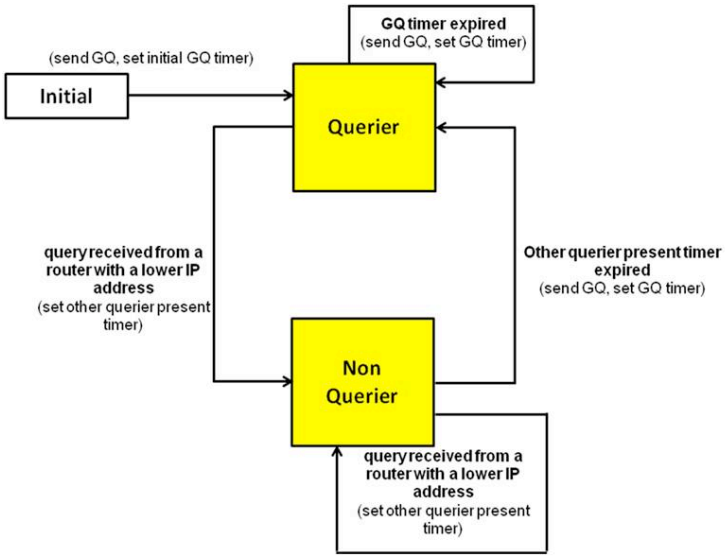


Figure 26: Router states

In addition, to keep track of which groups have members, a router may be in one of **four possible states** with respect to any single IP multicast group on any single attached network:

"**No Members Present**" state, when there are no hosts on the network which have sent reports for this multicast group. This is the initial state for all groups on the router; it requires no storage in the router.

"**Members Present**" state, when there is a host on the network which has sent a Membership Report for this multicast group

"**Version 1 Members Present**" state, when there is an IGMPv1 host on the network which has sent a Version 1 Membership Report for this multicast group.

"**Checking Membership**" state, when the router has received a Leave Group message but has not yet heard a Membership Report for the multicast group.

There are **six significant events** that can cause router state transitions:

"**v2 report received**" occurs when the router receives a Version 2 Membership Report for the group on the interface. To be valid, the Report message must be at least 8 octets long

and must have a correct IGMP checksum.

"**v1 report received**" occurs when the router receives a Version 1 Membership report for the group on the interface. The same validity requirements apply.

"leave received" occurs when the router receives an IGMP Group Leave message for the group on the interface. To be valid, the Leave message must be at least 8 octets long and must have a correct IGMP checksum.

"timer expired" occurs when the timer set for a group membership expires.

"retransmit timer expired" occurs when the timer set to retransmit a group-specific Membership Query expires.

"v1 host timer expired" occurs when the timer set to note the presence of version 1 hosts as group members expires. There are six possible actions that may be taken in response to the above events:

"start timer" for the group membership on the interface – also resets the timer to its initial value [Group Membership Interval] if the timer is currently running.

"start timer*" for the group membership on the interface – this alternate action sets the timer to [Last Member Query Interval] *[Last Member Query Count] if this router is a Querier, or the [Max Response Time] in the packet * [Last Member Query Count] if this router is a non-Querier.

"start retransmit timer" for the group membership on the interface [Last Member Query Interval].

"start v1 host timer" for the group membership on the interface, also resets the timer to its initial value [Group Membership Interval] if the timer is currently running.

"send group-specific query" for the group on the attached network. The Group-Specific Query is sent to the group being queried, and has a Max Response Time of [Last Member Query Interval].

"notify routing +" notify the routing protocol that there are members of this group on this connected network.

"notify routing -" notify the routing protocol that there are no longer any members of this group on this connected network.

The state diagrams for a router in Querier and in Non-Querier states follow:

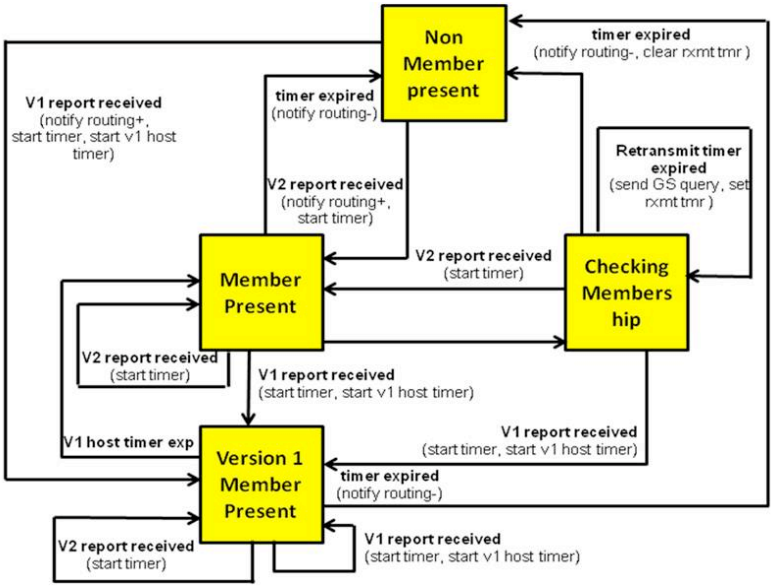


Figure 27: Router-Querier state diagram

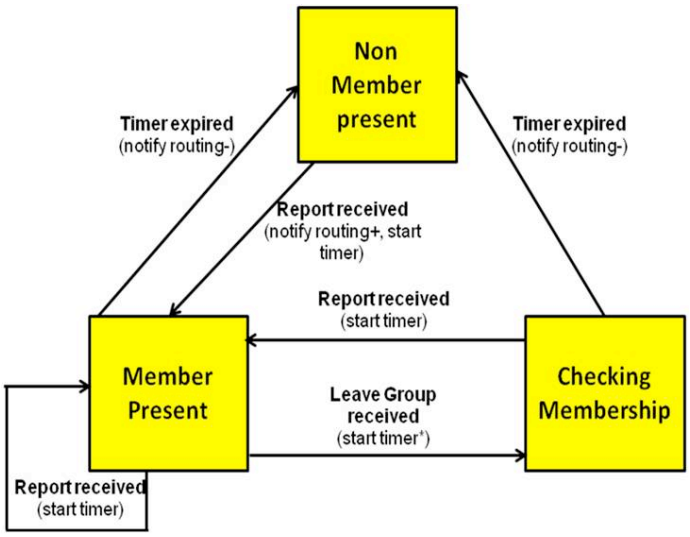


Figure 28: Router-Non-Querier state diagram

aggiungere una parte sui timers

The IGMP protocol enables **channel change** in the IPTV services. When IPTV users change the channel on their set-top box, the box does not tune to a channel as would happen with a cable system. The **IPTV set-top box is simply an IP receiver**. The set-top box switches channels by using the IGMP version 2 to join a new multicast group. When the local switch office receives this request, it checks to make sure that the user is authorized to view the requested channel then directs the routers in the local office to add that particular user to the channel's distribution list. In this way, only signals that are currently being watched are actually being sent from the local office to the DSLAM and onto the user.

In the channel Change scenario the time of zapping (switching from one channel to another) is an important aspect of quality of IPTV service. This time should tend to be close to that required with the Analog TV channel change (on the order of milliseconds). As for digital TV by satellite, this time in IPTV tends to reach the order of seconds it is due to the addition of many components of delay. The Channel Change Issue is the topic of the next chapter

.

Chapter 3

Optimizing the cost structure without compromising the service experience will enable them IPTV operators have now to face with two new challenges that give them the need to work closely with industry vendors on solving them. These challenges are:

a) the IPTV channel change (zapping) scenario

b) infrared communication speed between the set-top box and the remote control.

This chapter is focused on the fist challenge. Here there is a description of the issue and the presentation of the solution. These challenges follow the general need of the service provider that want to improve the **quality of experience** in enjoying television over IP networks. Concerning with the channel change topic the effort is on how to reach the same experience of fast channel zapping as people gets from the old analogue television. The "tune-in" time is a key indicator of the customer satisfaction. While using Analog TV this time was not an issue it became a topic to be improved in order to increase the **QoE** (Quality of Experience). As the number of channels available continues to grow, fast zapping capabilities are a key Performance Indicator for operators and end-user alike, especially when HD video comes into the equation.

1 Quality of Experience (QoE) in the IPTV scenario

IP television (IPTV) market is moving into the critical phase of large-scale commercial deployments across many regions. This may vary between carriers and geographies, but as a whole, service assurance and quality of experience (QoE) largely define this evolution. Each phase is tightly coupled with ongoing underlying technology evolution, content acquisition, and scaling the overall number of IP video subscribers.

• Phase I: Prove technical viability of technology, architecture, and basic service delivery to match existing cable and satellite TV service offerings.

• Phase II: Deliver service assurance and QoE guarantees, increase personalization,

and deliver any service, any time, in an effort to ensure and grow take rate.

• Phase III: Increase service differentiation and integration to achieve blended services and interactive TV on a large scale.

Assuring QoE for IPTV is rapidly becoming a top priority among vendors and service providers as the IPTV market evolves. In order for service providers to achieve the biggest customes base they can, the QoE of IPTV must meet and exceed what the services cable and satellite providers are currently providing. Large and small service providers are either planning for or putting the technology in place to meet the requirements surrounding service assurance and QoE. There are many factors influencing the challenge of achieving a high QoE in IPTV deployments. These factors should and will be transparent to the subscriber, who is simply expecting an always-on service offering. To that end, providing such QoE isn't necessarily simple, but it's a critical factor for the success of IPTV. It is, also, achievable with the right technology and network architecture as a foundation. Robust service control and QoE assurance must be put in place to attract initial subscribers, ensure their comfort with the new IP-based video service, and further increase the average revenue per user (ARPU) over time.

An important aspect to take under consideration is the user-interface. The primary criteria concerning with that are **speed** and **simplicity**. The consumer is used to being entertained on the sofa, with only the remote control to deal with. If the user-interface is slow, then it does not represent a satisfactory television experience. Beside that aspect there are many service delivery challenges that must be taken under consideration. In order to better understand they are listed below:

• **"Always On" Service Expectations, Hard QOS Guarantees :** The user expectation for "always on" services is based on current cable and satellite TV performance and drives end-to-end requirements from the IP STB to the end-to-end service delivery infrastructure and IPTV middleware. This entails everything from picture quality to channel-change performance to VOD availability at the time it's ordered, no matter what else is happening in the home or the network. Enhanced high-availability features in network equipment and many of the other items listed below will contribute to an always-on service

experience.

- **Network and Service Capacity Planning:** Network architecture and associated capacity planning are critical to optimize service-delivery cost, address multi-dimensional scalability, and provide QoE guarantees. Assuring congestion-free video transport across each link and node in the IPTV network is key to service quality and performance. Thus, it is vital to understand trends in capacity utilization and engineer additional capacity in time for anticipated demand, while maintaining plenty of headroom for growth. Additional provisions, such as IGMP snooping, distributed policy enforcement, IP multicast replication, flexible content insertion at the most economical points, non-stop service capabilities, and video admission control (VAC), help to cost-optimize an IPTV network without having to overprovision bandwidth.

- **Network Congestion Avoidance:** Network congestion avoidance and network capacity planning go hand in hand and are essential to avoid resource contention and minimize congestion. The first essentially depends on the second, to the extent that under-dimensioning the network is bound to result in congestion. While network oversubscription is the norm for traditional high-speed Internet and is acceptable for best-effort applications, support of deterministic H-QoS is needed to guarantee flawless QoE for triple-play, especially broadcast TV, VoD, and voice over IP (VOIP). This is not just a matter of putting in more capacity, because optimizing video content insertion by placing popular VoD content closer to end users helps avoid network congestion in the end-to-end video delivery path. It also encompasses tracking of capacity utilization and resource availability state changes (e.g., DSL training-rate variability in the first mile or potential fiber cuts in the metro aggregation network affecting second- or third-mile bandwidth) to anticipate and remedy potential resource contention before it occurs.

- **End-to-End QoE Measurement and Assurance**: While proper capacity dimensioning and video admission control mitigate the risk of service quality or availability degradation as a result of network congestion, the only way to fully insure flawless perceptual QoE is by adding components to measure and verify IP video and audio quality, disturbance rate, and channel-change delay. Many vendors supply standalone solutions to measure and

monitor video traffic, while others are integrating and/or partnering to provide an end-to-end solution taking advantage of network diagnostics and performance statistics, IPTV client measurements, middleware reporting, and MPEG-2/4 traffic analyzers and video monitors.

- **Video Access Control:** VAC is rapidly becoming an important new requirement for telecom equipment. It is most applicable to control dynamic admission of unicast VoD sessions, as VoD is a pay-per-use service and also the most bandwidth-intensive (i.e., the most likely to cause resource contention when unchecked). Multicast VAC is being proposed for cases where more broadcast channels are being offered than can be concurrently watched. Multicast VAC prevents bandwidth issues when changing channels, for example when bringing up a new channel (one that has not been previously watched). This may potentially result in some of the least-watched channels being unavailable, while ensuring that more popular channels are available with the expected quality. VAC is essentially a "safety valve" to ensure flawless QoE for video streams by preventing additional streams from entering the network when remaining capacity is insufficient to support them (e.g., extreme VoD, concurrency peaks with a new movie release, or reduced capacity due to failures). While VAC is a compromise between service quality and availability, the need to deny service requests due to insufficient network capacity should be an exception, not the rule. However, since the possibility of network congestion can never be ruled out, the implementation of an effective admission control strategy is an important issue involving network infrastructure, policy control systems, and IPTV middleware.

There are many additional service delivery challenges for IPTV service providers. Among them there are regulatory issues, franchise requirements, working with local permitting agencies, vast and unique content rights, lack of industry standards, and the integration of multiple IPTV hardware and software service-delivery components.

The Television scenario is changing: before the end of 2010 Italy will change from the Analog television to the Digital TV. The digital technology T-DVB will become the only one used. **DTT** is an implementation of digital technology to provide a greater number of channels and/or better quality of picture and sound using aerial broadcasts to a conventional antenna instead of a satellite dish or cable connection. There are many advantages and also some disadvantages related with the switch-off and they are listed below:

Advantages:

- Digital reception tends to be better overall, particularly with a good signal. With a weaker signal there is little perceptible difference, in fact analogue can be better.
- It is easier to obtain the optimum digital picture than the optimum analogue picture.
- Many more channels can fit on the digital transmission.
- Interactive (red button) services can be provided.
- Disadvantages:
- New equipment (set top box) may be required.
- Increased electricity consumption by the digital receiving equipment.
- An upgraded antenna installation may be required.
- Analog requires lower signal strength to get a watchable picture. By extension, digital does not degrade as gracefully as analog.
- **Switching channels is slower** because of the time delays in decoding digital signals.

This last disadvantages is related with our focus. In the switch-off scenario the IPTV is an alternative of the DTT. According with the research society "Screen Digest" this service, in Europe, is going to have, at least, 9 millions of new users in the next four years. There is also the expectation that this kind of service will be totally managed by the ISP.

2 Triple Play Service

Innovation is rapidly changing the television landscape, expanding the user's experience and service expectations. Service providers around the globe are aggressively staging and deploying converged networks that can deliver data, voice, and video services over a single broadband connection. IPTV is the effective delivery of the video component of **Triple Play Service.** The three services that are the actors in that solution can be positioned

according to their average bandwidth per subscriber and whether the service is managed or not. These are the drivers in the figure below respectively along the vertical and the horizontal axis. IPTV, for example, is a managed service with high QoS, high sustained bandwidth per subscriber, and high concurrency. By comparison, HSI access service is unmanaged in the sense that the service definition offers best effort access to a pool of bandwidth shared by the Internet community. Finally, Voice over IP (VoIP) service refers to digital voice offering and is a managed service with low average bandwidth per subscriber.

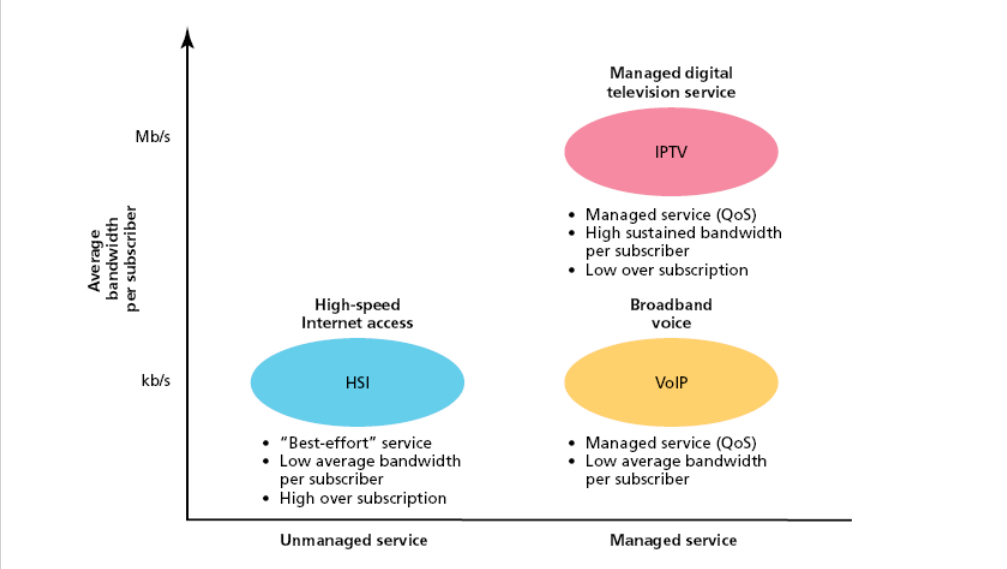


Figure 29: Triple Play Service definition

The new solution proposed by Alcatel Lucent concerning the triple play service is the TPSDA 2.0. This solution comes from the previous one with many enhancements added to the TPSDA that was selected by more than 55 operators. TPSDA is the blueprint architecture to accelerate the IP network transformation for delivery of video, voice, data and entertainment services. Illustrated in the figure below (Figure 30), TPSDA is an end-to-end architecture designed from the ground up to deliver high subscriber scale, high bandwidth throughput per subscriber and high concurrency.

