

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Towards a Scalable Video Interactivity Solution over the IMS

Prepared by:

Lesang Vincent Dikgole

Supervised by:

Neco Ventura

University of Cape Town
Department of Electrical Engineering



This thesis is submitted to the Department of Electrical Engineering
in fulfillment of the academic requirement for the Degree of Master of
Science in Engineering

August 1, 2011

Declarations

I declare that this thesis is my own work. Where collaboration with other people has taken place, or material generated by other researchers is included, the parties and/or material have been referenced or acknowledged as appropriate. This work is being submitted for the degree of Master of Science in Electrical Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

"I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own"

Lesang Vincent Dikgole

Neco Ventura

Date

Acknowledgments

I would like to thank the following people and organizations:

- My sponsors Telkom for granting me a bursary to enable me to pursue further studies.
- My supervisor Neco Ventura for providing guidance and support for my work.
- My Communications Research Group (CRG) colleagues for their input and support throughout the project. I particularly want to thank Robert Marston for his work on the UCT Advanced IPTV Server which proved to be immensely useful for the evaluations performed in this study. I also want to thank Keoikantse Marungwana for the intense meaningful discussions that transcended our love for engineering we had in the lab. Other veterans of the CRG group such as David Waiting, Richard Good, Eugene Golovins and Vitalis Ozianyi also provided much needed guidance, instruction and support that I will be ever thankful for.
- My Palmer house mates: Kurd Maluleke, Sizwe Zondo, Chris Louw and James Wood for their patience and support throughout the tough times while I stayed late on campus and when I became grumpy at times as a result of this.
- My friends, Mawethu Ncaca, Madalitso Phiri and Chris De Witt for enlightening me in other spheres of life outside of engineering: philosophy, politics, faith and what it means to be fully human.
- This study proved to be an immense challenge over time, the possibility of the final culmination of the research output was put in serious question by a number of struggles that I would not have managed to overcome without many of my other friends who helped me, I am very thankful to God for them. For these reasons, I owe a great debt to my family: my grandmother Salome Dikgole, my mother Rheniah Dikgole and my brother Gerald Dikgole.

Soli Deo Gloria

Synopsis

The current rapid increase in bandwidth, and the interactive and scalability features of the Internet provide a precedent for a future converged platform that will support interactive television. Next Generation Network platforms such as the IP Multimedia Subsystem (IMS) have been developed to support Quality of Service (QoS), fair charging and possible integration with other services for the deployment of IPTV services. The IMS architecture supports the use of the Session Initiation Protocol (SIP) for session control and the Real Time Streaming Protocol (RTSP) for media control.

There are suggestions in the literature that the SIP protocol should also be used for media control (or 'video interactivity') and there is currently no thorough conclusive evaluation and analysis of the use of SIP for video interactivity in the literature. In addition to this, the possible impact of interactivity requests from various protocol architectures, such as SIP and RTSP, on the video processing power needs to be investigated. There are also suggestion in the literature that these protocols need to provide a framework, in their design, for supporting a greater variety of video interactivity requests.

This study aims to investigate currently existing video interactivity designs over the Internet. An evaluation framework to examine the performance of both SIP and RTSP protocols over the IMS over different access networks will be developed. This study will also carry out an investigation on the relationship between the amount of video interactivity requests and the video processing load on a video server. This study proposes a Three Layered Video Interactivity Framework (TLVIF) that aims to reduce the video processing load on a video server.

This study provides the results of the tests carried out to investigate the relationship between video interactivity and the video server processing load. A SIP-based architecture that aims to provide an enabling platform for future interactivity requests will be presented. The architecture will also aim to minimise the impact of interactivity on the video processing load of a video server. The current UCT IMS client will be extended to support video interactivity modes; an extension of the SIP stack will be developed to support SIP-based interactivity and an RTSP stack will also be implemented to support RTSP-based interactivity. A measurement and analysis on the performance of RTSP and SIP will be provided. The proposed TLVIF is also evaluated and tested. The performance results (e.g. end-to-end playback delays and the server processing load) obtained in this study will serve to contribute towards further discussion on the viability of future interactivity signalling protocols that minimise the impact of many video signalling requests on video system performance.

Glossary

ADSL	Asymmetric Digital Subscriber Line
AS	Application Server
CoD	Content on Demand
CRG	Communications Research Group
CS	Circuit Switched
CSCF	Call Session Control Function
DHCP	Dynamic Host Control Protocol
DRM	Digital Rights Management
DVB	Digital Television Broadcasting
EDGE	Enhanced Data Rates for GSM Evolution
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
GSM	Global System for Mobile Communications
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
I-CSCF	Interrogating CSCF
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IMS	IP Multimedia Subsystem

I/O	Input / Output
IP	Internet Protocol
IP-CAN	IP Connectivity Access Network
IPTV	Internet Protocol Television
ISP	Internet Service Provider
ITU-T	Telecommunication Standardisation Sector of International Telecommunications Union
LAN	Local Area Network
LTE	Long Term Evolution
MDF	Media Distribution Function
MoD	Music on Demand
NGN	Next Generation Networks
P-CSCF	Proxy CSCF
PSTN	Public Switched Telephone Networks
PTToC	Push To Talk over Cellular
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Access Memory
RTP	Real-time Transport Protocol
RTCP	Real Time Control Protocol
RTSP	Real Time Streaming Protocol
SCF	Service Control Function
S-CSCF	Serving CSCF
SDO	Standards Development Organization
SDP	Session Description Protocol

SIP	Session Initiation Protocol
TAM	Technology Acceptance Model
TCP	Transmission Control Protocol
TISPAN	Telecommunications and Internet Converged Services and Protocols for Advanced Networks
TLVIF	Three Layered Video Interactivity Framework
TS	Technical Specification
TV	Television
UATV	UCT Advanced IPTv Server
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
URI	Uniform Resource Identifier
VI	Video Interactivity
VoD	Video on Demand
VoIP	Voice over IP
WAP	Wireless Application Protocol
WiMAX	Worldwide Interoperability for Microwave Access, IEEE 802.16 Standard
WLAN	Wireless LAN
XML	Extensible Markup Language
3GPP	Third Generation Partnership Project

Contents

Declarations	i
Acknowledgments	ii
Synopsis	iii
Glossary	iv
Contents	vii
List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Background Information	1
1.2 Video on Demand over IMS	3
1.2.1 IMS and IPTV	3
1.2.2 IMS-based Video Interactivity	6
1.3 Problem Statement	7
1.4 Thesis Objectives	9
1.5 Scope and Limitations	9
1.6 Thesis Contributions	10
1.7 Thesis Outline	11
2 Literature Review	12
2.1 Video Interactivity Protocol Architectures	12
2.1.1 RTSP-based Video Interactivity	12
2.1.2 SIP-based Video Interactivity	13
2.1.3 Other IMS Video Interactivity Solutions	15
2.2 Scalability in Video Interactivity Protocol Architectures	17
2.2.1 SIP Signalling impact on Proxy Server Scalability	17
2.2.2 Interactivity Signalling impact on Video Server and Bandwidth Scalability	18
2.3 Platform for Future Video Interactivity Development	20
2.3.1 Evaluating Quality of Experience for Video Interactivity	21