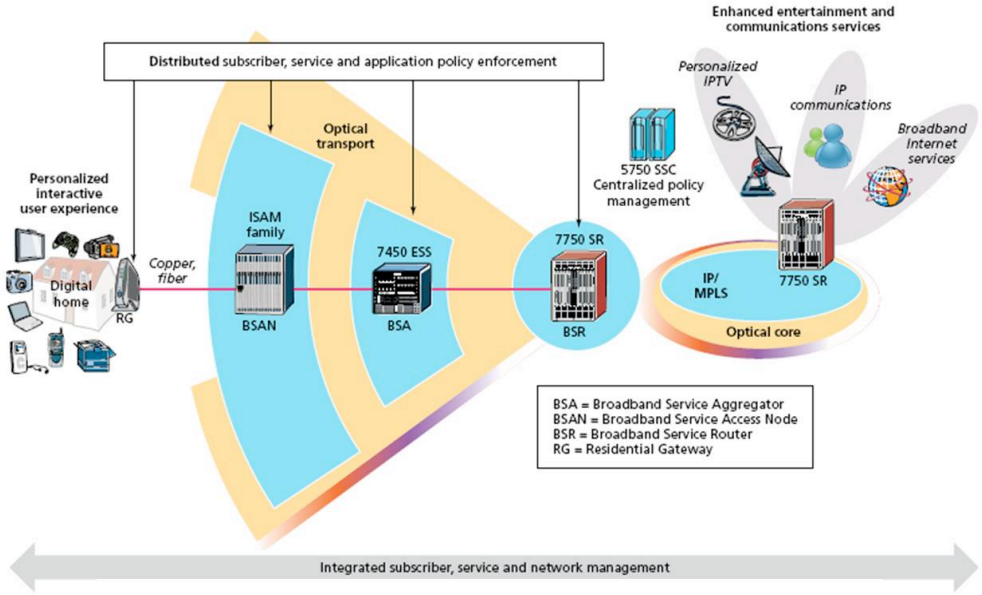


Figure 30:Triple Play Service Delivery Architecture

TPSDA is a significant upgrade to first-generation broadband network solutions, which simply tunnel best-effort Internet traffic across the network. By contrast, TPSDA leverages all parts of the network to deliver and enforce fine-grain service policy end to end. This enables IPTV operators to deliver multiple services to subscribers, including managed services such as IPTV, voice and managed online services, as well as unmanaged services such as High-speed Internet (HSI). TPSDA 2.0 begins by addressing linear TV delivery and its challenges with HD and multi-room offers. Solving these challenges is crucial to enabling an operator's baseline portfolio of IPTV services. Moreover, the technology used here provides the foundation for more **advanced personalized** and **interactive** services. With linear TV programming (that is, traditional broadcast TV delivered over IP), all channels are delivered to the edge of the operator's network, at which point only the channel (or channels) being watched in a given household are switched onto that subscriber's broadband connection. Using IP multicast technology, a single copy of each channel is pushed onto the network at the video office and thereafter the channels are replicated as required within the network until each is distributed to the edge of the network. Multicast is extremely efficient in its use of bandwidth.

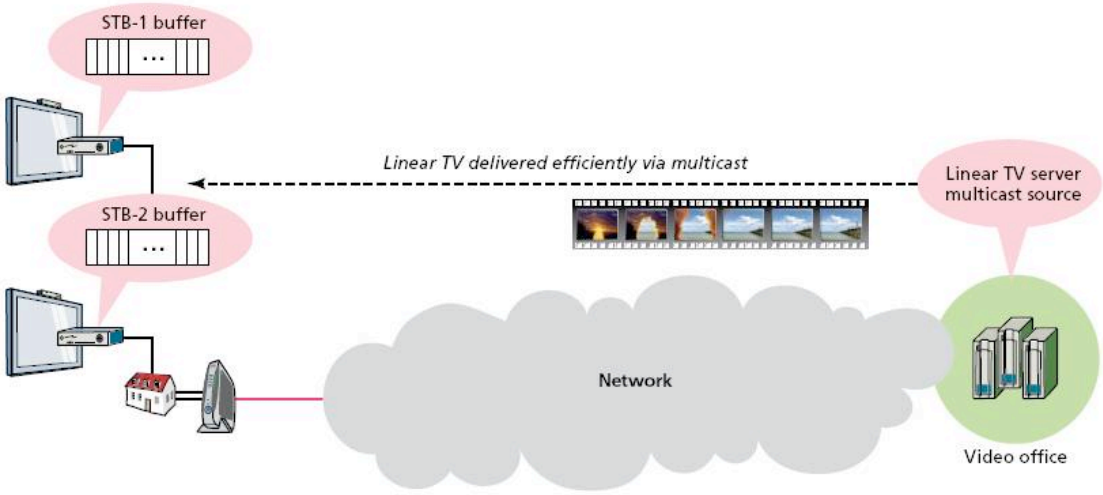


Figure 31: Efficient linear IPTV delivery via IP multicast

The process of switching a channel onto a subscriber's broadband connection is triggered by the subscriber pressing the remote control. When detected by the set-top box (STB), a signal is sent into the network and the requested channel is switched onto the subscriber's broadband connection. The emergence of ubiquitous HD content and multi-room offerings brings new challenges to IPTV operators. Both increase the amount of bandwidth to each household, increasing the likelihood of transmission errors. Since linear IPTV is UDP-based, there is no inherent mechanism to retransmit the damaged video content. As a result, visual artifacts appear on the TV screen when errors occur, deteriorating the user experience. In order to overcome these issues the Alcatel solution proposes FCC and RET. These are application-level techniques developed to avoid jeopardizing the user experience. Illustrated in Figure 32, centralized server-based implementations of FCC and RET are triggered in the home by the user and STB respectively. When the user changes a channel, a message is sent into the network to have the new channel switched onto the broadband connection via multicast (this part remains the same). An FCC server, however, also detects the channel change request and sends a burst of unicast video content, beginning with the I-frame, to the subscriber's STB with enough information to allow an immediate channel change along with several seconds of video information to play out while the STB synchronizes with the new multicast stream.

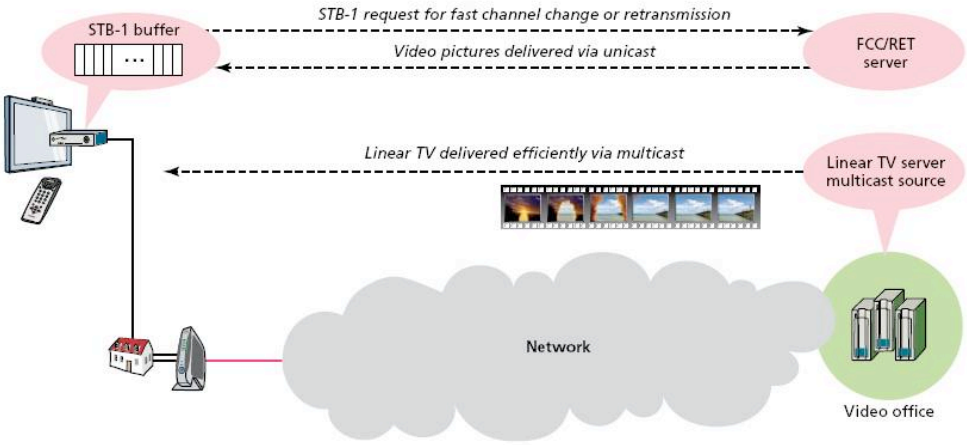


Figure 32: Linear TV delivery with centralized FCC and RET

RET works in a similar manner except that the STB triggers the retransmission request. When a STB detects a video error in its buffer, it requests a retransmission of the missing content and the end user never sees a visual artifact. A centralized server-based FCC/RET implementation overcomes the challenges outlined and is sufficient for low concurrency levels. The issue with centralized server-based FCC/RET implementations, however, is the amount of unicast bandwidth used: eight times more than the total bandwidth required to transport all multicast channels in the first place. Based on internal modeling of a typical tier 1 IPTV operator, the amount of unicast bandwidth required for FCC/RET is approximately 12.5 Gb/s whereas the total multicast bandwidth is approximately 150 Mb/s. Illustrated in Figure 33, this introduces a new dynamic to the economic equation: the cost to transport unicast traffic from servers located deep in the network to specific subscribers connected at the very edge of the network.

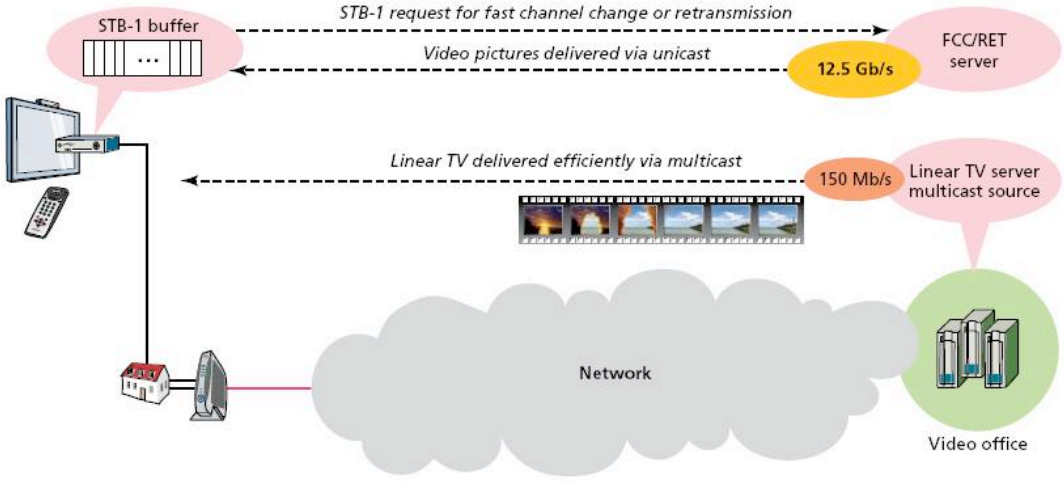


Figure 33: Centralized FCC and RET requires eight times more bandwidth than all multicast channels

Enhanced with application layer intelligence, TPSDA 2.0 is able to deliver immediate channel changes and retransmission with considerable cost savings while assuring a superior, uninterrupted viewing experience. Based on TPSDA 2.0's distributed and integrated FCC/RET function in the network elements, the bandwidth in the core network is reduced to zero; in the aggregation network, up to 1 Gb/s of bandwidth is saved; and on the precious broadband access node uplink, hundreds of megabits of bandwidth are saved. This result is illustrated in the figure below see Figure 34.As seen above the FCC/RET demonstrate that the volume of unicast bandwidth generated is huge but this is really just the tip of the iceberg, a sign-post of what's to come. As operators deploy more personalized services with greater interactivity, the amount of unicast traffic will increase. In order to keep IPTV as a winning proposition for service providers, they need an evolving architecture that enables them to take IPTV to the next level, beyond initial linear TV offerings and into the realm of personalized and interactive applications. The increase of the number of the new services impose future massive bandwidth demand on the network especially when used in high concurrency. With the cost of memory and processing resources declining faster than the

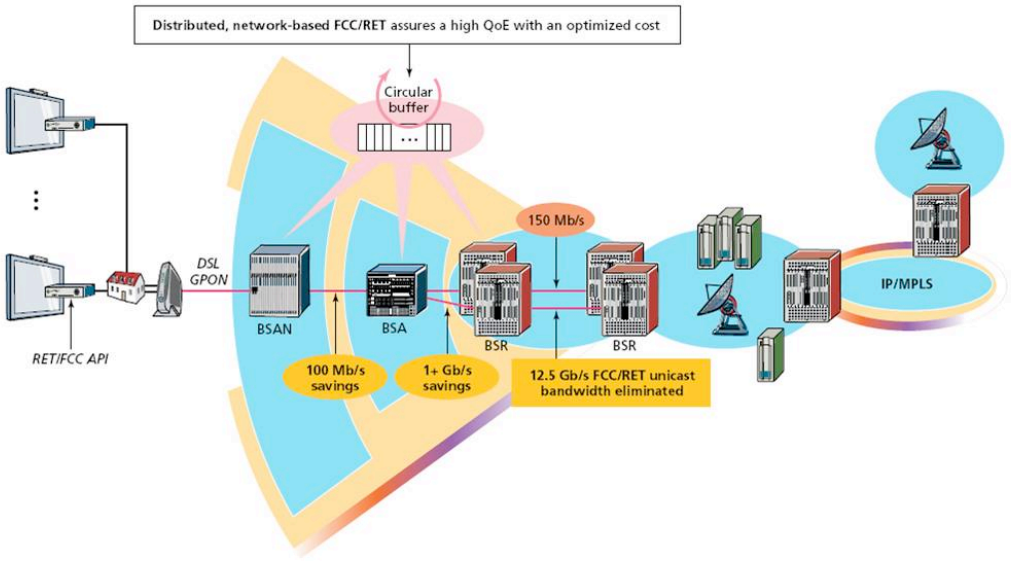


Figure 34: TPSDA 2.0 assures a high QoE with an optimized cost

cost of transport, market forces are converging, creating new opportunities to innovate and rethink traditional approaches to content distribution architectures. As is illustrated in the Figure 35, the traditional approach that is centralized storage, is moving toward **decentralized server solutions**, exchanging costly transport dollars for inexpensive memory dollars. Furthermore, the cost of real estate for decentralized servers combined with their power, cooling, interconnect ports and maintenance costs continues the transition, pushing storage into the network, closer to subscribers. By caching content at the edge of the network, unicast traffic is kept local and does not incur the high transport cost associated with centralized storage deep in the network. Furthermore, by pushing the cache and supporting processing requirements closer to subscribers, the performance requirements are lower as fewer subscribers must be supported per element, providing a consistent and high-performance user experience.

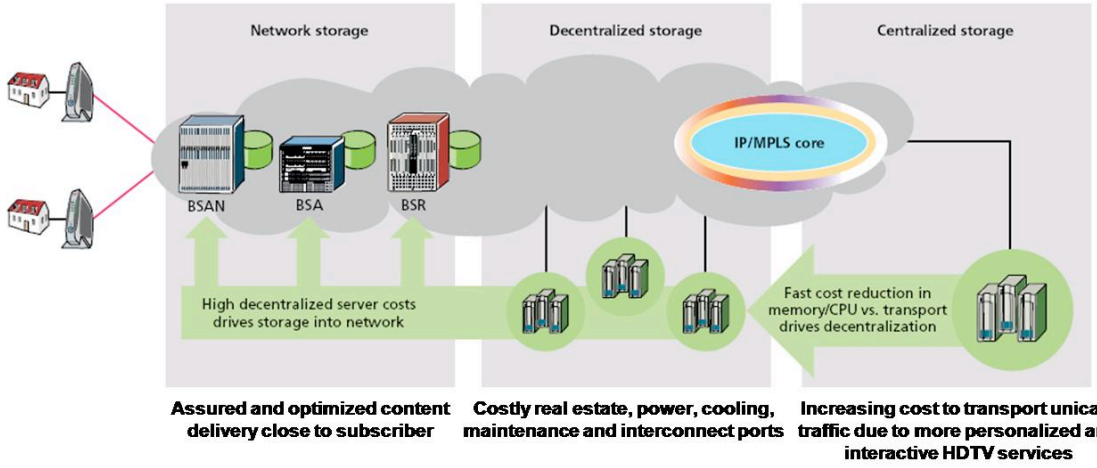


Figure 35: TPSDA 2.0 leverages declining memory/CPU costs to enable new, network-based content delivery architectures

3 Channel Change Issue

IPTV's current inability to change between channels in a timely fashion is, in the mind of vendors, one of the single biggest problem to solve. Channel changing or "channel zapping" in the IPTV domain is the equivalent of surfing channels on a television set in the traditional sense. Channel changing, then, always entails leaving one TV channel and joining/watching the next. In the traditional cable television network, channels were always "present" on the wire. For this reason, channel changing performance was generally not an issue. Switching from one channel to the next was usually instant, typically was not a source of user dissatisfaction. For satellite TV providers, on the other hand, the inherent distance that the video signals had travel always introduced a "delay" in switching between channels. The usual approach to reduce the perceived delay was to provide some kind of on-screen feedback when switching channels so that the user knew the channel change request had been received. Now, with video being delivered over an IP network, one that was not necessarily designed with video transport as a key goal, several optimizations and novel methods are being deployed throughout a providers' network to improve the perceived "instant" channel change behavior that is expected by traditional television consumers.