2.3.2	Video Interactivity Requirements	22
2.3.3	Current Video Interactivity Designs	23
2.4	Chapter Summary	24
3	The IMS-based Three-layered Video Interactivity Framework	25
3.1	Requirements of the Video Interactivity Architecture	26
3.1.1	Metrics for the Design	26
3.1.2	Design Requirements	26
3.2	The Proposed Three Layered Interactivity Design	27
3.2.1	A SIP-based Video Interactivity Architecture	29
3.2.2	A Three-Layered Video Server Architecture	31
3.2.3	A SIP-based Platform for Interactivity Development	37
3.3	Chapter Summary	39
4	Architecture of the Evaluation Platform	41
4.1	Requirements of the Evaluation Platform	41
4.2	Decision on Test-bed Implementation and Tools Used	41
4.3	Test-bed Software Overview	42
4.4	Test-bed Software Developed	44
4.4.1	UCT IMS Client	44
4.4.2	TLVIF Layers Implemented	46
4.4.3	The TLVIF Video Server	47
4.5	Test-bed IP-CANs	47
4.6	Setting Up the Evaluation Environment	48
4.6.1	SCF-MCF-MDF and The Open IMS Core	48
4.6.2	Setting up the SCF-MCF-MDF	49
4.6.3	Hardware Specifications	50
4.7	Setting up Measurements	50
4.7.1	Latency Measurement	50
4.7.2	Server Processing Load Measurement	52
4.7.3	Bandwidth Measurements	53
4.8	Chapter Summary	54
5	Evaluation Results and Analysis	55
5.1	WiMAX and Ethernet-based Testbed Environments	55
5.2	Evaluation of Video Interactivity Latencies by RTSP and SIP	56

5.2.1	Interactivity Delays by RTSP and SIP protocols - Without MDF Processing	56
5.2.2	Interactivity Delays by RTSP and SIP protocols - With MDF Processing	59
5.3	Evaluation of Video Interactivity and Video Server Load	61
5.3.1	Signalling Processing Overhead	61
5.3.2	MDF Processing Overhead	62
5.3.3	SIP-based Interactivity vs. Video Server Load Relationship	63
5.4	Performance Evaluation of Emulated TLVIF	64
5.4.1	TLVIF QoE	65
5.4.2	TLVIF Performance	66
5.5	Chapter Summary	68
6	Conclusions and Future Work	69
6.1	Summary	69
6.2	Conclusions	70
6.3	Recommendations and Future Work	70
A	IPTV Business Models and Current IPTV Standards	77
A.1	Telco Example	77
A.2	Why some IPTV Trials Fail	77
A.2.1	Flexibility	78
A.2.2	Good Content in the Media	78
A.2.3	Interoperability between Different Architectures	78
A.2.4	Integrated Services	79
A.2.5	The Need for an IPTV standard	79
A.3	Requirements of IPTV Standards	79
A.3.1	ETSI TISPAN	80
A.3.2	ITU-T IPTV Focus Group	80
A.3.3	Open IPTV Standard	81
A.3.4	ATIS IIF	82
A.3.5	3GPP MTV MBMS over 3G	82
A.3.6	The TV Anytime Forum	82
A.3.7	OMA	83
A.3.8	Video on Demand and the IP Multimedia Subsystem	83
B	Protocols' Overview and Implementation	85
B.1	RTSP	86

B.2	SIP	87
B.3	Video Interactivity Signalling Implementation	89
B.3.1	RTSP Trick-play	89
B.3.2	SIP-based Trick-play	89
B.4	Media Processing Layer Implementation	90
B.4.1	Implementing Pause and Play	90
B.4.2	Implementing Forward and Rewind	90
C	Bandwidth Estimations	92
C.1	MDF Video Server Setup	92
D	Ethics of Content, Hardware and Software Used	94
E	Accompanying CD-ROM	95

University of Cape Town

List of Figures

1.1	High Level VoD Architecture	2
1.2	IMS Architecture [1]	4
1.3	Signalling and Media Processing	6
1.4	Signaling and Media Processing Diagram	8
2.1	Hybrid RTSP/SIP based Media Control	13
2.2	SIP-based Video Interactivity, Sivasothy et al [2]	14
2.3	The Extended IMS Architecture[3]	15
2.4	The Enhanced IMS Architecture[3]	16
2.5	Scalability versus Interactivity	18
2.6	Flow Diagram - Download and Play	19
2.7	Flow Diagram - Content-caching	19
3.1	Interactivity and Video Server Load Overview	25
3.2	TISPAN VoD Architecture [4]	28
3.3	SIP-based Video Interactivity Framework	29
3.4	The Proposed Modified SIP Message	30
3.5	SIP-based Video Interactivity Sequence Diagram: Pause	30
3.6	SIP-based Video Interactivity Sequence Diagram: Rewind	31
3.7	Proposed Three Layer Interactivity Framework	32
3.8	The TISPAN User profiles with the Proposed 'Interactivity Profile' Extensions	33
3.9	Interactive Framework Layer	34
3.10	Media Processing Layer	35
3.11	Sequence Diagram for the TLVIF	36
3.12	Sequence Diagram for the TLVIF: Rewind, Forward	37
3.13	The Video Adaptation SIP Message	38
3.14	Sequence Diagram for New VI's in the TLVIF	39
4.1	Fokus Open IMS Core Project [5]	43

4.2	The UCT IMS CLIENT	44
4.3	Trick-play Interface	45
4.4	Interactive Platform	46
4.5	WiMAX Test-bed	48
4.6	HSS Interface	49
4.7	Open IMS Test-bed	49
4.8	Latency Measurements	50
4.9	Measuring RTSP Latency	51
4.10	Measuring SIP Latency	52
4.11	Interactive Layer Emulation	53
5.1	RTSP Tests Diagram	56
5.2	SIP Tests Diagram	58
5.3	SIP With MDF Test Diagram	60
5.4	Graph showing CPU Load for With and Without the Signalling Load	62
5.5	Graph showing CPU Load for Signalling Only vs. with Video Processing	63
5.6	The Server Processing Load With and Without TLVIF Graph cf. Signalling Only	67
A.1	Video on Demand Standards	83
B.1	Signaling and Media Processing	85

List of Tables

3.1	Video Interactivity Functions	38
3.2	Proposed vs. Existing Solutions	39
4.1	Implemented TLVIF Server Methods	47
4.2	Software and Hardware Specifications	50
5.1	WiMax Bandwidth Link Estimation	55
5.2	Ethernet Bandwidth Link Estimation	55
5.3	RTSP Latencies Without MDF	57
5.4	Network Delay Percentage Change for RTSP-based Video Interactivity	57
5.5	SIP Latencies Without MDF on WiMAX/Ethernet	58
5.6	Network Delay Percentage Change for SIP-based Video Interactivity	59
5.7	Latency Percentage Change for RTSP vs. SIP based Video Interactivity over WiMAX	59
5.8	Latency Percentage Change for RTSP vs. SIP based Video Interactivity over Ethernet	59
5.9	SIP Latencies With the MDF on Ethernet	60
5.10	Processing Load Averages With and Without the Signalling Load	62
5.11	Processing Load Averages with Processing	63
5.12	Processing Load Percentage Change With and Without Video Processing	64
5.13	Processing Load Percentage Change From Different Call Rates	64
5.14	QoE Performance of TLVIF at 0.5 calls per second	65
5.15	QoE Performance of TLVIF at 2.5 calls per second	65
5.16	QoE Performance of TLVIF at 5 calls per second	66
5.17	Processing Load Averages with Video Processing	66
5.18	Processing Load Percentage Change With and Without TLVIF cf. Video Processing	67
5.19	Processing Load Percentage Change With and Without TLVIF cf. Signalling Only	68
C.1	WiMax Link Speeds	92
C.2	Ethernet Link Speeds	93

Chapter 1

Introduction

1.1 Background Information

To date, it is estimated that more than 1.6 billion people use the Internet [6]. The Internet offers simple services such as e-mail, Instant Messaging, and enhanced services such as online videos, and social networking sites (e.g. Facebook and MySpace) which have become ubiquitous on mobile phone devices. On the other hand, television, which began in the early 1930's with the first black and white television broadcast and was followed by colour television a decade later, also offers interactive features. The first interactive television service that integrated voice, data, image and video was introduced in 1994 by Time Warner Cable company in Orlando, Florida [7]. The trial was offered to approximately 4000 subscribers and provided services such as Video On Demand, home shopping, interactive programming guide and games. The trial was however closed in 1997 due to rising costs and lack of premium content. The problems experienced in this trial are common to many IPTV-VoD deployments; some of these well-known IPTV deployment issues are discussed further in Appendix A.

Most television channels such as ABC, NBC, CBS, HBO and DStv are beginning to provide some of their premium content online [8, 9, 10, 11, 12]. The benefits of offering a television service over the Internet are:

- That users can conveniently access content on demand
- The ability to have playback control over the accessed video content
- That advertisements can be customised for clients and made less intrusive
- Being a packet based infrastructure, components of storage and transmission are distributed and therefore provide scalable support for more users

The interactive features of the Internet provide the necessary enabling features for future Internet-based interactive television. The provisioning of high-quality Internet Protocol (IP) based video content is made possible by the proliferation of wireless broadband technologies such as Worldwide Interoperability for Microwave Access (WiMAX), and Wideband Code Division Multiple Access (WCDMA) technologies such as Universal Mobile Telecommunication System (UMTS), High Speed Downlink Packet Access (HSDPA), High Speed Packet Access (HSPA), and HSPA+, as well as Long Term Evolution (LTE). This provisioning of high-quality video content over the Internet is referred to as Internet Protocol Television (IPTV).

The IPTV service provided by telecommunications service providers promises to be the main competitor to existing broadcasting services offered via terrestrial, satellite and cable networks. There is an increasing interest from Telecommunication Operators (Telcos) and from television subscribers for television services based on content delivery over IP. For example, Korean Telecoms performed a trial test (in 2007) of an IPTV video-on-demand service with 239 Korean households. Seventy percent of the participants polled afterward indicated a high service satisfaction [13].

There are various types of IPTV services and amongst these is the IPTV-Video on Demand (VoD) service. IPTV-VoD is an IPTV service that allows users to have direct interactivity with the content by choosing content they would like to watch and controlling the streams with the use of playback functions. There are several types of IPTV-VoD¹ services: Push VoD, which entails pushing video content from video server to an IPTV UE hard disk; Movie-on-Demand (MVoD), on demand video delivery of high quality video over a digital network with support for playback functions (D-VoD); Subscription VoD (SVoD) is an MVoD service with regular, often monthly, subscriptions; Television-on-Demand (ToD) allows users to record real-time broadcast TV programs; High Definition VoD, a video service which allows for the retrieving of High definition (HD) video for viewing on large display panels [14]. D-VoD is the focus of this study and will be summarily referred to as VoD in this study.

Figure 1.1 shows a simplified VoD architecture consisting of the following main components:

- Media servers
- Backbone network support
- Transport and transmission support
- Subscriber home network

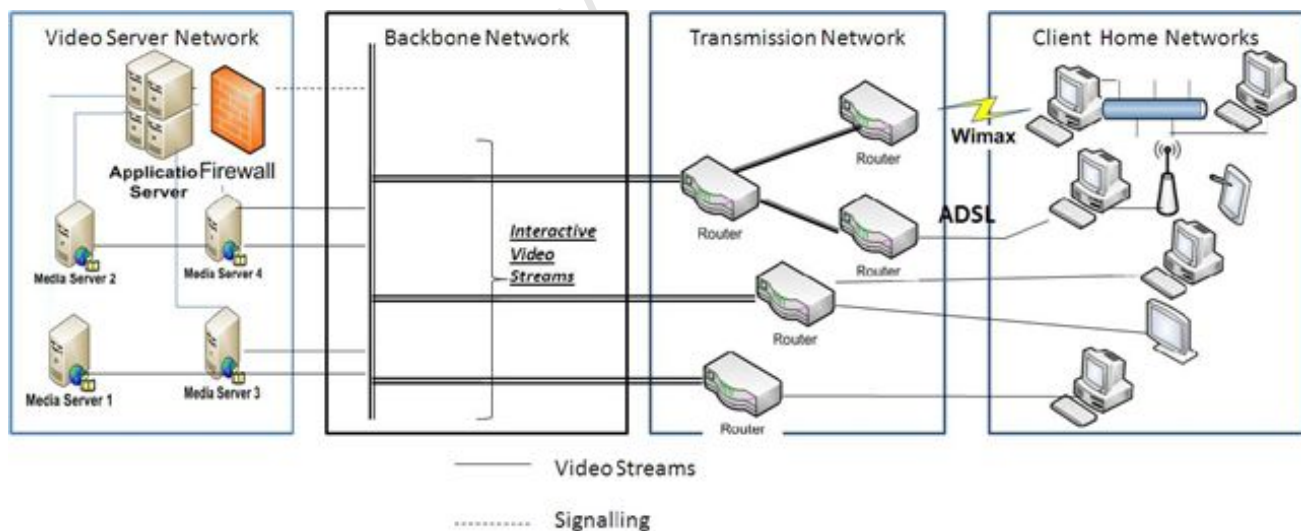


Figure 1.1: High Level VoD Architecture

Media servers consist of the storage and disk scheduling technologies for storing and retrieving video streams; the backbone support is primarily responsible for transporting video data from the storage systems for transport and transmission over a communication network; the network transports and distributes video streams to all the subscribers throughout the IP Connectivity

¹For the rest of the study, IPTV-VoD will be simply referred to as VoD

Access Networks (IP-CANs). An IP-CAN can be any access network technologies such as WiMAX, WiFi, ADSL or 3G. IP-CANs are connected to various subscriber home networks or stations. A subscriber home network could include a Local Area Network (LAN) or a home entertainment system that is customised for a VoD subscriber.

1.2 Video on Demand over IMS

1.2.1 IMS and IPTV

Video on Demand solutions need to be standardised to ensure successful adoption of the service. Installation cycle times can be significantly reduced when a common design standard is adopted. Standards can also enable different players on the VoD market chain to mutually participate in the proliferation of new services and aggregation of content that is attractive to VoD subscribers [15]. Next Generation Network (NGN) platforms have been developed by Standards Development Organisations (SDOs) and telecommunications forums, such as the Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN)², ITU-T³ and the Open IPTV Forum⁴ to provide IP based service control platforms that enable high Quality of Service (QoS). The most prominent NGN control platform for IP based multimedia services is the IP Multimedia Subsystem (IMS). The IMS was developed by the Third Generation Partnership Project (3GPP) with an aim to develop a horizontal control layer, which is separate from Service Delivery Platforms (SDPs), for managing services. This enables the provisioning of services without the need to redesign the control plane for every new service.