Conventionally, with IPTV, broadcast video is sent using multicast streams. To start viewing a channel, the user (client device) joins the stream. However, video entry points only occur periodically in the stream, such as every 0.5 sec or every 2-8 seconds. This means that when a client joins the stream, it might wait 2-8 seconds before it can start presenting video frames. To compete effectively with conventional TV, the time required to change channels must be minimized.

3.1 Channel changing requirements

The channel changing performance in an IP network is primarily the result of end-to-end processing of associated multicast protocols, packet switching, and the end-device's ability to decode or "show" the video with relative consistency. A deployed system must be able to performance predictably in a sustained mode. A sustained mode of operation refers to a realistic load condition, not a best-performance condition with light use. In addition to sustained performance, a deployed system must be resilient to fault conditions where peak loads may easily exceed the sustained performance limits. For a system to be resilient, it must be able to perform acceptably well and be able to recover from an overloaded condition. Channel surfing presents varying load conditions for a multicast network. For example, if there is a sudden power outage in a neighborhood on a typical Sunday "game day," a mass flood of traffic will be injected into the IP network requesting network configuration information followed by several channel changes to tune back to a desired channel by hundreds, if not thousands, of viewers at almost the same time. Testing such realistic load conditions is critical in ensuring optimal QoS on the network and help tune device settings.

With no fast channel change (FCC), users face a degraded experience in that a delay occurs between changing the channel and seeing video. The delay is particularly a problem if the user is channel surfing using the channel up/down buttons.

The FCC method adds an FCC server at or near the edge of the network. The method can be implemented in several ways, including a simpler multicast-to-unicast model and a multicast-only model. With the multicast-unicast model, an FCC server is placed in the access network near the serial device (SER) or the optical line terminator (OLT), if bandwidth is

sufficient. The FCC server receives all channels that support fast channel changing and buffers an amount of each stream (such as 8-seconds). The server outputs unicast streams from the buffer to the clients. When a client joins a new stream, the data from that stream's buffer is taken, starting at the most recent entry point in that buffer. As a result, the client does not wait for the next entry point to arrive.

3.2 Causes of the delay

The delay in "tune-in" acquisition of live streams is the time between the channel change and the initiation in rendering video in the screen. This delay has many causes and finding a solution to solve them is an open issue and many persons and associations are studying on these topics.

Some of the causes are:

- ✓ STB time for leaving a previous channel and resets the decoder
- ✓ RTSP negotiation (if requesting streams or stream data)
- ✓ IGMP session joining (if tuning to a multicast session)
- ✓ Video and audio random access points (**RAP**) acquisition
- ✓ Client stream buffering (to de-jitter reception, to allow time for re-transmission, acquisition of complete forward error correction blocks, or to handle video packet re-ordering)
- ✓ Synchronization between streams (RTCP sender report)
- ✓ Encryption key acquisition
- ✓ End-system delays, such as processing delays
- ✓ Network latency

Some of these delays can be mitigated, while others, like network latency, cannot. Below there is a description of these delay factors in detail.

3.2.1 RTSP negotiation

RTSP is a protocol that can be used to initiate tune-in to a multicast stream or to request a unicast stream from a server. When it is used, it can take several packet round-trips of request/response before the RTSP server sends the first media packet. In some situations, this can be significant.

3.2.2 IGMP session joining

If the content is multicast on the network, then there are typically no processes operating at the RTP level between the source and the client. Multicast is an efficient distribution protocol for live streams as each network link needs carry only one copy of each stream. Routers in the network plicate streams from inputs onto the output links which have one or more clients. Clients indicate their interest in the multicast by sending a special control packet (IGMP), which is intercepted by the router. The routers in turn refer to each other to find, and forward, the packet stream.

This find-and-forward process (called multicast 'join') can, in some circumstances, take some time.

3.2.3 RAP acquisition

Video is classically 'differentially coded'. Very few coded frames contain all the information needed to create a complete set of decoded pixels. Instead, frames are coded as differences from one or more other frames. In order to handle both recovery from loss and tune-in to live streams, 'independently decodable' frames or I-frames can be sent periodically. Decoder refresh frames or IDR-frames are I-frames that have the additional property that they mark a division of the sequence; frames displayed after the decoder refresh do not depend on frames before it. They are true Random Access Points (RAPs). Because these frames are so much larger (in bytes) than differentially coded frames, they are sent rarely in order to keep the bit-rate low. A delay occurs when tuning into a new stream, since the decoder must wait for a RAP before it can start displaying video. In addition, their size often means that traffic smoothing causes adjustment of their send time and that of adjacent packets.

3.2.4 Client buffering

If there is deviation in the arrival time of packets from the relative timing that they would have if they were to arrive just-in-time to be played, then a de-jitter buffer is needed if we are to avoid under-run (starvation).

This jitter has two causes: deliberately introduced jitter from the source, for traffic smoothing, and network introduced jitter (e.g. caused by cross-traffic in network equipment). The source-introduced jitter tends to occur most, or even exclusively, in video, where the variation in coded frame size (in bytes) can be large (e.g. I-frames can be many times as large as B frames). A buffer is also needed if there is to be time to perform re-transmissions. A delay occurs during the time the client fills its buffer before starting to render.

3.2.5 Synchronization between streams (RTCP sender report)

RTP streams have 'free floating' timestamps – they have arbitrary origins (and indeed, usually different streams have different tick-rates). In order to synchronize audio and video, their time stamps have to be related. Associated with each stream are periodic RTCP sender reports, which associate the RTP timestamps in the stream with a common clock at the transmitter (usually the time-of-day clock, but actually any clock is permitted as long as it is common to all streams). Until at least one RTCP sender report has been acquired for each stream, the streams cannot be played in synchronization. The only exception to this is that the RTSP control protocol has provision for sending the initial synchronization information at the beginning of a play interval. However, RTSP is not used in all situations. Thus, the frequency and timing of RTCP reports often contribute to the delay before audio and video are rendered, not just to their synchronization, because many clients will not render anything before synchronization has been established. Packet level, or Application Layer, Forward Error Correction, if used is usually applied to *blocks* of packets of the stream. Play out of the stream must be delayed at the client by a time equal to the largest such block in the stream, to allow time for blocks to be corrected if there is packet loss. Additionally, play out usually would not be started until the first packet of an FEC block, since lost packets before this point in the stream cannot be corrected by the FEC. FEC codes may be systematic or non-systematic. In

the case of systematic codes, the original data is sent followed by a number of "repair" packets which can be used at a received to recover original ("source") packets which were lost. In some cases, insufficient data may be received to perform the FEC recovery operation, in which case the received source packets may be passed to the A/V decoders for play out, potentially with loss concealment. In the case of non-systematic codes, the original data packets are not sent and instead only FEC encoded data is sent. In this case, if the FEC recovery operation is not successful then no data is available for play out. Systematic codes thus have the advantage that play out with loss concealment may still be possible even if the forward error correction is unsuccessful. Further advantages of systematic codes are that in low loss scenarios it may not be necessary to apply the forward error correction decoding algorithm at the receiver, thus lowering the overall computational load of the FEC and that source data can be transmitted without waiting for the FEC encoding operation to complete, reducing the end-to-end delay. Since the first packet available for play-out is generally the first packet of an FEC block, it is advantageous if FEC blocks can be aligned with the random access points (e.g. Groups of Pictures) so that in video this first packet is generally an IDR frame. This significantly reduces the additional play out delay incurred as a result of the introduction of FEC, since the player needs to wait for an IDR frame in any case. FEC may be applied at a location different from that performing the actual video encoding. Therefore codec-independent identification of IDR frames within the stream is necessary to avoid re-parsing of the video encoding to identify IDR frames for the purpose of aligning FEC blocks. Additionally, FEC is often applied after content encryption in which case IDR identification through video content parsing is impossible.

3.2.6 Encryption keys acquisition

If the content is encrypted, then the decryption keys must be acquired. Generally it is not desirable to acquire keys too far ahead of need (either for the future of the current stream, or for other potential streams). Since this problem is both independent of RTP, and dependent on the key-exchange method used, it is not discussed here.

3.2.7 Processing delays

Processing delays can occur in the terminal at a number of layers – network, RTP, codec, and so on. In general, this is a trade-off between terminal resources (memory, processor speed) and cost. However, there are recommendations that can be made to minimize the processing load on the end-system, and hence the delay.

In order to understand the causes of each delay there is a summary of them in the figure below.

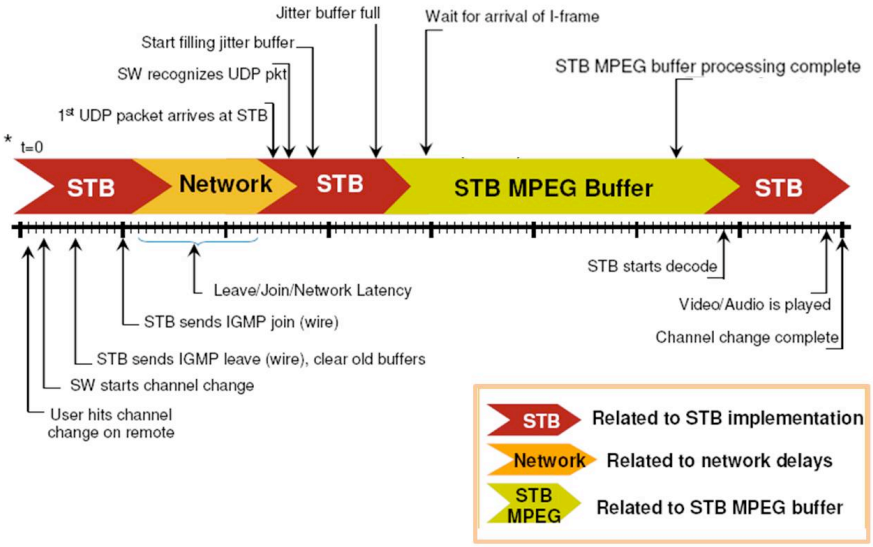


Figure 36:Channel Change event summary

3.3 A Benchmark for Channel Change in IPTV

Starting from the awareness that it is difficult to weight the costs and complexity versus benefits of any additional infrastructure, the aim of this paragraph is to demonstrate the lower limits achievable by pure multicast video transport without additional infrastructure.

- we show examples, where an IPTV channel with the same encoded content has significantly different channel changing times when streamed (multiplexed) slightly differently but with the same bandwidth

- we show under which conditions are the initial buffering delays minimal for a given bandwidth.

In order to compare the channel change times of IPTV channels, we are going to introduce a **model framework** for this scenario. An IPTV channel consist of n elementary streams (video, audio, different enhancement layers in scalable coding, etc.). The n elementary streams are encoded independently and piped into a streamer that packetizes and streams the channel via IP-multicast either as a single multiplex (e.g., with an MPEG-2 transport stream) or as a collection of separate elementary streams (e.g., over real-time transport protocol). In the former case, the channel will be associated with a single multicast address and port, in the latter case, it will be assigned a collection multicast addresses and/or ports. Our model is shown in Figure 37. In the figure there is an IPTV channel with *n* elementary streams. Notations for sample *i* in elementary stream *k*: s_i: sampling time; $\delta_i^{E,K}$: encoder delay; $\delta_i^{S,K}$: streamer delay; δ: network delay; τ^K_i: arrival time at the decoder; $\delta_i^{D,K}$: decoder delay including synchronization; t_i: play-out time after synchronization. The end-to-end latency $t_i - s_i$ is constant.

Each media (video, audio, etc.) sample will enter the system at a given sampling time s_i and undergo several stages involving different delays until it is displayed at the receiver side at play-out time t_i. To be more precise, we use the convention that all delay values indexed with *i* pertain to the first bit of each encoded sample. To facilitate the analysis, the sampling times are aligned in the following way: each media stream has the same sampling times and sample durations, which means for example that several audio samples are collected together to form a block of samples with the same duration as the corresponding video picture. The delays from Figure 37 are explained as follows:

- Encoder delay: each elementary stream encoder maintains a buffer for rate control, which introduces a delay.
- Streamer delay: the streamer may introduce buffering for multiplexing, rate shaping, etc.
- Network delay: in typical IPTV scenarios, each elementary stream is forwarded along the same multicast distribution tree. Therefore, we assume that the network delay is equal

for all elementary streams. The value of will be taken as constant. This assumption is valid for networks with reasonable quality of service capabilities and receivers with a corresponding jitter buffer.

- Decoder delay: each elementary stream decoder maintains a buffer for the incoming encoded samples. In general, each elementary stream sample will undergo different encoder and streamer delays. It is necessary to align these samples for play-out. The decoder buffering must account for this synchronization delay as well.

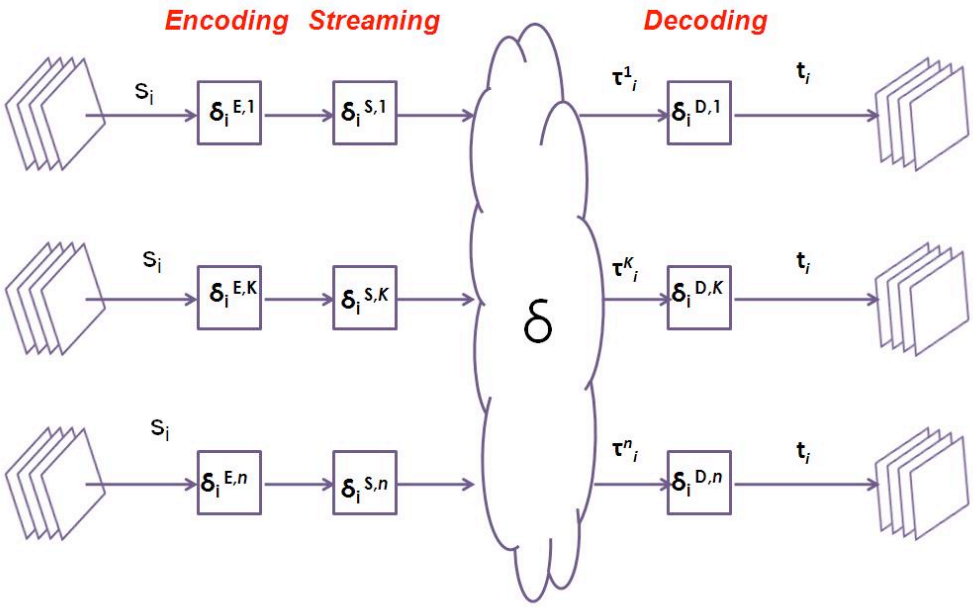


Figure 37: IPTV channel

We ignore the delays corresponding to sample reordering, which is typical in some video encoding standards. For our analysis, we assume that the samples are encoded, sent, decoded, and displayed without reordering. The computational time needed for the actual decoding is assumed to be zero as usual in such models. Note that the delays from Figure 37 are not independent. In particular, the decoder delay is expressed as:

$$\delta_i^{D,K} = t_i - \tau_i^k$$

Equation 1

There is a lower bound for this delay which is obtained as follows. The first bit of the

encoded sample *i* from elementary stream *k* arrives at the receiver at time τ_i^k. The first bit of the next encoded sample *i* + *1* arrives at τ_{i+1}^k.. Assuming that the bits of the sample *i* are arriving in this time period with rate R_i^k, we obtain:

$$\delta_i^{D,K} \geq \tau_{i+1}^k - \tau_i^k = \frac{b_i^k}{R_i^k}$$

Equation 2

where b_i^k is the size of the encoded sample i in elementary stream *k*.

3.3.1 Modelling the Channel Change

When a channel change is requested, the receiver joins one or more multicast groups and starts to receive the corresponding elementary streams. The receiver listens until it gets to the next random access position. Assuming that the random access position corresponds to the sample *i*, the receiver will get to this position at time $\min_{k=1,\ldots,n} \tau_i^k$. After this, the receiver will wait until the encoded sample *i* arrives in each elementary stream and continue to buffer the streams until t_i. Finally, the samples are instantaneously decoded and played-out. The resulting initial buffering delay is the difference $t_i - \min_{k=1,\ldots,n} \tau_i^k$.

What is the earliest possible play-out time t_i? Considering Figure 37 and Equation 2 we obtain the lower bound:

$$t_i \geq \max_{k=1,\ldots,n} \left(s_i + \delta + \delta_i^{E,k} + \delta_i^{S,k} + \frac{b_i^k}{R_i^k} \right)$$

Equation 3

To sustain a constant frame rate, the value of the end-to-end delay $t_i - s_i$ must be constant. The smallest possible value for t_i for which this difference is constant is given by:

$$t_i = s_i + \delta + \max_{k=1,\ldots n} \Delta_k$$

Equation 4

where

$$\Delta_k = \max_i \left(\delta_i^{E,k} + \delta_i^{S,k} + \frac{b_i^k}{R_i^k} \right)$$

Equation 5

If we define the total delay $\Delta := \max_{k=1,\ldots,n} \Delta_k$ then the end-to-end latency of the system can be written as:

$$t_i - s_i = \Delta + \delta = \cos t$$

Equation 6

We have derived the earliest possible synchronized play-out time t_i and the corresponding initial delay is:

$$d_i = t_i - \max_{k=1,\ldots,n} \tau_i^k = \max_{k=1,\ldots,n} \left(\Delta - \delta_i^{E,k} - \delta_i^{S,k} \right)$$

Equation 7

From now on, we will refer to these values for t_i and d_i as the earliest play-out time and the corresponding initial delay. Note that the receiver cannot compute the default play-out time t_i from the encoding parameters only. It depends on the streaming scheme and the streamer must signal these values to the receiver. However, the streamer can only compute $t_i - \delta$, since the exact network delay is unknown. The usual way to avoid these difficulties is to signal $\delta_i^{D,k} = t_i - \tau_i^k$, since this is independent of the network. The receiver can measure the arrival time τ_i^k of the corresponding packet and reconstruct $t_i = \delta_i^{D,k} + \tau_i^k$. Note that the streamer and the receiver must have the same notion of time duration, which is achieved by clock synchronization. The second consequence is that without additional infrastructure, the lower bound for the channel change time is given by the sum of **two delay factors**:

- waiting for the next random access position
- initial buffering.