The IMS Architecture

The IMS is based on the Session Initiation Protocol (SIP). SIP is used for service control, QoS control as well as for provisioning services. Figure 1.2 shows the IMS architecture.

The IMS control plane (shown in Figure 1.2) comprises the following components:

Home Subscriber Server (HSS): complete information about particular subscribers is stored in the HSS. This includes profiles, policies, subscriptions and preferences for video subscribers.

Proxy-CSCF (P-CSCF): is the first point of contact for IMS users. The main task of the P-CSCF is the processing of signalling messages between the network and the subscribers and allocation of resources for media flows through the interaction with Resource and Admission Control Subsystem (RACS).

Serving-CSCF (S-CSCF): is the main control entity within the IMS. It processes registrations from subscribers and stores their current location. It is responsible for subscriber authentication and call management. Subscriber policies stored in the HSS control the operations performed by the S-CSCF for a particular subscriber.

²TISPAN is European Telecommunications Standards Institute (ETSI) core competence centre for fixed networks and their migration to Next Generation Networks <http://www.etsi.org/tispan/>

³International Telecommunication Union - Telecommunication Standardisation Sector <http://www.itu.int/ITU-T/>

⁴The Open IPTV Forum (OIF) was created in March 2007, to provide an IPTV solution enabling a plug-and-play experience. OIF is a pan-industry initiative with the purpose of producing end to end specifications for IPTV <http://www.openiptvforum.org/>

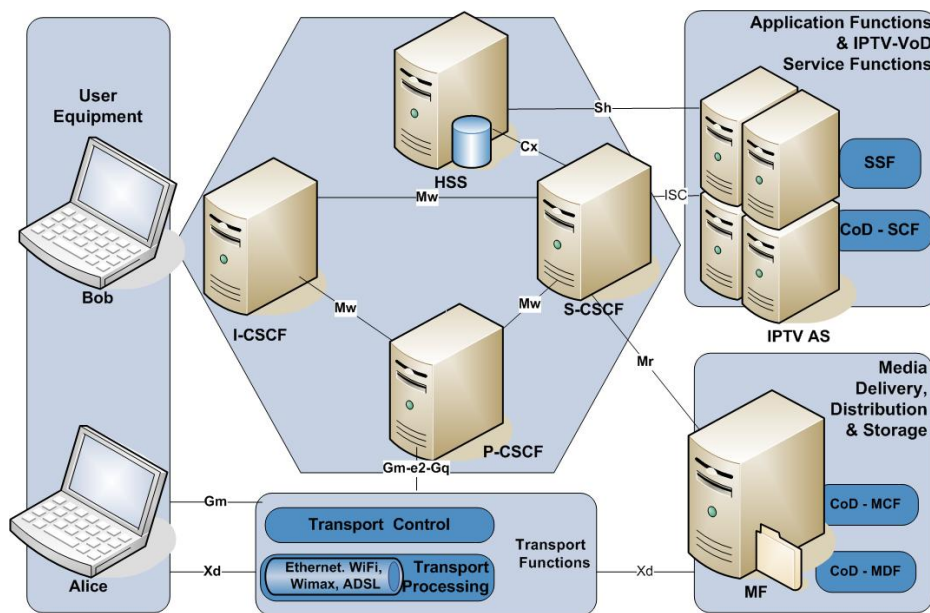


Figure 1.2: IMS Architecture [1]

Interrogating-CSCF (I-CSCF): queries the HSS to find the appropriate S-CSCF for the subscriber. It can also be used to hide operators' network topology from other networks.

Service Control Function (SCF): the SCF authenticates and authorises users to subscribe to the IPTV service and content. The SCF is a therefore functional entity that provides IPTV service logic and the functions required to support execution of such logic.

Service Selection Function (SSF): an SSF is a functional entity that provides service discovery and selection information to the UE.

Media Control Function (MCF): the MCF is a functional entity that provides the UE with functions required to control media flows and manages the Media Distribution Functions (MDFs) under its control.

Media Delivery Function (MDF): the MDF is a functional entity that delivers content data to the UE.

The IMS TISPAN *Draft* [1]⁵ standard defines the roles of the Call Session Control Functions (CSCFs), Service Control Function (SCF) and the Media Functions (MDF) that deal with the integration of the service management and service interactivity.

The IMS presents the most promising platform for IPTV as it allows high (service level) control of IPTV services; offers user-costumed charging and advertising platforms; seamless integration between services (e.g. telepresence over TV); and network mobility between different access technologies.

The IMS also allows for the easy provisioning of the following services:

- Push To Talk over Cellular (PTToC)
- Presence services
- MMSC - Multimedia Session Continuity

⁵This draft was released in August 2009

- IPTV
- Many other IP-based services

IMS-based Video and Advantages

There are many Internet based video services such as Youtube, Joost TV and television channel websites such as HBO which offer video content that subscribers can access when they want [16, 17, 11]. But VoD service provisioning over the IMS is particularly attractive as the IMS provides a framework for the enhancement of current video services such as the rental video service, live television and Web television.

In order to use a rental video service, a subscription account is normally first created at a local video shop. The rental video shop collects or buys DVDs of various movies and television programmes. These are then rented out to subscribers of the rental service. The main benefit of the video rental service is that video rental prices are much lower than DVD purchase prices. A rental video service however has the following limitations:

- Long video delivery time i.e. Most DVDs arrive several weeks after being ordered from overseas
- Limited content selection
- The length of the video rental lease is generally a day
- The video can only be played on DVD players or computers with DVD readers
- There are often geographical restrictions on where certain DVDs can be played

For fair charging, VoD over IMS provides reasonable charging using IMS entities such as Offline Charging System (OCS) and Call Charging Functions (CCF) [18]. The IMS also supports numerous fixed and mobile access networks. These networks can be accessed from a large variety of devices with different functional capabilities. An IP-based video service analogous to video rentals is the download option for retrieving video on demand over the Internet [19].

Live TV content offered over satellite and cable is often of a high signal quality [20]. The main limitation of live TV is that content is streamed indiscriminately to users who subscribe to the service. Users have limited content selection from broadcasted channels. The IMS can alternatively offer unicast video. This is achieved by videos targeted at individual users due to the personalised framework offered by the IMS that requires users to register on the Home Subscriber Server (HSS). The HSS keeps track of the user profiles and personalised service information that the service provider network stores and activates for every service request [21].

Web TV has also been widely used over the web by websites such as Youtube and Google Videos [22]. These websites offer various premium and user generated content online. Web TV, being an online service, provides the advantage of allowing users to interact with other users by the use of related Internet forums and chat rooms while streaming video. The limitation of Web TV is the relatively low signal and content quality of the videos streamed to various users [23]. To address Web TV limitations, VoD over IMS is designed with service and content quality control mechanisms that ensure an enhanced service experience (i.e. Quality of Experience - QoE) while also delivering content with the highest Quality of Service (QoS).

1.2.2 IMS-based Video Interactivity

A la carte menu principles are enjoyed by many people who like variety of choice in products. Accessing video a la carte is no exception. VoD, as already introduced, allows users to watch content of interest anytime, anywhere and over any network.

Interactive TV is concerned with providing a plethora of interactive and conversational service features by integrating various services and interactive functions with the provisioned video content. Interactive functions that are concerned with controlling the video content are given a particular focus in this study. These functions are hereby referred to as video interactivity functions. Video interactivity concerns a subset of interactive IPTV, which is direct interaction and manipulation of video streams.

To provide the user with a fully interactive TV experience, a video on demand service supports the following operations:

1. Basic VCR operations: *play/resume, stop, pause*
2. Advanced stream control operations: *jump forward/fast forward, jump backward, speed up, slow down, reverse, fast reverse, and slow reverse* [24]
3. *Video adaptation* during a video session (e.g. swapping the device or the access network during a session)

These interactive features are required over the IMS platform. There are various video interactivity protocols that can be employed to achieve this. Protocols such as the Session Initiation Protocol (SIP) and Hypertext Transfer Protocol (HTTP) are used for service and session functions such as starting a video chat (e.g. SIP), or receiving data with the media content. Other protocols are concerned with controlling the media content (e.g. Real Time Streaming Protocol (RTSP)). It is not clear, however, which protocol would be more suitable for implementing these functions in order to achieve a full interactive experience.

Figure 1.3 provides a simple overview of the signalling and media planes of a video server network. The signalling plane involves the control of service and media operations through protocols such as SIP, HTTP and RTSP. The media plane primarily involves the processing and transmission of media streams. The video interactivity requests sent on the signalling plane of a video server network (i.e. MCF), are forwarded to the media plane (i.e. MDF) for execution.

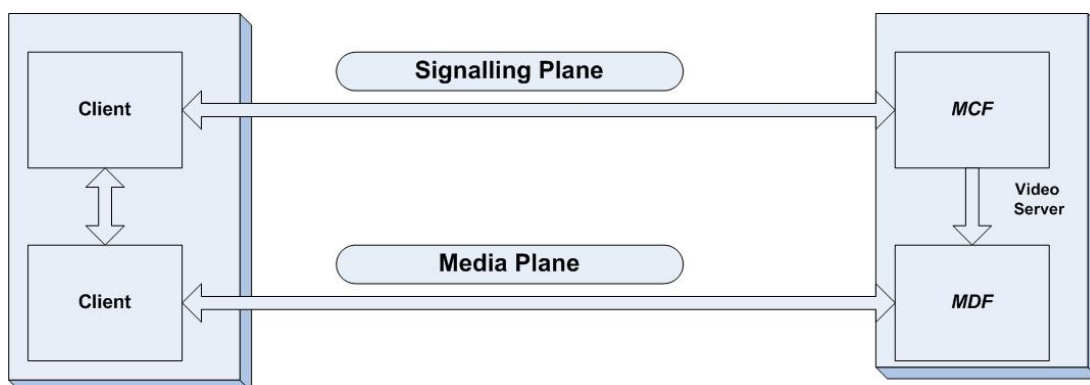


Figure 1.3: Signalling and Media Processing

In this study, the term 'video interactivity latency' includes components of both signalling and video processing. Any request that is sent to the video server, such as 'pause', is transported

through a protocol such as RTSP on the signalling plane. This request is received at the video server and then processed (i.e. the video stream is paused), and whilst a signalling reply is sent back to the client, the video client will also reflect the video stream being paused. Latency will be reflected by the time between when the user sends a request and when the request is executed on the client video player. Unless clearly stated otherwise, latency, l_{Total} , in this study therefore refers to:

$$l_T = l_{signalling} + l_{videoServerProcessing} \quad (1.1)$$

In this equation, $l_{signalling}$ indicates the signalling latency component and $l_{videoServerProcessing}$ indicates the processing latency component. This equation therefore assumes that the signalling reply is only sent after the video stream processing has been completed (i.e. the video server pauses the video stream before sending a '200 OK' reply to the client). In some of the tests carried out in this study, the $l_{videoServerProcessing}$ component will be excluded in order to exclusively determine the latency due to the signalling.

Video server processing load, in this study, refers to components of both signalling and media processing load on the video server⁶ from all requests received at the video server. Unless clearly stated otherwise, the video processing load, $CPULoad_T$, in this study refers to:

$$CPULoad_T = CPULoad_{signalling} + CPULoad_{videoServerProcessing} \quad (1.2)$$

In this equation, $CPULoad_{signalling}$ indicates the processor load due to the processing of signalling messages and $CPULoad_{videoServerProcessing}$ indicates the processor load due to the processing of video streams for video interactivity (i.e. rewind, forward and video adaptation). In some of the tests carried out in this study, the $CPULoad_{videoServerProcessing}$ component will be removed in order to assess the exclusive load of signalling requests' processing on the video server. In other instances only the $CPULoad_{videoServerProcessing}$ is being referred to.

1.3 Problem Statement

In many on-demand video server architectures, VoD sessions are commonly controlled and managed by RTSP. The use of SIP for media session control has, however, been increasingly suggested in literature [2]. Several IETF media session control draft documents also recommend SIP for media session control [25, 26]. SIP has many characteristics that mimic and expand on the operations that RTSP executes.

Firstly, there is currently no thorough conclusive evaluation and analysis on the use of SIP for video interactivity in the literature. If the SIP protocol is to be widely adopted for controlling the media playback, significant delays for interactive requests (e.g. for a 'pause' request) would cause a degradation in the user service experience. RTSP is widely used for such controls and provides generally accepted video interactivity latencies. This makes it necessary to perform an investigation into whether SIP video interactivity latencies are comparable to that of RTSP. Given that the IMS standard accepts the use of both protocols, the use of SIP for session control and for media control presents the following research problem: *How can SIP be deployed over the IMS for media playback control with minimum signalling latency and ease of implementation comparable to the RTSP protocol?*

⁶For simplicity, both 'signalling' and 'media processing' components are presumed to co-exist on the same physical video server *entity*.