Our goal is to investigate the possibility of reducing this time by optimizing the

streamer delays. Since we cannot influence the periods between the random access points (this is internal to the encoders), we can only try to reduce the buffering delay by optimizing the values $\delta_i^{S,k}$. We consider two examples for streaming the same IPTV channel and quantify the underlying initial buffering delays

Minimum latency streaming

Assume that we stream the n elementary streams independently, so, we do not introduce streaming delays, they are set to zero. This is done in order to achieve the minimum latency for each elementary stream individually. This is typical for real-time communication systems.

Equation 5 reduces to:

$$\Delta_k = \max_i \left(\delta_i^{E,k} + \frac{b_i^k}{R_i^k} \right)$$

Equation 8

The initial buffering for the synchronized play-out becomes:

$$d_i = \max_{k=1,\ldots,n} \left(\Delta - \delta_i^{E,k} \right)$$

Equation 9

with $\Delta = \max_{m=1,\ldots,n} \Delta_m$ as usual.

As an example, if the maximal video buffering is $1.8s$ and the maximal audio buffering is equal to $40ms$, then the initial buffering delay will be equal to $1.76s$ after synchronization.

DVB-typical streaming

Consider the minimum-latency streaming as introduced above. We modify this scheme by introducing non-zero streaming delays. For example, in typical MPEG-2 transport streams multiplexed for digital video broadcasting (DVB), the audio samples are delayed relative to the corresponding video samples at the streamer. For the n elementary streams from the minimal-latency example, we introduce the modified streaming

delays: $\delta_i^{S,k} = \Delta - \Delta_k$ ad so, equation 5 becomes:

$$\Delta_k^* = \max_{k=1,\ldots,n}\left(\delta_i^{E,k} + \delta_i^{S^*,k} + \frac{b_i^k}{R_i^k}\right) = \Delta$$

We obtain a new value for the total delay. The initial Buffering for the synchronized play-out becomes:

$$d_i = \max_{k=1,\ldots,n}\left(\Delta^* - \delta_i^{E,k} - \delta_i^{S^*,k}\right) = \max_{k=1,\ldots,n}\left(\Delta_k - \delta_i^{E,k}\right)$$

Equation 10

The initial buffering delay will be significantly reduced compared to the minimum latency streaming.

Reducing the initial buffering by aggregation

Let R_k denotes the bandwidth (peak bit rate) of the elementary stream k and let $R = R_1 + \ldots.. + R_n$ be the bandwidth of the IPTV channel.

Consider the modified streaming delays:

$$\delta_i^{S^*,k} = \Delta - \delta_i^{E,k} + \sum_{m=1,\ldots,n} \frac{R_m}{R}\left(\delta_i^{E,m} + \delta_i^{S,m}\right)$$

Equation 11

With these streaming delays, the bandwidth does not exceed R. The end23nd latency becomes $\Delta_A + \delta$ where $\Delta_A = \Delta + \sum_{k=1,\ldots,n} \frac{R_k}{R}\Delta_k$.

The initial buffering delay of the modified system is:

$$d_i^* = \sum_{m=1,\ldots,n} \frac{R_m}{R}\left(\Delta_m - \delta_i^{E,m} - \delta_i^{S,m}\right) \leq d_i$$

Equation 12

The modified arrival times have the property that $\tau_i^{k*} = \tau_i^{m*}$ for all the k and m, the encoded samples from each elementary stream are sent and arrive at the same time. A

streaming with this property is sometimes called an aggregation scheme, meaning that the encoded samples corresponding to the same sampling time s_i are streamed jointly, they can be considered as a single ”aggregated” encoded sample. The proposition states that we can in principle always improve the streaming in terms of initial buffering by aggregation We demonstrate the improvement obtained over the minimum-latency and DVB-typical streaming. We have shown that the initial buffering times can be drastically different depending on the streaming scheme used.

We have shown how to optimize the multicast delivery in order to reduce the initial buffering. The better result is given by the aggregation that is the last case we examined. The initial buffering time is still too much high and so how can we do in order to go beyond these values?

One solution could be to increase the bandwidth of the multicast channel. Considering the bandwidth R of the IPTV channel, nothing speaks against increasing this bandwidth beyond the original one. The aggregation is done according to the specified bandwidth R. We observe that the initial buffering, according to Equation 12, will actually tend to zero if we increase the available bandwidth for the multicast channel. Using Equation 12, we can derive a worst case bound on the channel change time when the available bandwidth is R as:

$$T(R) = \frac{\sum_{k=1,\ldots,n} R_k \Delta_k}{R} + T_0$$

Equation 13

where T_0 denotes the maximum difference between the arrival of two consecutive random access points. This reciprocal law is sketched in Figure 38, where the upper bound on the channel change time is shown. This curve, computed for the specific channel parameters could be used as a benchmark. Assuming that it is impossible to eliminate the delay due to the waiting for the next RAP without using additional infrastructure so, any method for fast channel change should have better performance in terms of channel change time versus bandwidth than the benchmark with multicast delivery in order to justifying additional infrastructure.

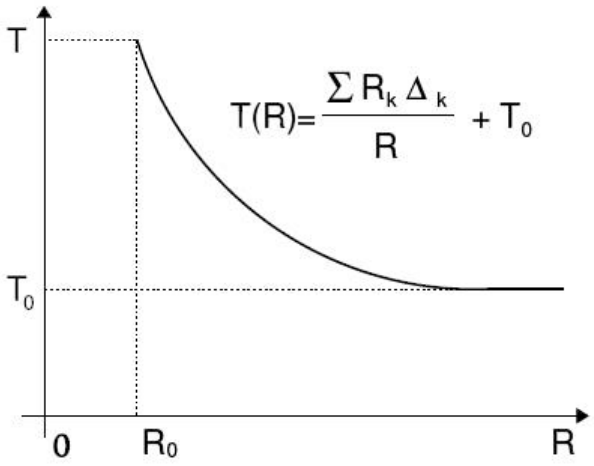


Figure 38: Upper bound on the channel change time T versus bandwidth R

In order to reduce the delay due to the waiting for the next RAP, it is possible to use a *proxy server approach* as show in Figure 39. Normally, the proxy server just forwards the multicast downstream to the clients. However, upon channel change requests, the server sends a supplementary unicast stream to the client consisting of the channel segment starting from a previous random access point up to the current position in the multicast stream. In this way, the receiver does not have to wait for a key frame, it will receive one immediately. Additionally, the supplementary unicast stream may fill the buffers faster than the multicast stream, since the bit rate of this stream may be higher than the bit rate of the multicast channel.

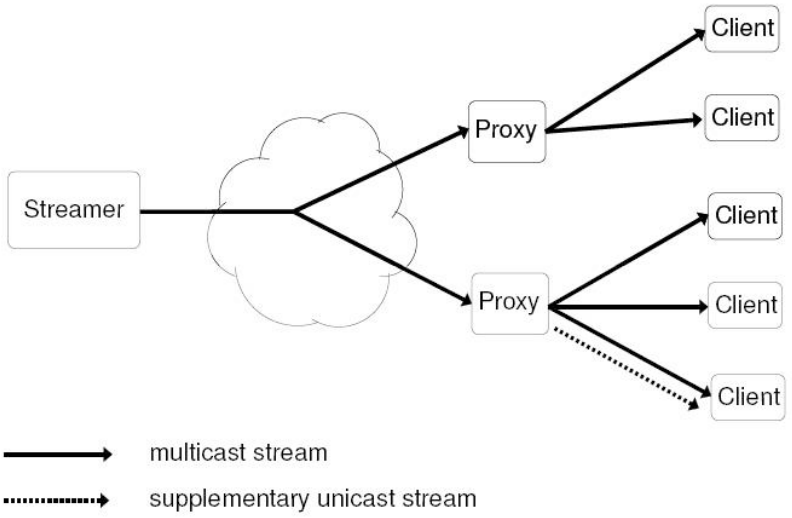


Figure 39: The proxy server approach

3.4 Channel Change performance

In order to provide a better comprehension of the issue and to determine the channel changing performance, several **performance metrics** are used. The following terms are defined and used to provide objective performance. Note that the terms defined here comprise generally accepted industry terms and protocol-specific terms that help characterize the performance.

IGMP Join Latency - The time between a request to join a multicast group and the receipt of the first byte of data for a multicast group.

IGMP Leave Latency - The time between a request to leave a multicast group and the receipt of the last byte of data for the multicast group

Channel Overlap - The duration of time when data is received for a new joined multicast group and a previously left multicast group. This time is usually zero units.

Channel Switch Delay (STB dependent) - An internal IGMP processing delay between a Leave and Join request. This value is ideally insignificant; however, it can be otherwise.

Channel Change/Zap Delay - The inter-channel change delay, which is the time

between a channel Leave request sent and the receipt of the first byte of data from the new multicast channel. It is the *IGMP Join Latency + Channel Switch Delay (STB dependent)*. This value is ideally very close to the IGMP Join Latency; however, the STB can introduce a significant delay. The following timing diagrams outline the delays for 2 channels. The following timing diagrams outline the delays for 2 channels.

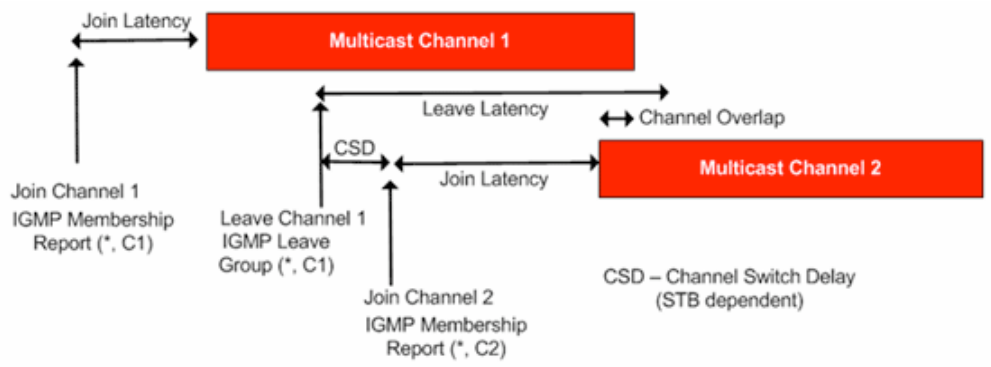


Figure 40: IGMP Join and Leave latency timing diagram

The specific metrics outlined above must be available on a per channel/per-subscriber basis. The various metrics make up the parts of a sum that help provide a check for every subscriber and the channels being watched. It is not sufficient, though, to use such metrics alone to characterize the performance of an IPTV network/device. To isolate adverse network conditions causing unacceptable channel change performance, network centric measurements must be available to provide a check of the network on a per-channel/per-subscriber.

- **Media Delivery Index (MDI)** – A scoring mechanism that combines packet delay variation (jitter) and media packet loss to determine the quality of the network to transport good quality video. It is measured in milliseconds.
- **MDI:DF (Delay Factor)** – Defined as cumulative IP jitter. It represents the time it would take to drain an output buffer and ensure good video playback.
- **MDI:MLR (Media Loss Rate)** – Defined as the packet loss rate due to dropped packets, bad/corrupted packets, or out-of-sequence packets.

The Media Delivery Index (MDI) is important because it characterizes the **performance** of the network and its **ability** to handle good video streams. The index provides two measures

separately so that each IPTV device can be tested with various channel change patterns to help isolate device-specific issues. By identifying problems on an IPTV network device, it becomes easy to troubleshoot and optimize the configuration to provide optimal performance end-to-end. This approach is useful to characterize the end-to-end channel change performance of an IPTV deployment.

3.4.1 Testing IPTV service performance

Hereafter there is a presentation of some test scenario made by IXIA in order to test the performance of IPTVservice.

IXIA is a leading provider of performance test systems for IP-based infrastructure and services. Its highly scalable solutions generate, capture, characterize, and emulate network and application traffic, establishing definitive performance and conformance metrics of network devices or systems under test. Ixia's test systems are used by Network and Telephony Equipment Manufacturers, Semiconductor Manufacturers, Service Providers, Governments, and Enterprises to validate the functionality and reliability of complex IP networks, devices, and applications. Ixia's Triple Play test systems address the growing need to test voice, video, and data services and network capability under **real-world conditions**. Ixia's vision is to be the world's pre-eminent provider of solutions to enable testing of next generation IP Triple Play networks. Ixia's test systems utilize a wide range of industry-standard interfaces and are distinguished by their performance, accuracy, reliability, and adaptability to the industry's constant evolution.

Test Setup

The topology presented here is representative of a typical Triple Play deployment network A video server port is used to stream 100s of standard and high definition video streams. The IGMPv3 protocol is used for the multicast communication and the PIM-DM is used for routing. This is the topology used for both the tests described below.

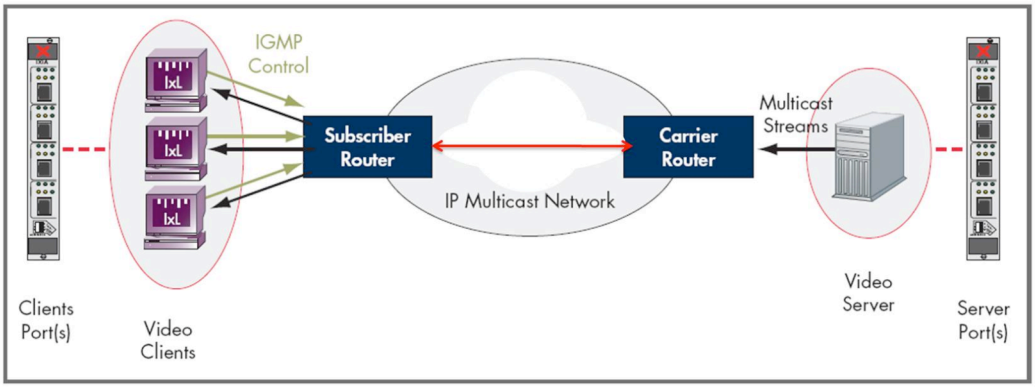


Figure 41: Typical Triple Play deployment network

Testing Optimal Channel Change Performance

This test setup focuses on determining the optimal channel change performance with IPTV traffic. Since IPTV subscriber has a different usage pattern, it is necessary to determine the end user's experience based on realistic channel change patterns. It is also important to determine how the subscriber's channel change performance changes as the complexity of the subscriber's behavior (channel change profile) increases and as the number of subscribers increase. The variability of the CC patterns will have a direct impact on the performance of devices supporting such traffic. The importance of determining the optimal performance of an IPTV system is to ensure that it is within operating limits of providing excellent channel change performance experienced by subscribers. For example, is there channel overlap when users are rapidly "surfing" channels? How does poor jitter experienced by one video stream affect the others on the same link? How is the transport of video streams affected by a growing number of subscribers? These are just some of the questions that can be answered by emulating realistic user channel change behavior to determine various acceptable levels of performance. This test emulates typical load conditions of 1000s of subscribers with multiple channel changing profiles and 100s of video streams. Several iterations of the test will help ensure that the device/network has sufficient raw bandwidth to support the load, reveal at localized congestions, and assist in fine-tuning devices for maximum performance.

The maximum number of users supported by the network does not necessarily translate to optimal channel change performance. There is a trade-off between the best channel change

performance with realistic subscriber loads and the maximum performance possible by the device/network. Both metrics have merit and both must be determined.

Some channel changing profiles can include:

• A set of subscribers surfing through channels 1-50 for 10-20 seconds each.

• A set of subscribers surfing through channels 51-100 for 1-30 seconds each.

• A set of subscribers emulating a typical habit of watching channel A for 30 minutes with frequent channel surfing within that time for commercial breaks.

The channel-changing profiles must be iteratively set up to test the multicast network and monitor an average subscriber's channel change performance.

A brief result of three RUNs is reported below:

Stat Name	Stream Bit Rate (bps)	MDI-DF (us)	MDI-MLR	MPEG2 TS Loss	Jitter (ns)	Inter Pkt Arrival Time (ns)	Packet Latency (ns)	Join Latency (ms)	Leave Latency (ms)
RUN 1/ User 1 Channel 1	3750007	2835	0	0	280	2805680	38160	69	8479
RUN 2/User 1 Channel 1	3750007	2835	0	0	1780	2807180	38320	108	8460
RUN 3/User Channel 1	3750007	2835	0	0	300	2216420	38540	100	5045

Table 5: channel change performance for multiple runs (test 1)

The first run simulated 300 users watching 100 channels for 10-20 seconds each. The second run increased the number of users to 1200. The jitter increased considerably with the increase in subscriber count. The third run reduced the subscriber count to 800 and modified the channel change behavior for different sets of subscribers.