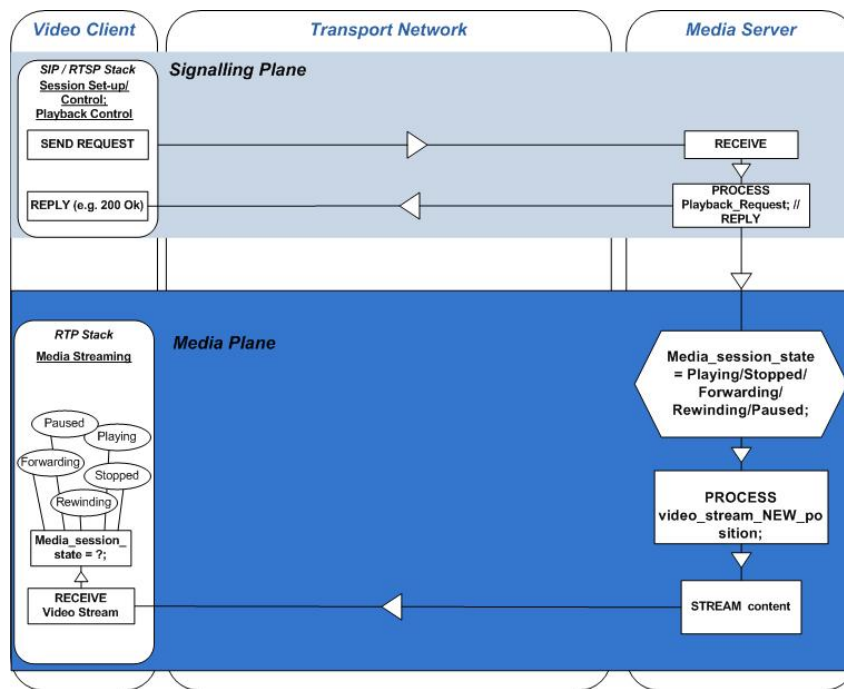


Figure 1.4: Signaling and Media Processing Diagram

Secondly, Figure 1.4 shows the signalling layer and a message route for each playback request. A playback request traverses the transport network to the media server, where it is received and processed (and a reply message is sent). If the playback request is accepted, the playback request is then forwarded to and executed at the media layer. This may involve re-packaging the video stream (in case of video adaptation) or changing the video stream position. The figure shows that for every video playback message sent across, the media layer will respond correspondingly. In this case, the video server processing load will increase as more requests are sent from the clients. The media system of a video server could be over-loaded by many video playback requests that require frequent media processing and adaptation. If this does occur, then the implication of the interactivity protocol on the scalability of the rest of the system needs to be assessed with the aim of alleviating a system over-load. This leads to the following research problem: *What is the relationship, if any, between a video interactivity protocol such as SIP (or RTSP) and the video processing load of a video server system? If such a relationship exists, which protocol design reduces the processing load of a video server system?* A video server system, in this case, refers to both the signalling and media components on the video server that are responsible for processing the received signalling requests as well as performing the corresponding media functions.

Lastly, protocols such RTSP and SIP need to provide an enabling framework, in their design, to support a greater variety of video interactivity requests. The RTSP protocol for example, only supports basic operations for pausing, stopping, forwarding, and rewinding a video stream. This limits any future development of service specific interactivity requests such as live video adaptation, thus presenting a need for the design of a protocol that provides a framework for the development of future interactivity features. *Which protocol design, that makes use of either the SIP or the RTSP protocol, can provide an adequate framework for the development of future interactivity requests?*

1.4 Thesis Objectives

Many current video interactivity architectures either propose the integration of RTSP into the SIP-based IMS architecture or the exclusive use of SIP. Current SIP-based solutions have not been comparatively evaluated and assessed in light of the latencies experienced and the implementation complexity. The aim of this study is to provide an evaluation of SIP-based video interactivity. The author will propose a design framework to support SIP-based interactivity. The proposed design will aim to minimise SIP-based video interactivity latency (as defined in Equation 1.1).

To assess the performance of SIP over the IMS, an evaluation platform will be implemented. The performance evaluation will provide an analysis of the latencies incurred by both RTSP and SIP protocols on playback requests. Different access networks will be used in the evaluations in order to observe the impact of different access networks on each playback request latency for both protocols.

There are many video scalability architectures in the literature [27, 28, 29]. This study will provide a discussion on Internet-based streaming architectures. It must be noted that these architectures are not the primary focus of this study. These are only discussed here to assess their impact on both video server load and video interactivity features. The discussion on various Internet-based streaming techniques (that aim primarily to reduce the amount of bandwidth used) is provided here due to the lack of focused research on the relationship between video interactivity and the video server processing load (as defined in Equation 1.2). The aim of this discussion will be to examine whether some video server designs have a specific impact on video interactivity (e.g. decreasing interactive features). The aim of the video interactivity design suggested in this study is to allow for full interactivity features while minimising the impact on the video server processing load.

This study proposes that the nature of the relationship between interactivity and video server load is such that: *if more interactivity requests lead to more processing load on the video server, these requests (for video playback) need to be rationed⁷ in order to avoid server over-loading.*

This hypothesis will be tested using the evaluation platform implemented in this study. It will evaluate the impact a video interactivity protocol has on the total video server processing load (as defined in Equation 1.2). The video processing load of video interactivity requests will be measured. Furthermore, the processing load will be measured with respect to how many interactivity requests are enabled.

This study will provide a discussion on the role, importance and relevance of interactivity in IPTV services in the literature. A discussion will then be provided on current video interactivity designs that aim to provide a framework for the future development of video interactivity requests. This study will also provide an enhanced SIP-based video interactivity design that aims to enable seamless development of interactivity requests.

1.5 Scope and Limitations

There are many signalling mechanisms that suggest the integration of SIP, RTSP and even HTTP. These vary from Voice over IP (VoIP) systems that make use of both RTSP and SIP protocols (for voicemail messaging), to IMS-based architectures that primarily aim to harmonise the use of SIP and RTSP for easier service creation and session management. This study however only primarily

⁷An enhanced service profile architecture is proposed that will allow or limit certain requests depending on the user subscription profile.

focuses on the video interactivity (i.e. video playback control) component of the use of both protocols. Given that this study is done in the IMS context, some issues related to the integration of the two protocols will be considered, but they are discussed merely for comprehensiveness and do not constitute a major aspect of this study.

In this study, the scalability of video on demand systems is assessed primarily in terms of the processing capacity. The processing capacity was assessed on the video server. Although video system scalability (with regards to both bandwidth scalability and video server processing load) will be discussed in this study, this will only be addressed in relation to the possible impact on video interactivity design. Video server processing load, in particular, is examined primarily to establish the nature of the relationship between video interactivity requests and the overall video server system design.

Due to the time constraints of this study, video interactivity functions examined in this study are limited to well-known trick-play modes (i.e. PAUSE, PLAY, REWIND, FORWARD).

There are many other existing IPTV platforms that provide enhanced interactivity video services. This study only focuses on video interactivity functions over the IMS architecture. Evaluations will be carried out to analyse the performance of both protocols mainly over the IMS architecture. This was decided on two factors. Firstly, as already argued in Section 1.2, the IMS provides the most advantageous platform for the deployment of IPTV. Secondly, the current research and development on the Fraunhofer Fokus Open IMS test-bed and on the UCT IMS Client at the UCT Communications Research Group (CRG) provided a reliable platform for the evaluations carried out in this thesis.

In assessing the performance of SIP over the IMS, only video playback latencies will be considered. The access networks used for evaluating playback latencies are WiMAX and the Wired LAN (Ethernet).

1.6 Thesis Contributions

The current UCT IMS client was extended to support video interactivity modes. An extension of the SIP stack was developed (using eXoSIP modules) to support SIP-based interactivity. An RTSP stack was implemented (using C modules) to support RTSP-based interactivity. The UCT IMS IPTv server [30, 31] was also further developed to support VoD stream requests, streaming and interactivity processing from both RTSP and SIP. The software implementation contributions can be found in Appendix B as well as in the accompanying CD-ROM.

The following papers were published by the author:

1. Lesang V. Dikgole and Neco Ventura, "Video on Demand Service for Next Generation Networks," Proceedings of South Africa Telecommunication Networks and Applications Conference (SATNAC), Wild Coast Sun, South Africa, 6-10 September 2008.
2. Lesang Dikgole, Neco Ventura, "An Advanced Video Interactivity Framework for IPTV Services," Southern African Telecommunication Networks and Applications Conference (SATNAC 2009), Mbabane, Swaziland, August 2009.
3. Keoikantse O. A. Marungwana, Lesang V. Dikgole, Neco Ventura, "An Efficient Community-centric IPTV Deployment Model for Developing Regions," International Conference on Ultra Modern Telecommunications (ICUMT 2009), St. Petersburg, Russia, October 2009.

1.7 Thesis Outline

Chapter 2: In this chapter, a critical review of current research and standardised interactivity architectures and various scalability solutions for video interactivity is provided. This chapter provides a discussion on current RTSP and SIP implementations, highlighting their advantages and disadvantages. This chapter will also discuss current standards and designs for video interactivity platforms for developing future video interactivity requests.

Chapter 3: In this chapter an IMS-based video interactivity framework that aims to integrate video interactivity into the functionality of SIP over the IMS is proposed. The IP Multimedia Subsystem is motivated as a choice platform for the implementation of the proposed three-layered architecture. The proposed framework aims to minimise SIP-based video interactivity latency and the impact of video playback requests on the video server processing load. The proposed framework also provides an enabling platform for future video interactivity requests.

Chapter 4: This chapter will discuss the evaluation platform. Fraunhofer Fokus Open IMS Core (FFOIMS) and the UCT IMS test-bed (UIMS) are used as part of the evaluation platform. All the main implemented components of the proposed framework are discussed.

Chapter 5: This chapter presents various experiments carried out to evaluate protocol latencies and processing overhead. The trick-play request latencies by both SIP and RTSP in the IMS are measured. The proposed Three Layered Video Interactivity Framework (TLVIF) is evaluated. The processing overhead is examined by assessing the processing (CPU) load of unicast streaming with or without video processing and also with selective video processing.

Chapter 6: This chapter presents the conclusions drawn from the study and from the experiments carried out from the results' chapter. Possible recommendations and future work from this study are also provided.

Chapter 2

Literature Review

This chapter will discuss current video interactivity solutions that use both RTSP and SIP protocols. This will be followed by a discussion on different scalability architectures and on the relationship between video interactivity and the video server load. The last discussion will provide an assessment of various requirements for future interactivity solutions. This will then be followed by an analysis of design frameworks on which future interactivity requests can be developed.

2.1 Video Interactivity Protocol Architectures

The literature discussed in this section relates to the first research question: *How can SIP be deployed over the IMS for media session control with minimum playback latency and ease of implementation in comparison to the RTSP protocol?*

The discussion below will focus on various literature that deals with the use of SIP and RTSP for video interactivity, video interactivity over the IMS and various SIP/RTSP implementations. A fuller description of the structure and functionality of both protocols is provided in Appendix B.

2.1.1 RTSP-based Video Interactivity

The TISPAN IMS-based IPTV standard suggests two methods for performing trick-play. The methods were initially proposed by Riede et al [32]. They provide one of the first known designs that integrate the use of both SIP and RTSP in the IMS architecture.

Figure 2.1 shows the signalling plane for the two methods. In both methods, the SIP protocol is used for session management and content authorisation. The RTSP protocol is used for performing media control commands such as PLAY, PAUSE and STOP.

The difference that exists between the two methods is that Method 1 uses RTSP only for trick play. Session control (i.e. set-up and teardown) is done by the SIP protocol, as shown in Figure 2.1. SIP methods used for session set-up and teardown are the INVITE and BYE methods respectively.

This method is similar to the one proposed by Singh et al [33] who make use of both protocols in the context of a voice mail system. The benefit of this method is that SIP-based session control is supported in the IMS architecture. This makes it easier to seamlessly monitor video sessions in the (SIP-based) IMS context. It also allows for possible inter-working of the video service with other IMS based services such as VoIP. An example of the benefit of inter-working services in the IMS is the function of automatically pausing a video stream when a user receives an urgent phone call.

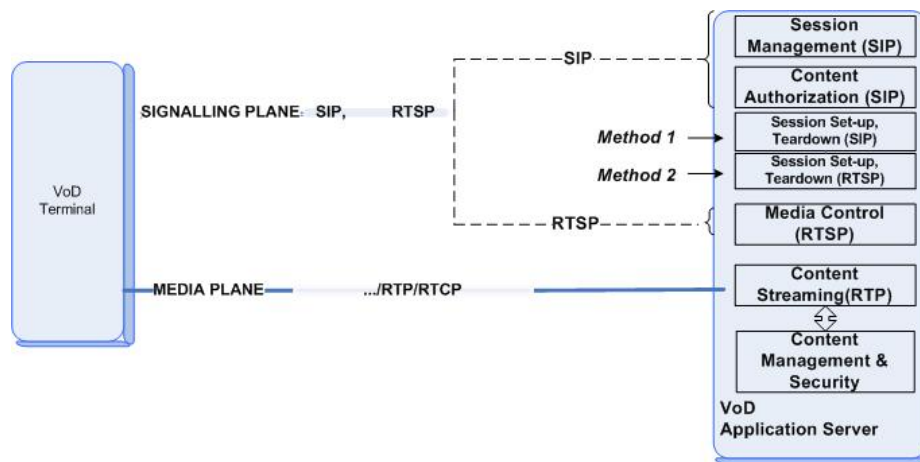


Figure 2.1: Hybrid RTSP/SIP based Media Control

The only possible drawback of this method is that the delays of setting up a video session may be large due to the bulky design of the SIP protocol as well as due to a number of IMS proxies that a SIP message traverses to reach the Media Control Function (MCF)¹.

Method 2 uses RTSP for setting up the video session and for controlling the media streams. Trick play commands are handled similar to Method 1. For session description and control, usual RTSP DESCRIBE, SET-UP and TERMINATE methods are used instead of using SIP (i.e. INVITE and BYE) methods.

The use of RTSP for video playback in the IMS context has been questioned in the past, including in an IETF draft [34]. The use of RTSP in the IMS has the following drawbacks:

- RTSP session control cannot be effectively integrated and monitored in the SIP based IMS architecture. The RTSP messages do not have to traverse the various security enabled and monitored entities and interfaces in the IMS architectures (such as the SCF and the MCF). This makes the proper management, monitoring and control of RTSP requests in the IMS platform a challenge.
- RTSP-based media session control essentially duplicates some of the session control functionality already provided by the SIP protocol in the IMS. A simplified and unified session control platform could be more elegantly achieved by the use of SIP for media session control.
- RTSP does not fully comply with the requirements set by the IETF MMUSIC internet draft [34] in that it cannot support Interactive Voice Response signalling.

These designs do not include the option of alternatively using SIP for media control as a unified mechanism for both session and playback control. The use of SIP for full media protocol, without the use of RTSP, needs to be investigated.

2.1.2 SIP-based Video Interactivity

SIP was not primarily designed for handling media control functions such as PAUSE, FORWARD and REWIND. SIP is however increasingly being recommended for video interactivity.

¹See Section 1.2.1 on the MCF.