The optimal channel change performance, as said above, is a trade-off between the maximum performance possible by the network without packet loss and the acceptable channel change latencies under realistic load conditions. Optimal channel changing performance was measured at an equilibrium point that was below the maximum device performance but still simulated realistic subscriber load conditions.

Testing Single Subscriber Experience

This test is composed of a series of runs with 1000s of subscribers with various channel change patterns. The test will assess the experience of a single subscriber who is watching a few channels while several other subscribers place different load requirements on the multicast network. The real-time nature of monitoring a single subscriber's Join/Leave latencies is essential.

Some channel changing profiles can include:

• One subscriber (pilot) watching a few channels for long durations.

• A set of subscribers emulating many channel change patterns to stress the multicast network differently. The channel changing profiles must be iteratively set up to test the multicast network to monitor a single subscriber's channel change performance for the iterations.

An instantaneous view of the real-time information of the single subscriber is presented below:

Stat Name	Stream Bit Rate (bps)	MDI-DF (us)	MDI-MLR	MPEG2 TS Loss	Jitter (ns)	Inter Pkt Arrival Time (ns)	Packet Latency (ns)	Join Latency (ms)	Leave Latency (ms)
RUN 1/ User 1	3750004	2229	0	0	198	2216220	38360	100	4503
RUN 2/User 1	3750004	2233	0	0	422	2217140	38020	170	6901

Table 6: Channel change performance for multiple runs (test 2)

The first run included 500 users with varying channel changing patterns. The second run increased the subscriber count to 1000. From the table above, it can be seen that increasing the subscribers and hence the load on the multicast network, increased the subscriber's perceived Join/Leave latencies by 70% and 65% respectively. The channel change/zap delay observed for the single subscriber was similarly affected. The behavior of a single subscriber being affected by other subscribers is primarily a result of the network having no QoS setup to police any traffic. Generally, the QoS on the network must be set up so that the single subscriber sessions are not affected by other subscribers on the same link. By monitoring a single subscriber, it is possible to troubleshoot devices that may not be observing the QoS settings or traffic policy properly. Such monitoring, however, may not be possible if the only metric observed is the optimal channel change for all aggregate

subscribers.

4 Description of some solutions

4.1 Unicast-Based Rapid Synchronization with RTP Multicast Sessions

This issue is explained in the Internet-Draft edited by Alcatel-Lucent, Cisco and Microsoft engineering. Rapid Multicast Synchronization is supported as part of the **Microsoft Mediaroom Internet Protocol Television** (IPTV) and multimedia software platform. This system is in wide commercial deployment.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

4.1.1 Introduction

Most multicast flows carry a stream of **inter-related data**. Certain information must first be acquired by the receivers to start processing any data sent in the multicast session. This document refers to this information as **Reference Information**. The Reference Information is conventionally sent periodically in the multicast session and usually consists of items such as a description of the schema for the rest of the data, references to which data to process for the receivers, encryption information including keys, as well as any other information required to process the data in the multicast flow. Real-time multicast applications require the receivers to buffer data. The receiver may have to buffer data to smooth out the network jitter, to allow loss-repair methods such as Forward Error Correction and retransmission to recover the missing packets, and to satisfy the data processing requirements of the application layer. When a receiver joins a multicast session, it has no control over what point in the flow is currently being transmitted. Sometimes the receiver

may join the session right before the Reference Information is sent in the session. In this case, the required waiting time is usually minimal. Similarly, the receiver may also join the session right after the Reference Information has been transmitted. In this case the receiver has to wait for the Reference Information to appear again in the stream before it can start processing any multicast data. In some other cases, the Reference Information is not contiguous in the flow but dispersed over a large period, which forces the receiver to wait for all of the Reference Information to arrive before starting to process the rest of the data. The net effect of waiting for the Reference Information and waiting for various buffers to fill up is that the receivers may experience significantly large delays in data processing. In this document, we refer to the difference between the time a receiver joins the multicast session and the time the receiver acquires all the necessary Reference Information as the **Synchronization Delay**. The synchronization delay may not be the same for different receivers; it usually varies depending on the join time, length of the Reference Information repetition interval, size of the Reference Information as well as the application and transport properties. The varying nature of the synchronization delay adversely affects the receivers that frequently switch among multicast sessions. In this specification, we address this problem for RTP-based multicast applications and describe a method that uses the fundamental tools offered by the existing RTP and RTCP protocols [RFC3550]. In this method, either the multicast source (or the distribution source in a single-source multicast (SSM) session) retains Reference Information for a period after transmission, or an intermediary network element joins the multicast session and continuously caches the Reference Information as it is sent in the session and acts as a feedback target for the session. When a receiver wishes to join the same multicast session, instead of simply issuing an Internet Group Management Protocol (IGMP) Join message, it sends a request to the **feedback target address** for the session asking for the Reference Information. The feedback target starts a unicast retransmission RTP session and sends the Reference Information to the receiver over that session. If there is spare bandwidth, the feedback target may also burst the Reference Information at a faster than natural rate. As soon as the receiver acquires the Reference Information, it can join the multicast group and start processing the multicast data. This method potentially reduces the synchronization delay. We refer to this method as Unicast-based Rapid Synchronization with

RTP Multicast Sessions. A simplified network diagram showing the rapid synchronization method through an intermediary network element is depicted in Figure 42.

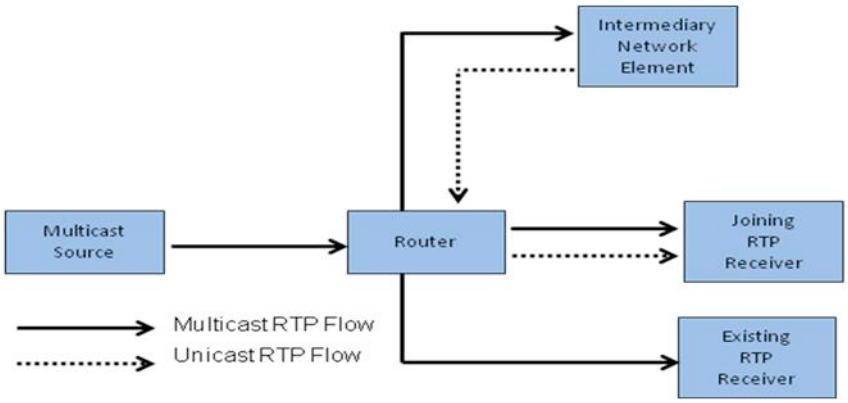


Figure 42: Rapid synchronization through an intermediary network element

A primary design goal in this solution is to use the existing tools in the RTP protocol family. This improves the versatility of the existing implementations, and promotes faster deployment and better interoperability. To this effect, we use the unicast retransmission support of RTP [RFC4588] and the capabilities of RTCP to handle the signaling needed to accomplish the synchronization. The packet(s) carrying the Reference Information are sent by the feedback target in the auxiliary unicast session for rapid synchronization. These are constructed as retransmission packets that would have been sent in a uncast RTP session to recover the missing packets at a receiver that has never received any packet. In fact, a single RTP session may be used for both rapid synchronization and retransmission-based loss repair. Furthermore, the session can be used to simultaneously provide unicast burst traffic for the rapid synchronization and repair packets requested by the receiver when it detects lost burst packets or lost RTP packets from the primary multicast stream (in the case it is receiving both streams at the same time). The conventional RTCP feedback message that requests the retransmission of the missing packets indicates their sequence numbers. However, upon joining a new session the receiver has never received a packet and thus, does not know the sequence numbers. Instead, the receiver sends a newly defined RTCP feedback message to request the Reference Information needed to rapidly synchronize with the primary multicast session. It is also worth noting that in order to issue the initial RTCP message to the feedback target, the SSRC of the session to be joined must be known prior to any packet

reception, and hence, needs to be signaled out-of-band (or in-band). In a Session Description Protocol (SDP) description, the SSRC must be signaled through the 'ssrc' attribute.

4.1.2 Elements of Delay in Multicast Streams

In an any-source (ASM) or a single-source (SSM) multicast delivery system, there are three major elements that contribute to the overall synchronization delay when a receiver switches from one multicast session to another one. These are:

- Multicast switching delay
- Reference Information latency
- Buffering delays

Multicast switching delay is the delay that is experienced to leave the current multicast session (if any) and join the new multicast session. In typical systems, the multicast join and leave operations are handled by a group management protocol. For example, the receivers and routers participating in a multicast session may use the Internet Group Management Protocol (IGMP). When a receiver wants to join a multicast session, it sends an IGMP Join message to its upstream router and the routing infrastructure sets up the multicast forwarding state to deliver the packets of the multicast session to the new receiver. Depending on the proximity of the upstream router, the current state of the multicast tree, the load on the system and the protocol implementation, the join times vary. Current systems provide join latencies usually less than **200 milliseconds** (ms). If the receiver had been participating in another multicast session before joining the new session, it needs to send an IGMP Leave message to its upstream router to leave the session. In IGMP version 3, the leave times are usually smaller than the join times, however, it is possible that the Leave and Join messages may get lost, in which case the multicast switching delay inevitably increases.

Reference Information latency is the time it takes the receiver to acquire the Reference Information. It is highly dependent on the proximity of the actual time the receiver joined the session to the next time the Reference Information will be sent to the receivers in

the session, whether the Reference Information is sent contiguously or not, and the size of the Reference Information. For some multicast flows, there is a little or no interdependency in the data, in which case the Reference Information latency will be nil or negligible. For other multicast flows, there is a high degree of interdependency. One example of interest is the multicast flows that carry compressed audio/video. For these flows, the Reference Information latency may become quite large and be a major contributor to the overall delay.

The buffering component of the overall synchronization delay is driven by the way the application layer processes the payload. In many multicast applications, an unreliable transport protocol such as UDP is often used to transmit the data packets, and the reliability, if needed, is usually addressed through other means such as Forward Error Correction and retransmission. These loss-repair methods require buffering at the receiver side to function properly. In many applications, it is also often necessary to de-jitter the incoming data packets before feeding them to the application. The de- jittering process also increases the buffering delays. Besides these network-related buffering delays, there are also specific buffering needs that are required by the individual applications. For example, MPEG decoders require a significant amount of content to be available in the decoder buffers prior to starting to decode the content.

4.1.3 Protocol Design Considerations and their effect on resource management for Rapid Synchronization

Rapid synchronization is an optimization of a system that must continue to work correctly whether or not the optimization is effective, or even fails due to lost control messages, congestion, or other problems. This is fundamental to the overall design requirements surrounding the protocol definition and to the resource management schemes to be employed together with the protocol (e.g., QoS machinery, server load management, etc). In particular, the system needs operate within a number of constraints.

-First, a rapid synchronization operation must fail gracefully. The user experience must, except perhaps in pathological circumstances, be not significantly worse for trying and failing to complete rapid synchronization compared to simply joining the multicast session.

-Second, providing the rapid synchronization optimizations must not cause collateral damage to either the multicast session being joined, or other multicast sessions sharing resources with the rapid synchronization operation. In particular, the rapid synchronization operation must avoid self-interference with the multicast session that may be simultaneously being received by other hosts. In addition, it must also avoid interference with other multicast sessions sharing the same network resources. These properties are possible, but difficult to achieve.

One challenge is the existence of multiple bandwidth bottlenecks between the receiver and the server(s) in the network providing the rapid synchronization service. In commercial IPTV deployments, for example, bottlenecks are often present in the aggregation network connecting the IPTV servers to the network edge, the access links (e.g., DSL, DOCSIS) and in the home network of the subscribers. Some of these links may serve only a single subscriber, limiting congestion impact to the traffic of only that subscriber, but others can be shared links carrying multicast sessions of many subscribers.

A second challenge is that for some uses (e.g., high-bit rate video) the burst bandwidth is high while the flow duration of the burst is short. This is because the purpose of the burst is to allow the receiver to join the multicast quickly and thereby limit the overall resources consumed by the burst. Such high-bit rate, short-duration flows are not amenable to conventional admission control techniques. For example, per-flow signaled admission control techniques such as RSVP have too much latency and control channel overhead to be a good fit for rapid multicast synchronization. Similarly, using a TCP (or TCP-like) approach with a 3-way handshake and slow-start to avoid inducing congestion would defeat the purpose of attempting rapid synchronization in the first place by introducing many RTTs of delay.

These observations lead to certain unavoidable requirements and goals for a rapid multicast synchronization protocol. These are:

The protocol must be designed to allow a deterministic upper bound on the extra bandwidth used (compared to just joining the multicast group). A reasonable size bound is e*B, where B is the "nominal" bandwidth of the multicast flow, and e is an "excess-bandwidth" coefficient The total duration of the burst must have a reasonable bound; long

bursts devolve to the bandwidth profile of multi-unicast for the whole system.

The scheme should minimize (or better eliminate) the overlap of the burst and the multicast flow. This minimizes the window during which congestion could be induced on a bottleneck link compared to just carrying the multicast or unicast packets alone.

The scheme must minimize (or better eliminate) any gap between the unicast burst and multicast flow which has to be repaired later, or in the absence of repair, will result in loss being experienced by the application.

In addition to the above, there are some other protocol design issues to be considered. First, there is at least one RTT of "slop" in the control loop. In starting a rapid multicast synchronization burst, this manifests as the time between the client requesting the burst and the burst description (and possibly the first burst packets) arriving at the receiver. For managing and terminating the burst, there are two possible approaches for the control loop: The receiver can adapt to the burst as received, converge based on observation and explicitly terminate the burst with a second control loop exchange (which takes a minimum of one RTT, just as starting the burst does). Alternatively, the server generating the burst can pre-compute the burst parameters based on the information in the initial request and tell the receiver the burst duration. The protocol described in the next section allows either method of controlling the rapid multicast synchronization burst.

4.1.4 RMS Overview

The draft specifies an extension to the RTP Control Protocol (RTCP) to use unicast feedback in an SSM session. It defines an architecture that introduces the concept of **Distribution Source**, which - in an SSM context - distributes RTP data and redistributes RTCP information to all receivers. This RTCP information is retrieved from the Feedback Target, to which RTCP unicast feedback traffic is sent. The Feedback Target may be implemented in one or more entities different from the Distribution Source, and different RTP receivers may use different Feedback Targets. This document builds further on these concepts to reduce the synchronization time when an RTP receiver wants to join a multicast session at a random point in time by introducing the concept of the **Burst Source** and new RTCP

feedback messages. The Burst Source has a cache where the most recent RTP packets from the SSM distribution are continuously stored. When an RTP receiver wants to receive an SSM RTP stream, prior to joining the SSM session, it will first request an RTP burst from the Burst Source. In this burst, the packets are formatted as RTP retransmission packets and the data carried in these packets allow the RTP receiver to synchronize quickly with the SSM session. Using an accelerated rate (as compared to the rate of the primary multicast stream) for the RTP burst implies that at a certain point in time, the payload transmitted in the RTP burst is going to be the same as the payload multicast in the SSM session, i.e., the unicast burst will catch up with the primary multicast stream. At this point, the RTP receiver no longer needs to receive the unicast RTP burst and can join the SSM session. This method is referred to as the **Rapid Multicast Synchronization** (RMS) method. This document proposes extensions to the RFC4585 for an RTP receiver to request an RTP burst as well as for additional control messaging that can be leveraged during the synchronization process.

4.1.5 Messages Flow

The entities involved in the Rapid Multicast Synchronization are:

Multicast Source

Feedback Target (FT)

Burst Source

Retransmission Source

RTP Receiver (RR)

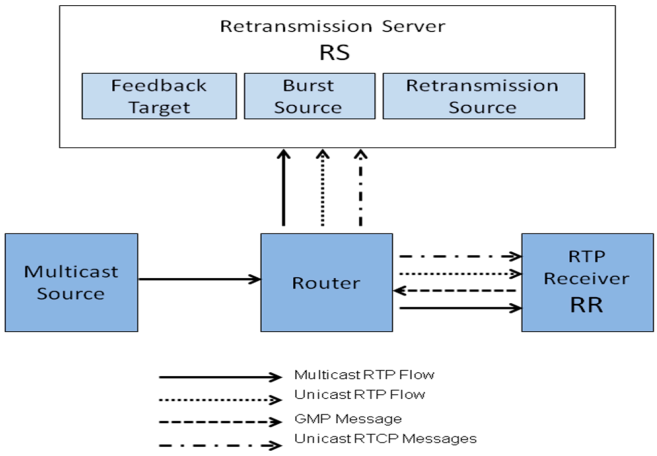
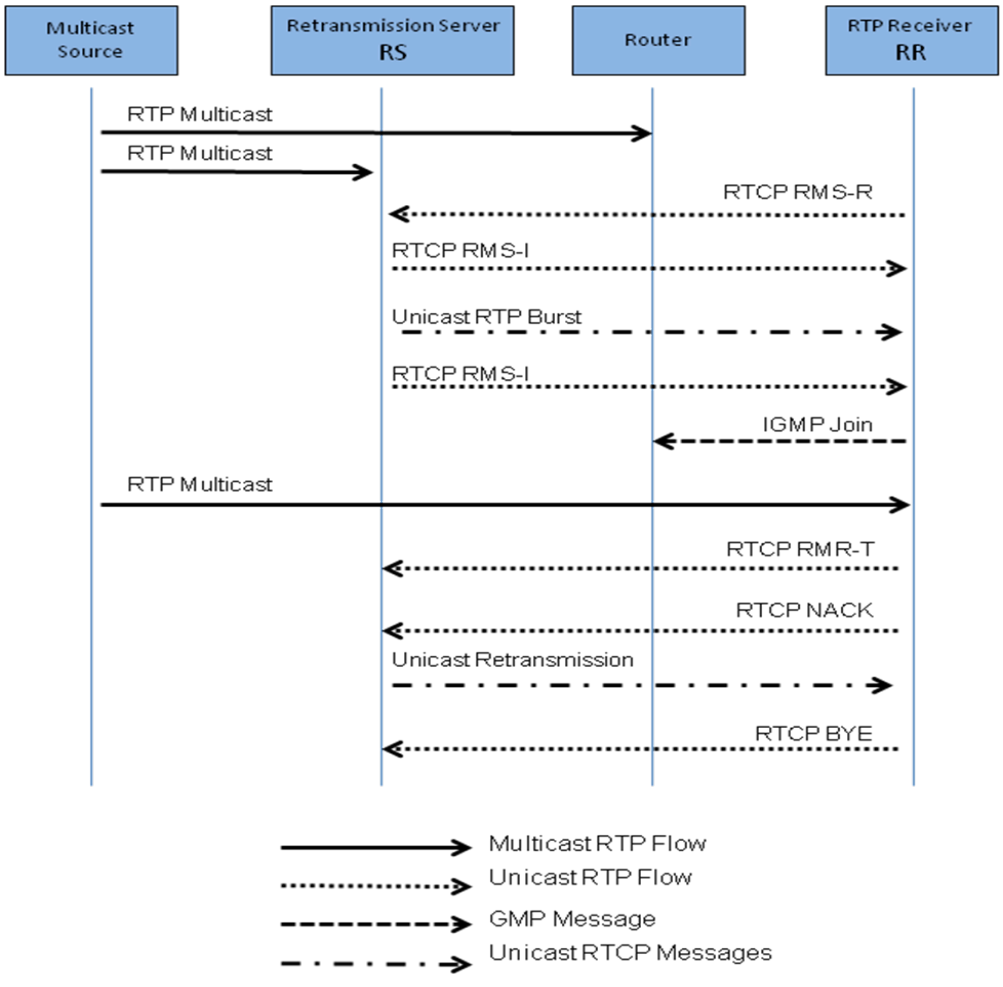


Figure 43: Flow diagram for unicast-based rapid synchronization

A Retransmission Source can equally act as a Burst Source. The Retransmission Source can also incorporate the Feedback Target (permits the feedback target to be a retransmission server, since it is a logical function to which RRs send their unicast feedback), and we will use the term **Retransmission Server (RS)** in the remainder of the document to refer to a single physical entity comprising these three entities. Note that the same method (with the identical message flows) would also apply in a scenario where rapid synchronization is performed by a feedback target co-located with the media source. As the **RTP burst packets** are formatted as RTP retransmission packets, the unicast RTP burst and RTP retransmissions may be provided in one and the same RTP (retransmission) session. The RTP burst is triggered by the RTCP feedback message defined in this document, whereas an RTP retransmission is triggered by an RTCP NACK message. Pending on RMS practices, there may be a gap between the end of the burst stream and the reception of the primary multicast stream because of the imperfections in the switch-over. RR can make use of the RTCP NACK message to request a retransmission for the missing packets in the gap. Note that FT, Burst Source and Retransmission Source are logical entities. For efficiency and simplicity, they may be implemented by a single physical **Retransmission Server (RS)**. The figure below depicts an example of messaging flow for rapid synchronization. The RTCP feedback messages are explained below.

Figure 44: Message flows for unicast-based rapid synchronization



This document defines the expected behaviors of RS and RR. It is instructive to have independently operating implementations on RS and RR that request the burst, describe the burst, start the burst, join the multicast

stream and stop the burst. These implementations send messages to each other, but there must be provisions for the cases where the control messages get lost or are not being delivered to their destinations. The following steps describe rapid synchronization in detail:

Request: RR sends a rapid synchronization request for the new multicast RTP session to the feedback target address of that session. The request contains the SSRC of RR and the SSRC of the media source. This RTCP feedback message is defined as Rapid Multicast Synchronization Request (**RMS-R**) message and may contains parameters, which may constrain the burst, such as the bandwidth limit. Other parameters may be related to the amount of buffering capacity available at RR, which may be used by RS to prepare a rapid synchronization burst that conforms to RR's requirements. Before joining the primary multicast session, a new joining RR learns the addresses associated with the new multicast session (addresses for the multicast source, group and retransmission server) by out-of-band means. Also note that since no RTP packets have been received yet for this session, the SSRC must be obtained out-of-band or in-band.

Response: RS receives the RMS-R message and decides whether to accept it or not. RS must send an (at least one) RMS-Information (**RMS-I**) message to RR. The first RMS-I message may precede the burst or it may be sent during the burst. Additional RMS-I messages may be sent during the burst. The join-time information (for the new multicast session) must be populated in at least one of the RMS-I messages. Note that RS learns the IP address and port information for RR from the RMS-R message it received.

If **RS cannot provide a rapid synchronization service**, RS rejects the request and informs RR immediately via an RMS-I message. If RR receives a message indicating that its rapid synchronization request has been denied, it abandons the rapid synchronization attempt and may immediately join the multicast session by sending an IGMP Join message to its upstream multicast router for the new multicast session.

If **RS accepts the request**, it sends an RMS-I message to RR (before commencing the burst or during the burst) that comprises fields that can be used to describe the burst that will be sent by RS (e.g., the bit rate and the size of the burst). There may also be optional payload-specific information that RS chooses to send to RR. Such an example for

transmitting the payload-specific information for MPEG2 Transport Streams. The burst duration may be calculated by RS, and its value may be updated by messages received from RR.

Updated Responses: RS may send more than one RMS-I messages, e.g., to update the burst bit rate information when the bit rate is adapted and/or to signal RR in real time to join the SSM session. For redundancy purposes, an RMS-I message may also be sent multiple times. RR may depend on RS to learn the join-time for the SSM session. The join-time can be conveyed by the first RMS-I message, or it can be sent/revised in later RMS-I messages. If RS is not capable of determining the join-time in the first RMS-I message, it will have to send another RMS-I message later.

SSM Join: In principal, RR can join the primary multicast session any time during or after the end of the RTP burst via an IGMP Join message. However, there may be missing packets if RR joins the SSM session too early or too late. For example, if RR starts receiving the primary multicast stream while it is still receiving the RTP burst at a high excess bit rate, this may result in an increased risk of packet loss. Or, if RR joins the SSM session some time after the RTP burst is finished, there may be a gap between burst and multicast data (a number of RTP packets may be missing). In both cases, RR may issue retransmissions requests to fill the gap. Yet, there are cases where the remaining available bandwidth may limit the number of retransmissions that can be provided within a certain time period, causing the retransmission data to arrive too late at RR (from application layer point of view). To cope with such cases, the RMS-I message allows RS to signal explicitly when RR should join the SSM session. Alternatively, RS may pre- compute the burst duration and the time RR should join the SSM session. This information may be conveyed in the RMS-I message and can be updated in subsequent RMS-I messages. RR may use the information from the most recent RMS-I message, or it may use a locally calculated join-time.

Multicast Receive: After joining the SSM session, RR starts receiving the multicast RTP flow.

Terminate: RS may know when it needs to stop the unicast burst based on the burst parameters, or RR may explicitly let RS know the sequence number of the first RTP packet it

received from the multicast session, or RR may request RS to terminate the burst immediately. RR shall use the RMS-Termination (RMS-T) message when it wishes to provide information to RS regarding the cessation of the burst. RR can choose to send the RMS-T message before or after it starts receiving the multicast data. In the latter case, RR shall include the sequence number(SN) of the first RTP packet received in the SSM session in the RMS-T message, and RS should terminate the burst after it sends the RTP burst packet whose OSN field in the RTP retransmission payload header matches this number minus one. If RR wants to stop the burst prior to receiving the multicast data, it sends an empty RMS-T message (i.e., without an RTP sequence number). Note that regardless of whether RS knows when to stop the burst or not, RR must send at least one RMS-T message. Against the possibility of a message loss, RR may repeat the RMS-T messages multiple times as long as it follows the RTCP timer rules defined in the RFC4585.

Terminate with RTCP BYE: When RR is no longer interested in receiving the primary multicast stream and the associated burst, RR shall issue an RTCP BYE message to the Feedback Target to terminate the burst and RTP retransmission session. Upon receiving an RTCP BYE message, RS must terminate the rapid synchronization operation, and cease transmitting any further packets of the associated unicast burst. The RFC3550 mandates the RTCP BYE message always to be sent with a sender or receiver report in a compound RTCP packet (If no data has been received, an empty receiver report MUST be included). With the information contained in the receiver report, RS can also figure out how many duplicate RTP packets have been delivered to RR (Note that this will be an upper-bound estimate as one or more packets might have been lost during the burst transmission). Note that if RR decides to switch to a new multicast session after it already joined a multicast session following a rapid synchronization request, RR must also send an RTCP BYE message for the session associated with the current multicast source stream. For the purpose of gathering detailed information about RR's rapid synchronization experience, it could be defined an RTCP Extended Report (XR) Block. This report is designed to be payload-independent, thus, it can be used by any multicast application that supports rapid synchronization. Support for this XR report is, however, optional.

4.1.6 Message format

This section defines the formats of the RTCP transport-layer feedback messages that are exchanged between the Retransmission Server (RS) and RTP Receiver (RR) during rapid multicast synchronization (RMS). These messages are **payload-independent** and should be used by all RTP-based multicast applications that support rapid synchronization regardless of the payload they carry. Specific payload formats are not defined in this document, but a framework is presented that allows payload-specific information to be included in the exchange. The common packet format for the RTCP feedback messages is defined in Section 6.1 of the RFC4585 and the Figure 45 describes it.

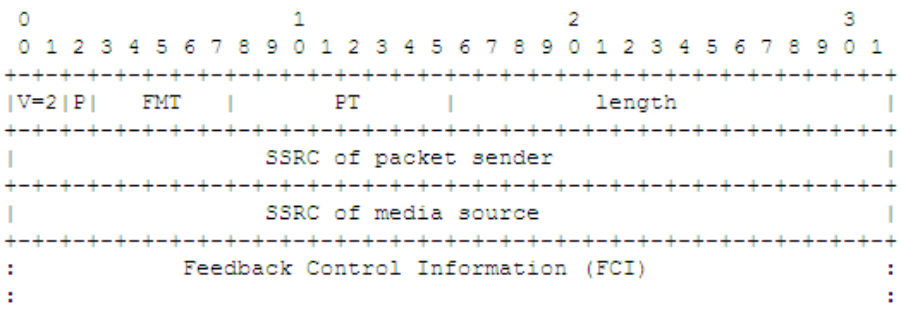


Figure 45: Common Packet Format for Feedback Messages

Each feedback message has a fixed-length field for version, padding, feedback message type (FMT), payload type (PT), length, SSRC of packet sender, SSRC of media source as well as a variable-length field for feedback control information (FCI). In the transport-layer feedback messages, the PT field is set to **RTPFB (205)**. Note that rather than having special cases for "value unknown or unspecified" in each of the fields carried in the messages, we are considering to format the fields in these structures as **Type/Length/ Value (TLV)** elements. If the originator of the message does not have a value for a field, the field will not be present. The advantage of this design is that there will not be any ambiguity in the value of the field. The disadvantage is that the message is variable length, and slightly more complex to generate/parse.

Extensions: Optional extended parameters may be encoded using TLV elements as

described below:

Type: A single-octet identifier that defines the type of the parameter represented in this TLV element. Length: A two-octet field that indicates the length of the Value field.

Value: Variable-size set of octets that contains the specific value for the parameter. If a TLV element does not fall on a 32-bit boundary, the last word must be padded to the boundary using further bits set to 0.

Note that a design that will allow vendor-specific extensions to be used in these messages is also desirable. For interoperability purposes, the design must avoid collisions. Some approaches are currently being examined.

RMS Request

The RMS Request message is identified by **PT=RTPFB** and **FMT=5**.

The FCI field must contain only one RMS Request. The RMS Request is used by RR to request rapid synchronization for a new multicast RTP session. The FCI field has the structure depicted in the figure above.

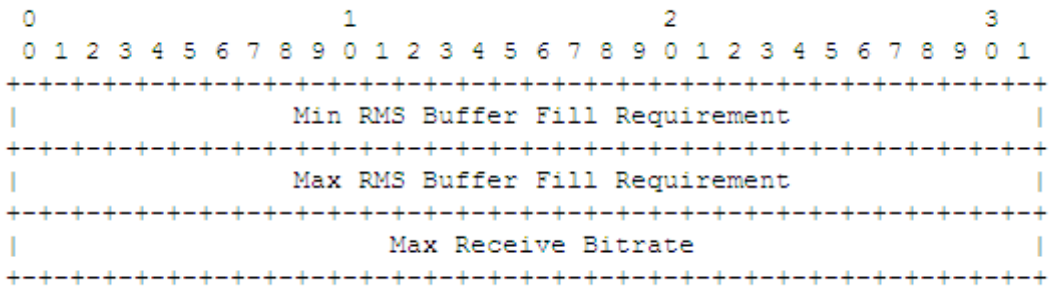


Figure 46: FCI field syntax for the RMS Request message

Min RMS Buffer Fill Requirement (32 bits): The minimum amount of data (in ms) required by RR after the burst completes. If specified, the amount of backfill that will be provided by the unicast burst should not be smaller than this value since it will not be able to build up the desired level of buffer at RR and may cause buffer under runs.

Max RMS Buffer Fill Requirement (32 bits): The maximum amount of data (in ms) that can be received by RR after the burst completes. If specified, the amount of backfill that

will be provided by the unicast burst should not be larger than this value since it may cause buffer overflows at RR. **Max Receive Bit rate** (32 bits): The maximum bit rate (in bits per second) that RR can receive. If specified, the unicast burst bit rate should not be larger than this value since it may cause congestion and packet loss. The length of the feedback message must be set to 2+n, where n is the number of fields contained in the message. The semantics of this feedback message is independent of the payload type.

RMS Information

The RMS Information message is identified by **PT=RTPFB** and **FMT=6**.

The FCI field must contain only one RMS Information. The RMS Information is used to describe the unicast burst that will be sent for rapid synchronization. It also includes other useful information for RR. The FCI field has the structure depicted in the figure above.

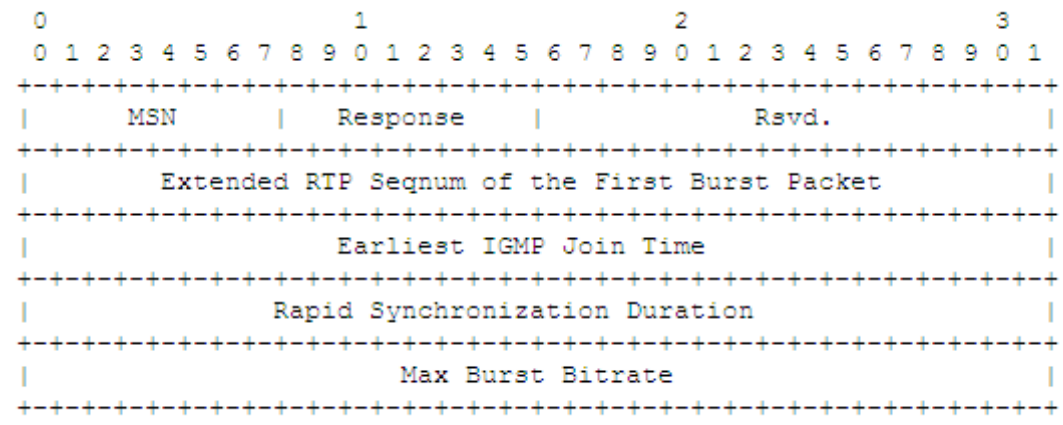


Figure 47: FCI field syntax for the RMS Information message

Message Sequence Number (8 bits): During rapid synchronization, the RMS-I message(s) may be sent more than once. The first RMS-I message SHALL have an MSN value of 0. This value SHALL NOT be changed if the same RMS-I message is sent to the same RR multiple times for redundancy purposes. If new information is conveyed in a new RMS-I message, the MSN value SHALL be incremented by one.

Response (8 bits): Three values are defined: A value of 0 indicates that rapid

synchronization request has been rejected. This may trigger RR to proceed with joining the primary multicast session. A value of 1 indicates that the rapid synchronization request has been accepted. A value of 2 means that RR should Immediately join the primary multicast session. Other values may be defined later.

Rsvd (16 bits): Reserved.

Extended RTP Seqnum of the First Burst Packet (32 bits): The extended RTP sequence number of the first packet that will be sent as part of the rapid synchronization in the burst. This allows RR to know if one or more packets have been dropped at the beginning of the burst. 32-bit extended RTP sequence number is constructed by putting the 16-bit RTP sequence number in the lower two bytes and octet 0's in the higher two bytes.

Earliest IGMP Join Time (32 bits): Time difference between the arrival of the RMS-I message and the earliest time instant when RR could join the new multicast session (in RTP clock ticks). A value of all 1's means that it is not specified.

Rapid Synchronization Duration (32 bits): Time difference between the timestamps of the first and last RTP packets in the unicast burst (in RTP clock ticks). A value of all 1's means that it is not specified.

Max Burst Bit rate (32 bits): The max bandwidth used by RS for the unicast burst, expressed in bits per second. A value of 0 means that it is not specified. The length of the feedback message MUST be set to 2+n, where n is the number of fields contained in the message. The semantics of this feedback message is independent of the payload type. The RMS-I message MAY be sent multiple times at the start of, prior to, or during the RTP unicast burst. The subsequent RMS-I messages may signal changes in any of the fields.

RMS Termination

The RMS Termination message is identified by **PT=RTPFB** and **FMT=7**.

The FCI field must contain only one RMS Termination. The RMS Termination may be used to assist RS in determining when to stop the burst. If prior to sending the RMS-T message RR has already joined the multicast session and received at least one RTP packet from the multicast session, RR includes the sequence number of the first RTP packet in the

RMS-T message. With this information, RS can decide when to terminate the unicast burst. If RR issues the RMS-T message before it has joined and/or begun receiving RTP packets from the multicast session, RR does not specify any sequence number in the RMS-T message, which indicates RS to stop the burst immediately. However, note that RS may not receive this message or may alter the burst. The FCI field has the structure depicted in the figure above.

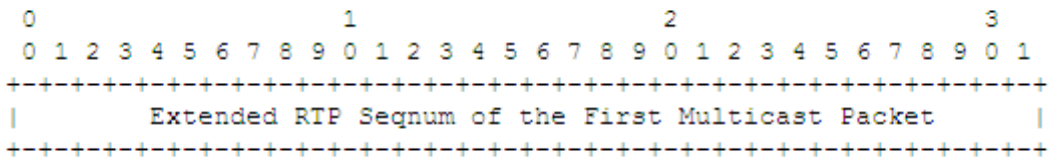


Figure 48: FCI field syntax for the RMS Termination message

Extended RTP Seqnum of the First Multicast Packet (32 bits): The extended RTP sequence number of the first packet received from the multicast session. Note that we need to have TLV syntax for this field, add a "valid flag" or come up with a reasonable value for "unknown". The length of the feedback message must be set to 2+n, where n is the number of fields contained in the message. The semantics of this feedback message is independent of the payload type.

5 Fast Channel Change solution

The solution for the **Fast Channel Change** is the idea of unicasting to the set-top box during the zapping-period, merging seamlessly with the multicast stream after a little while. In order to describe this solution is possible to consider three different perspectives on the scenario of this implementation:

1. The IPTV Set-top Box Perspective
2. The Network Perspective
3. The GOP Server Perspective

Below you see more detailed analysis of these three perspectives.

5.1 IPTV Set-top Box Perspective

The IPTV set-top box scenario can be best explained in the following user scenarios steps:

- **Step 1**: User presses button on remote.
- **Step 2**: STB leaves previous channel and resets decoder. In a properly configured network this is a very fast operation taking not more than 10 ms. this should not be a blocking operation for the STB software, i.e. no waiting involved.
- **Step 3**: STB opens UDP socket.
- **Step 4**: STB software notifies GOP Server. Message includes multicast group and port.
- **Step 5**: STB joins multicast group (sends IGMPv2 membership report).
- **Step 6**: for duration of a maximum of 500ms, traffic from both the GOP server and Multicast group arrives at the STB. This traffic is destined for the same UDP port. Traffic from the GOP server is unicast. Other traffic is multicast. The STB should be able to receive both types of traffic on a single socket. The STB should be able to deal with duplicate and out-of-order traffic. To do this in a reliable manner, RTP transport is preferred.

Notes to the above scenarios:

Notification as explained in step 3 is done through a UDP message and it is not a blocking procedure for the STB software. Even if UDP is not possible, a persistent TCP session would do the job as well.

The STB wouldn't bind to the multicast group, as it uses a local address on the UDP socket to receive the multicast traffic. It would bind to 0.0.0.0 (any local address). This means that the STB would be able to receive unicast and multicast traffic on the same socket. Even if the current implementation is not binding to an "any" address, this would be a minor change to the STB software.

An MPEG-TS (Transport Stream) layer or MPEG2 Video decoder must be able to

deal with repeated and out-of order MPEG-TS 188 byte packets, through time stamp or sequence elimination. The easiest way to do this is to use RTP rather than plain UDP for sending data.

5.2 Network Perspective

- **Step 2**: IGMP Leave. IGMP fast-leave kicks in. Multicast group left.
- **Step 4**: Single UDP message or TCP segment. Forwarded to GOP server.
- **Step 5**: IGMP membership report. Join PIM group. SPT switch-over. Deliver multicast packets.
- **Step 6**: Traffic from multicast group flows through, together with traffic from GOP server. To avoid interface overload when zapping, traffic from GOP server should be treated as less-than-best-effort (LBE)

5.3 GOP Server Perspective

The GOP Server keeps a copy of the current incomplete GOP in its memory. In a small setup, a GOP server would listen to a lot of multicast groups.

- **Step 4**: The GOP server receives a request.
- **Step 6**: The GOP server sends a shaped burst of UDP packets from the beginning of the GOP to current time plus a constant. To avoid interface queue overload when zapping, traffic from GOP server should be treated as a less-than-best-effort (LBE). Also traffic is shaped to a predefined bandwidth. For a 4Mbps stream, the size of the burst can be as big as 250KB.

Notes on the GOP Server:

The server needs to cache all the relevant data of the current incomplete GOP. This is easily accomplished by parsing the transport stream and reading the MPEG sequence and GOP headers when they arrive. No further processing or decoding of the stream needs to be done on the server.

The only interaction needed is with the middleware. The middleware needs to send as fast as possible a request to the GOP cache server, and the server will flood the current GOP to the unicast IP address of the STB. That solution is simple and will work with most of the IPTV head-ends. Our major issue is the time needed from the pressing of the button of the remote to the receiving of the GOP. The current incomplete GOP will be transmitted by unicast to the STB. There can be an issue with the buffer in the STB (imagine a situation where just the last frames have to be played, we send a full GOP, then comes the second GOP, the STB will have to receive both GOPs, or picture freezing will be introduced), but this can be addressed on the set top box, either by having a large enough buffer for two whole GOPs or by implementing a selective buffer purge. This is hard to address on the server side, because the network latency can vary. This solution will shorten the channel change time significally.

Assumptions on Timeline

- Button-press to IGMP leave = 200 ms (rather slow STB reaction)
- IGMP leave = 10 ms
- Decoder reset is a blocking operation = 50 ms
- Decoder start-up once, the jitter buffer is full = 50 ms
- IGMP join = 10 ms
- Network delay = 10 ms
- Stream bandwidth = 4Mbps
- Stream is shaped. i.e. I frame takes much more time than other frames
- GOP server burst shape = 15Mbps
- GOP server guard time = 20ms
- Join time offset = average case (250ms after beginning of GOP transmission)
- Jitter buffer threshold in STB = 2Mb (250KB)

Timeline list - Without the improvement

1. 0ms - User presses remote button
2. 200ms (+200) - IGMP leave, decoder reset start
3. 210ms (+10) - last Multicast packet from previous stream arrives
4. 250ms (+50 from nr.3) - decoder reset end, Open UDP socket, IGMP join
5. 260ms (+10) - first multicast packet from new stream arrives
6. 510ms (+250) - GOP begins
7. 1010ms (+500) - jitter buffer threshold reached, decoder start
8. 1060ms (+50) - First picture displayed

Timeline list - With improvement

1. 0ms - User presses remote button
2. 200ms (+200) - IGMP leave, decoder reset start
3. 210ms (+10) - last Multicast packet from previous stream arrives
4. 250ms (+50 from nr.2) - decoder reset end, Open UDP socket, signal GOP server, IGMP join
5. 260ms (+10) - first multicast packet from new stream arrives. Request arrives at GOP server, GOP burst begins at server
6. 270ms (+20 from nr.4) - GOP burst begins (20ms roundtrip to GOP server). 3.08Mb (exactly 1 GOP + 250ms random access time + 20ms guard time) at 15Mbps = 206ms. 1 GOP will arrive in 134ms
7. 404ms (+134) - Jitter buffer threshold reached, decoder start
8. 454ms (+50 from nr.7) - First picture displayed
9. 476ms (+206 from nr.6) - GOP burst ends, 20ms overlap with multicast

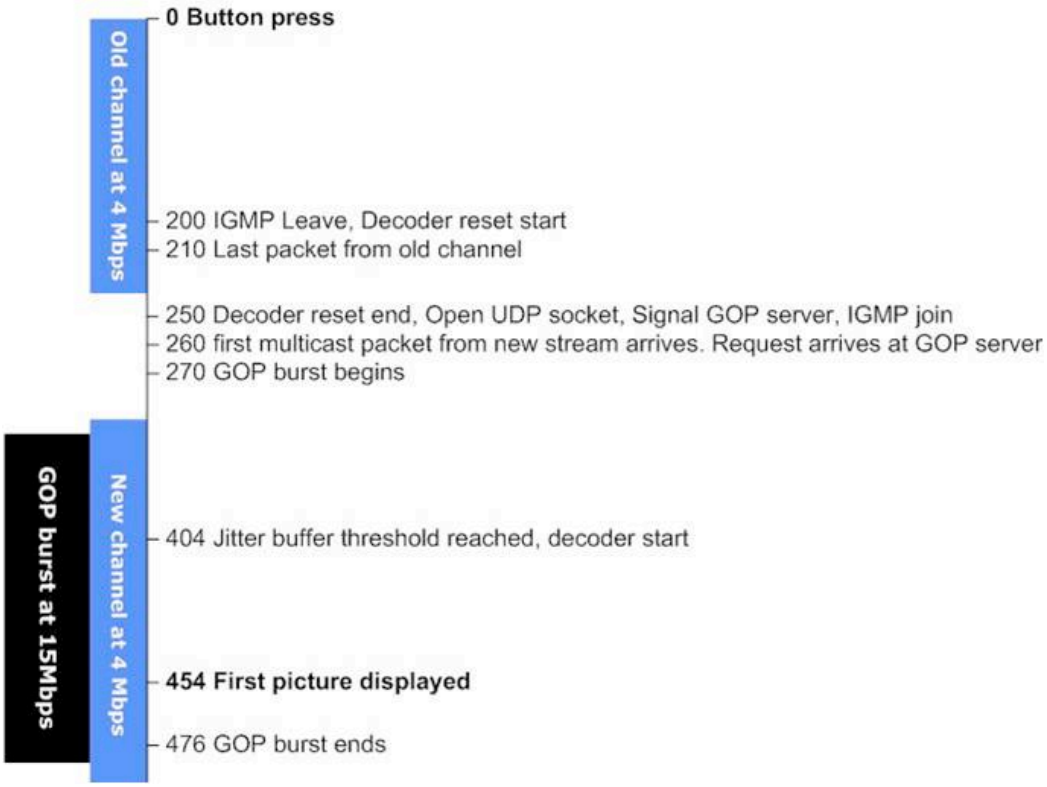


Figure 49

Timeline Discussion

In the above example the proposed improvement reduces actual channel zapping time by 606ms. It is worth noting that if STB associated delays (reaction, decoder reset, decoder start) were zero, the picture would appear in only 154ms (20ms roundtrip + 134ms for first GOP and buffer threshold) from remote button press which is about 5 PAL frames. This is very close to an analogue TV experience.

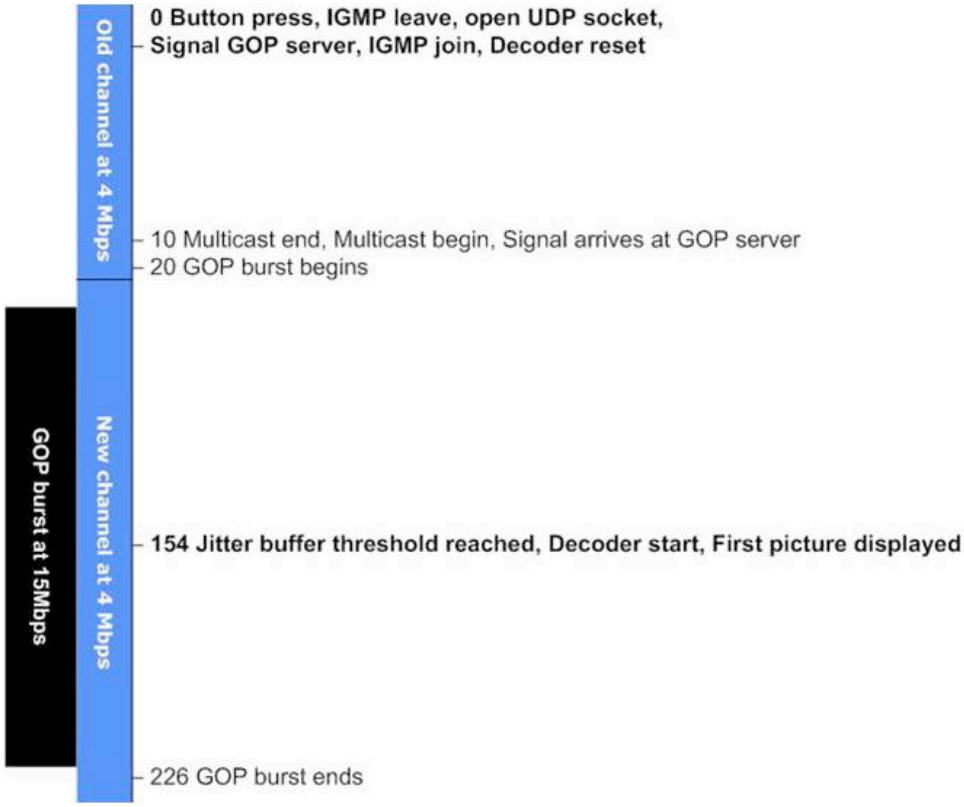


Figure 50

A solution like this could be quite a natural extension for companies within the video server field. GOP Server functionality would be added as an additional service. The business case for service providers to add servers to enable a fast-channel zapping experience for its customers could easily be calculated and would have similarities to VOD network capability cost calculations. More and more servers can be added upon demand, based on what level of experience the service provider would like to enable for its subscribers. I could also foresee a project like this being implemented as an open source cooperation project between Industria, several vendors and service providers, in terms of fixing one of our main challenges in enabling a true television experience over IP networks.

Chapter 4

1 Fast Channel Change alternative use: User Traceability

Since the 1950's, television has been a dominant and pervasive mass media; it is watched across all age groups and by almost all countries in the world.

Alcatel-Lucent has enhanced its Triple Play Service Delivery Architecture (TPSDA) with **more capacity**, **better control** and **higher scalability** for mass-market multimedia services. New capabilities are aimed at providing distributed, fine-grain policy enforcement with centralized policy control to deliver multiple services to subscribers, including **managed services** such as IPTV and voice as well as **unmanaged services** such as High Speed Internet (HSI). The goal is to enable per-subscriber, per-service and per-application control in the network for millions of users. TPSDA 2.0 is a more cost-effective and flexible platform for high bandwidth services such as HDTV, an improved IPTV user experience. It features immediate channel changing and more reliable TV service, and new capabilities to facilitate the insertion into TV programs of advertisements that are targeted to specific communities or localities. This will enable the development of "new TV advertising business models". The declared goal is to enable smart, IP-based, video-centric networks that offer subscribers a consistent and **high-quality**, **personalized** and **interactive viewing experience**.

According with the need of a better experience for the end user Alcatel-Lucent TPSDA 2.0 supports the **Fast Channel Change** for the IPTV service. This system helps users finding the right channel quickly and this factor is imperative to both improving end user quality of experience and minimizing the impact in the network.

The process of switching a channel onto a subscriber's broadband connection is triggered by the subscriber pressing the remote control. When detected by the STB, a signal is sent into the network and the requested channel is switched onto the subscriber's broadband connection via multicast. The Innovation introduced by the FCC is the introduction of a server named FCC server that detects the channel change request and sends a burst of unicast video content,

beginning with the I-frame, to the subscriber's STB with enough information to allow an immediate channel change along with several seconds of video information to play out while the STB synchronizes with the new multicast stream.

From the use of the unicast transmission comes the idea of the **User Traceability.** The purpose is to describe the User behaviour watching TV in order to characterize the user arrival and departure patterns across channels, which could be used as an input to design future IPTV systems and create a new advertising model. The User Traceability provides an analysis focused on aggregate user behaviour information to understand the impact in the network and it can help the SPs making some better marketing strategies and plans, but does not characterize viewing patterns of individual users.

2 Current studies on TV viewing in Italy: Auditel

In Italy the society that surveys the television audience through all the different technology is Auditel and this is a "super parts" society.

Auditel works with an organization recognized as the most advanced international level, a formula "tripartite", governed by a "JIC" (Joint Industry Committee), which gathers together all the components of the television market: companies that invest in advertising, agencies and centers media, business issuers. In line with the guidelines developed by the European Community, Auditel is a "consortium" which is structured as the "tripartite", i.e. a company that is in harmony, the three fundamental components of the market:

• Users Companies, Agencies and Media Central.

• Public Networks.

• TV private, national and local.

Auditel is a system of accountability and control "cross" between players in competition. Reliability and independence are essential conditions of Auditel.

The reliability comes from the constant technological and control methodology. The independence of the corporate formula.

National networks, local broadcasters, thematic channels, digital terrestrial and satellite elements of a television system in continuous evolution. Magnitude and composition of television made it

necessary to have an advanced system of audience measurement. Every day, the information gathered, the investments are planned advertising and editorial choices made, is guaranteed a "snapshot" of the means of communication also useful to the institutions. Through a system of rigorous and transparent search, which provides detailed information on volumes of listening and choices of different audiences, Auditel is a guarantee of clarity for all.

2.1 Identification of the Behavioral information

In Italy, as in every country all around the word, the audience knowledge is a key factor for Television operators, advertisers and also institutions.

Using the audience survey is possible :

- to plan investment of users to advertising companies
- to evaluate the performance of programs
- to provide evidence to improve television
- to analyze the behaviour of public

Despite the widespread usage of television and its importance to emerging applications, the ingrained TV **viewing habits** are not completely understood. Many Media Research Companies spearheads a long-standing research effort to estimate TV viewing behaviours through monitoring and surveys. However, due to the difficulty of equipping monitoring devices at individual homes, it is hard to monitor user behaviour across the entire network. Sophisticated methods such as stratified sampling, systematic sampling, and cluster sampling are used to find a representative set of users and extrapolate their behaviours to the entire population. The recent large-scale deployment of IPTV systems is a big opportunity for the Service Providers towards understanding TV viewing habits. IPTV systems give more visibility on TV viewing activities across an entire network. At the same time, the large user base provides a clearer picture of how people watch TV across different groups of users (location, content genre, etc).

2.1.1 How to gather it

In order to have a picture of the user activities in the network, the SPs should create a system that automatically collects "User Data" in a suitable repository such as the Service Management System. Each "user data" has to contain:

- Timestamp in unit of seconds
- IP address of the STB
- IP address of the multicast group
- IP address of the DSLAM
- Multicast option of join and leave
- Unicast option

In order to cater for customers' channel zapping behaviour in browsing channels which will generate large amount of useless data, the system should pre-process the data excluding non-video multicast groups. It is, also, useful to make some assumptions related with the user activity modes and channel holding time.

User activity modes

It is possible to divide user's activity in three different modes:

- Surfing
- Viewing
- Away

These modes are related with the **channel holding time** that is the time interval between channel switching.

In Surfing mode the user is searching for a channel, in Viewing mode the user is watching that channel and in an Away mode the user in turning off the TV. In this last mode we can consider also another situation where the TV is left on without anyone watching it. It could be difficult to differentiate these two options, because many users leave the IP set-top-box on

and continue receiving multicast streams, even when the television is off.

According with different studies and also with the media research companies the thresholds between the modes could be:

- **1 minute** from Surfing to Viewing
- **1 hour** from Viewing to Away

Although these divisions may somehow look arbitrary, they coincide very well with those divisions made by the traditional media research. If the SP likes to compare his data with the ones proposed by Auditel it can set the same viewing thresholds. It is important to consider that the results are not sensitive to thresholds changes, for instance, from 1 minute to 5 minutes or from 1 hour to 2 hours.

Channel genre

IPTV providers can group the channel following the genre criteria:

- free
- kids/teens
- culture/education
- local
- cinema
- sports
- music
- news
- mixed (comedy, soaps and reality shows)

According with this kind of assumptions it is interesting to understand what kind of information the SPs can gather from the network in order to compare them with the ones provided by Auditel.

It is possible to characterize several properties of the TV viewing behaviour that describe

how, when, and what people watch:

• User session characteristics: when people watch TV and what is their attention span across genre.

• Channel popularity and dynamics: how user interests are spread across channels over time.

• Geographical locality: whether users in the same region or DSLAM show similar viewing patterns.

2.1.2 How to use it

Starting from the data in the repository is possible to obtain the behavioural information.

From timestamps it is possible to obtain the information about the duration of the on-line time of the user.

- The IP address of the STB is used in order to identify the user
- The IP address of the multicast group is used in order to understand the content that the users are watching
- The IP address of the DSLAM is necessary if there is an interest on understanding similarity among users' interests in the same geographical area.
- Multicast option of join and leave and Unicast option are used to know when users change channel.

2.2 Benefits and repositioning from User traceability

The presented solution comes from the idea that Alice Home TV needs of some improvement in order to have a repositioning. Telecom Italia, as a Service Provider, could collects "User Data" and process these providing audience information. That could be a big change in the Television scenario and this could be a strength factor in the relation between Telecom Italia and the Media Company and the Content Providers.

2.2.1 Current study on TV viewing in Italy: Auditel analysis

By applying a rigorous statistical methodology, Auditel has constructed a representative sample of the Italian population (all individuals aged over 4 years, ISTAT data residing on their territory). This continuous sample (panel) is a kind of "condensate" of the population with its diverse geographical, demographic and socio-cultural. An electronic device, the meter automatically detects every day, minute to minute, listening to all channels of any TV that is running in your household sample. The "meter systems" are now used for their reliability, in all the most advanced countries.

Published Broadcasters

At present (the site may be affected by the timing of updating the list of issuers who have requested the publication of data) Auditel distributes market data obtained by listening:
• national broadcasters (terrestrial analog, digital and satellite).

— Their daily data minute by minute are published for single issuer (with a sub-total TO PUBLISHER). The layout of the publication also includes the other clusters, complementary and total: other terrestrial analog, satellite broadcasters, total other, the total issuers.

— Other issuers are published (on request) on a monthly average is the case, in particular of some satellite channels.

• Local broadcasters

Auditel publishes monthly data for approximately 140 local broadcasters, operating in the region.

There are about 70 regional broadcasters - at their request - with daily data, minute by minute, given the need for planning (advertising and not editorial).

The basic survey

The statistical foundations of the project comprises the so-called "basic research" means a continuing series of surveys on the general Italian families who feed divided by 9 monthly, a large database. Game after a very extensive research on 41,000 cases, Auditel interview, every year, a sample of 30 thousand families - in their homes and not by telephone - to assess, among other things, the allocation of television equipment and entertainment (video, satellite links,

digital terrestrial and cable TV, DVD, pay-tv, etc..). The research conducted by Auditel is the most important "snapshot" of the television phenomenon implemented in Italy by:

• sample size

• dispersion of the sample (over 1700 municipalities sampled)

• depth of information collected

• methodology used (CAPI, face-to-face)

The "reservoir" of basic research will extract the sample for the panel "meterizzato". The questionnaires are an indispensable tool for the selection of the sample and the subsequent weighting. Indeed, the structural characteristics of the population represented by Auditel are derived in part from information provided by ISTAT, estimated in part through basic research.

The Sample

The families included in the sample are extracted in a random and anonymous, and come from research-continuous basis. The resulting sample has a composition and a weighting system that can properly represent the collective reference, more broadly, from which it was extracted. The representation is based on:

• a system of recruitment of cells in which cross geographical variables (area and size of municipalities), structural characteristics of households (age of head and number of components), allocation of television equipment;

• A dual system of expansion (one for the family and one for those individuals) characterized by a "pre-expansion" for cells and a subsequent "weighing iterative marginal."

The sample is allocated on the 103 Italian provinces in proportion to population (except the province of Aosta and Cagliari have arisen where institutional needs to monitor the "test" on DTT). The spatial dispersion of the panel allows you to cover about 2090 of the 8100 Italian municipalities.

The Meter

The families of the panel are equipped with electronic equipment called "people-meter" that automatically detects the channel tuned on the TV. Schematically, the meter is composed of 3 parts:

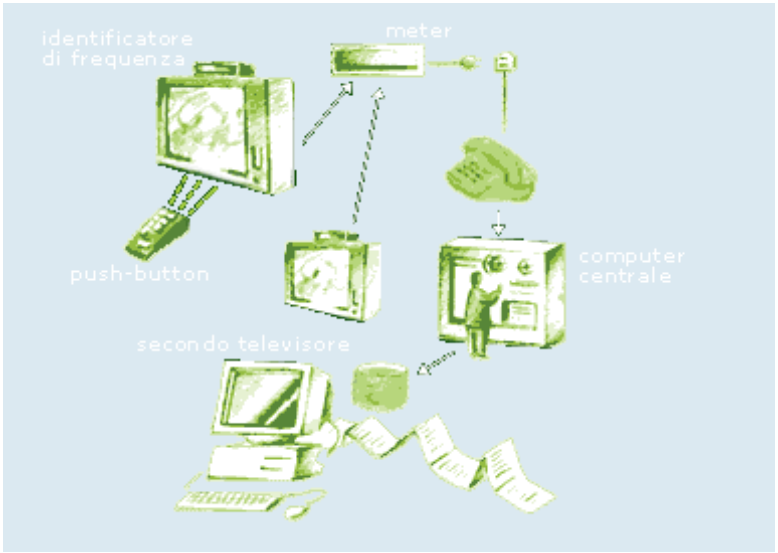
1. Unit identification: recognize and record the TV channel received by each unit in the family (TV, VCR, DVD, satellite TV receiver and digital terrestrial, play station);

2nd Remote control: recommend attendance for each individual TV set through keys assigned to each member of the family and guests;

3rd Units of transmission: collecting the data (from the TV) and then pass it on to the central computer via telephone line or GSM.

Produced by AGB, a company responsible for the collection, the meter is the property of Auditel. The meter is connected to every TV in the family sample. Every day, minute by minute, the meter collects the figures of both components of the family that the guests will be present. The numbers of the panel Auditel initiated the collection of audience in December 1986. Since August'97 the number in the sample is doubled.

Today, the collection system works with **5163 families**: more than **9500 detectors meter**, active on many television sets, "photograph" the choices of about **14,000 users** at any time of day. At present, the "panel" Italian Auditel is a sample search of television among the most numerous in the world (the ratio population / meter). And it is between those who invest more resources in monitoring activities.



Recognize the TV sources

After installation of new equipment, Auditel (since January 2002) notes listening to satellite television. Not only now, with the renewal of the "park meter", Auditel can satisfy the complete tracking of new television technologies and is already active to protect the gradual transition from analogue to digital terrestrial. Since 2005 there is a total conversion of the technologies in use through the adoption of UNITAM meter. For each sample household to equip itself with digital technologies (digital terrestrial, satellite, PVR, cable, etc...) it is now installing the new meter. These allow a collection of listening to everything from independent broadcasters, as they have adopted an innovative system where the recognition of the channel is through the comparison of "tracks" digitized. Details of collection stations record, for each channel, broadcasts the day by creating a digitized database (reference) to be compared with the individual audio tracks (sample) collected in the sample households on the basis of the acts of listening. This will determine, with certainty, the channel / broadcaster on television tuned monitored. This update is done every day - continuously - and allows you to search Auditel of "cover" all the different sources of irradiation of the signal. Even with the meter TVM4 method for recognition of different sources of television signals was open to different solutions. The satellite and DTT could be efficiently served by the technique called PDC (Program Delivery Control), which was recognized by the EBU identifier code generated by each channel and incorporated in the television signal through a system of "data broadcasting". An alternative highly secure - also used at international level (e.g. in the United Kingdom) - provides that the meter draws directly from the serial port of the decoder, the **SI** (Service Information), which allow the recognition of all the programs broadcast. It is currently on SKY decoders of subscribers.

Data production

The data collected by the meter is collected and validated procedures with sophisticated analysis and control. A special process looks at each individual and calculated the factor of expansion. It is a system of weighting of the units of the sample to obtain accurate estimates compared to the parameters considered. This process of "weighing" allows Auditel to reliably represent all of the choices of television consumption in the country. Knowing the data: The data

collected allow knowing:

• Audience Rating: average number of viewers of a program. It is equal to the ratio between the amount of viewers in each minute of a given time interval and the duration in minutes of it.

• Share: percentage ratio between the listeners of a certain issuer and the total number of listeners who are watching any other program on the different networks.

• Penetration: percentage ratio between the listeners of a certain class and their statistical universe of reference. For example, how many kids are 15 years of that program compared to the total 15enni not watch television at that time?

• Contacts: they are all individuals, different from each other, who see at least 1 minute of a certain program. There are only once.

• Minutes: is the average number of minutes viewed by the viewers for each program. It is equal to the ratio of the average audience of that program, multiplied by the duration and divided by the net contacts.

• Permanence is an indicator of fidelity of vision. Is obtained as the percentage ratio between the number of minutes viewed on average by listeners of a certain program and life itself.

What is measured

The television ratings, minute by minute, for programs, breaks and commercials broadcast by local and national broadcasters in Italy. The ratings of the guests. All modes of transmission: terrestrial, satellite, digital terrestrial ...

The audience of families, as a group of people living under one roof and head resident in Italy. The ratings of the individuals if they are present in the room where the TV is located and are actually listening to (including the use of Televideo / Teletext).

What is not measured?

All external ratings and the main housing of:

• Children under 4 years of age

• Italian broadcasters found in other countries

• community.

Public analyzed

I found different audiences, namely the parameters considered by Auditel and used for the analysis of listening ("who sees what") vary from 60 per day to about 80 per month. The public is split analyzed a number of targets that are different depending on the gap of time you are analyzing:

• Daily and weekly

• Monthly and annual

• Monthly and annual (regional)

Data Auditel add some psychographic profiles called "lifestyles". These criteria allow a classification related to psychological factors and socio-cultural, incorporating the classic geographic or demographic criteria. Conducted in collaboration with Eurisko, the initiative also addresses the attention to quality aspects and behavior.

Checked for penetration of the signal in the area, Auditel notes all broadcasters who request them. The system of research, in fact, is able to detect in an equitable manner the audience of each subject, although, for the publication of data are needed:

• a prior assessment of a technical / statistical;

• the explicit request of the broadcaster that wants to be monitored.

2.2.2 Advantages of User Traceability compared to current solutions

According with the methodology used by the Media Research Companies to get behavioural information there are some known issues:

- Data are produced on a sample of the total users (**14.000 meter/60 mil. Population in Italy)**
- Selected households are aware of being monitored and, even if it is not possible to

understand how this can bias the statistics, this awareness could have an influence on viewing habits (After the first phase with the "base analysis" about a 30% of the selected households refuse to be in the study)

- It's necessary to install extra devices in the houses

In contrast, IPTV providers can get behavioural information:

- Data are produced evaluating ALL the users (**330.000 today number of users** for Alice Home TV)
- Households are Not aware of being monitored
- It is NOT necessary to install extra monitoring devices.

It is important to notice that total number of Alice users is higher of the sample used by Auditel. Another advantage of this possibility is the disagreement of content providers(e.g. Sky) in the reliability of the data produced by Auditel. This opinion depends on the fact that Auditel is controlled by the main shareholder by the CdA.

However, IPTV providers lack information about what the individual members of each family are watching because is not possible to have information about the family member who is watching television in that moment.

Conclusion

The aim of this work was to describe the FCC technique in order to provide an idea of the necessary time for the channel zapping. Together with this study there is also a proposition for the future implementation of a model describing the viewing habits of people watching TV. This study wants to take advantages from the characteristic of the use of IP network for delivering television. In this way Service operators have the power of the knowledge of what people watch.

3 Acronyms and definitions

Acronym	Definition
DRM	Digital Right Management
HD	High Definition
IGMP	IP Group Membership Protocol
IPTV	Internet Protocol Television
ISP	Internet Service Providers
ITU-T FG IPTV	International Telecommunication Union Focus Group IPTV
QoS	Quality of Service
RTSP	Real Time Stream Protocol
SD	Standard Definition
STB	Set-Top-Box
VoD	Video on Demand
VoIP	Voice on IP
TCP	Transport Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol

Table 3-1 Acronisms and Definition

4 Bibliography

· Directorate for science, technology and industry committee for information, computer and

communications policy,2007, “IPTV:Market developments and regulatory treatment ”, Web

· Wikypedia, “IPTV”, web-link founded on 2009/02/13

· “Understanding IPTV”, cap.1, cap.4

· RFC 791: INTERNET PROTOCOL , DARPA INTERNET PROGRAM, September 1981

· IP multicast

· RFC 988: Host Extensions for IP Multicasting, Appendix I ,July 1986

· RFC 2236: IGMPv2, November 1997

· RFC 3376: IGMPv3, October 2002

· ISIMM, Analisi del mercato della IPTV – *Le forze competitive,pag 21-30*

· *Alcatel-Lucent web-site,2009, TPSDA 2.0 Assured and Optimized Content Delivery, pag 1-6*

· http://www.grassvalley.com/news/2008/20081125-ThomsonFastChannelChange.html

· http://tools.ietf.org/id/draft-versteeg-avt-rapid-synchronization-for-rtp-02.txt pag. 1-24

· http://www.rfc-editor.org/rfc/rfc4585.txt pag 32

· “IPTV, la televisione arriva con l’ADSL”, Notiziario tecnico Telecom Italia anno 14 n°2, pag.47-68

5 Figures and tables

Non è stata trovata alcuna voce dell'indice delle figure.

6 Appendix A

"Eventuale"