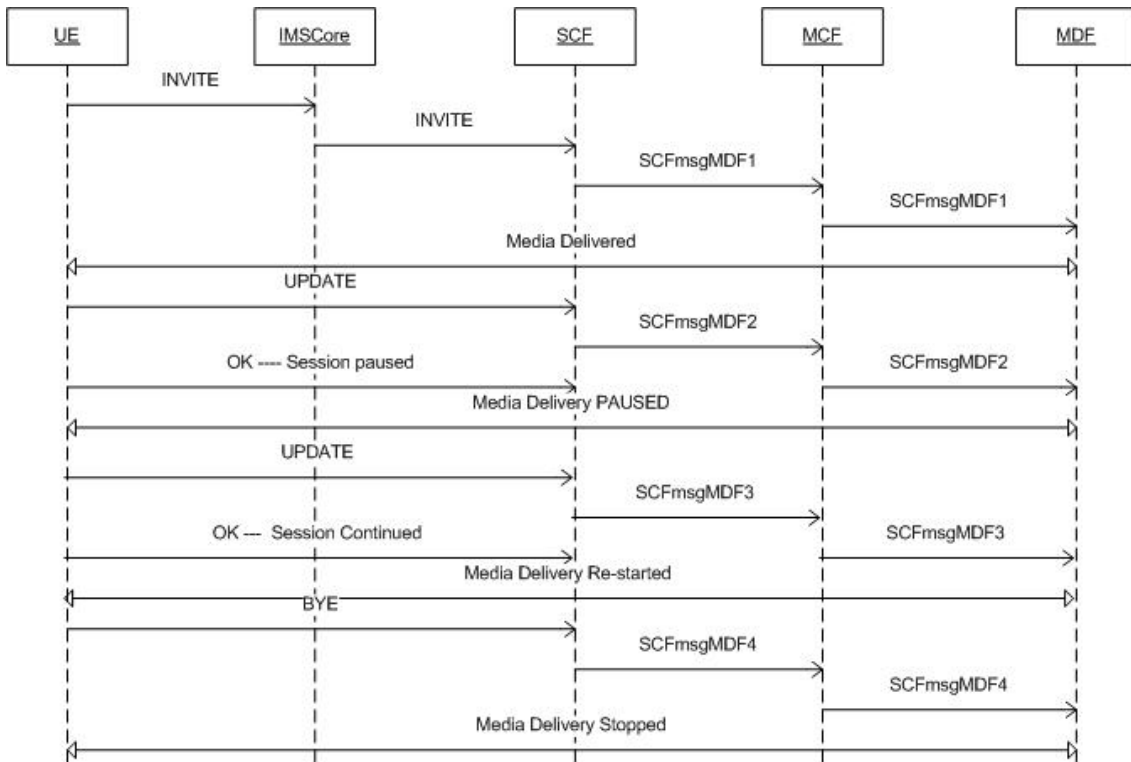


Figure 2.2: SIP-based Video Interactivity, Sivasothy et al [2]

Sivasothy et al [2] suggest the use of SIP as a unified session control protocol for IPTV services. This work has been incorporated into the IETF² SIP Working Group (WG) draft [35] in July 2009. They suggest the implementation of video interactivity requests using the SIP UPDATE method and the introduction of a new SIP-MEX header. They propose (as shown in Figure 2.2) that, to pause a video stream, a SIP UPDATE message will be sent to the Service Control Function (SCF)³ carrying a SIP-MEX header from the User Equipment (UE). The SCF then replies to the UE with a 200 OK SIP and the response body message (in XML). The SCF sends either a proprietary (shown in Figure 2.2 as SCFmsgMDFx) or a SIP message to the Media Control Function for the media to be paused.

SIP-based interactivity designs provide advantages of ease-of-integration into the IMS platform. The main challenge of such designs is reducing latencies that may be caused by traversing a number of SIP proxies and by the bulky design of the protocol. In this design, the UE can only directly talk to the SCF and not to the Media Function itself, such that the delay of the SIP-based interactivity request-response will be increased. By introducing the new SIP-MEX header line, the design also stipulates a modification of the SIP header. SCFs that do not support this SIP header will not be able to process (SIP-based) playback requests. The authors did not include the results of associated playback latencies on the test-bed implementation and evaluation of the proposed solution.

²The Internet Engineering Task Force (IETF) develops and promotes Internet standards through various working groups such as the SIP WG

³See Section 1.2.1 on the SCF.

2.1.3 Other IMS Video Interactivity Solutions

In an attempt to integrate the use of both protocols in the IMS, Khan et al [3] present an analysis of various frameworks that integrate SIP, RTSP and HTTP usage in the IMS framework. Their studies outline the limitations of parallel (or separate) protocol architectures. They point the disadvantages of redundancy; increased maintenance; the need for multiple registrations and authentications; and the complexity of deploying seamless applications. The authors then present two alternative architectures for IMS interactivity signalling: the extended IMS architecture and the enhanced IMS architecture.

The extended IMS architecture is shown in Figure 2.3. This architecture makes use of the SIP protocol for controlling and manipulating all service sessions. Most of the vendor (third-party) services are executed through protocols such as RTSP and HTTP. The extended IMS architecture does not allow users to directly interact with the IMS core through any protocol other than SIP. There is one central SIP-based proxy entity that manages all the generated sessions. For multimedia signalling, an Application Policy Function (APF) is introduced that translates SIP messages to RTSP and vice versa.

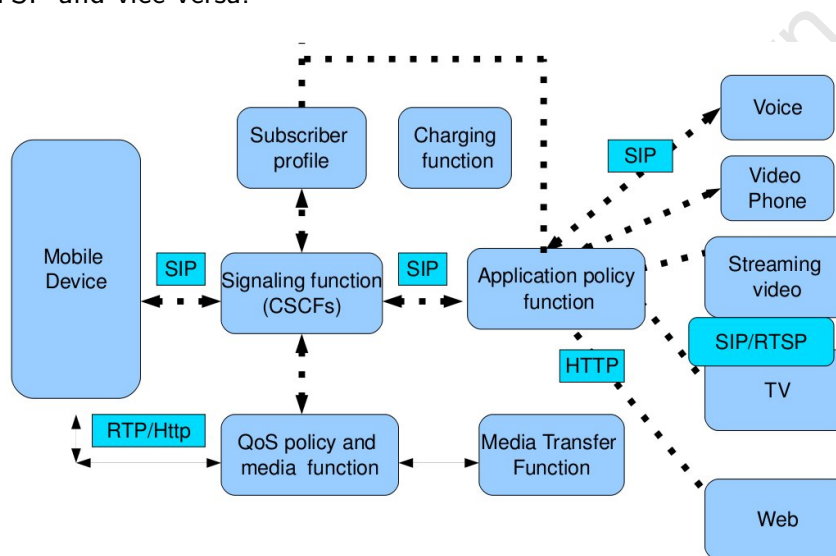


Figure 2.3: The Extended IMS Architecture[3]

This architecture offers an advantage in the unified application and use of the SIP protocol for all session and media control functions. A major limitation of the architecture is that it requires the addition of an extra component to the IMS IPTV architecture, which would add to the cost of development of the IMS IPTV system. The added intermediate component and the translation of SIP to other protocols may also cause unwanted delays in the signalling messages.

The enhanced IMS-based architecture proposed by Khan et al comprises a unified proxy architecture from current proxy architectures. Figure 2.4 shows the enhanced IMS architecture. The enhanced IMS-based architecture has essentially the same design as the extended IMS architecture. The main difference in the two architectures is that the enhanced architecture allows the use of all protocols (not just SIP) to access the APF. Many protocols such as RTSP, HTTP, and SIP have established proxy mechanisms that are deployed over the Internet. These mechanisms are used for authentication, authorisation, accounting and management of multimedia sessions. In the enhanced IMS architecture, the APF acts as a central proxy that synchronises all RTSP, HTTP and SIP information. The APF acts as a unified proxy solution that harmonises all incoming interactive requests.

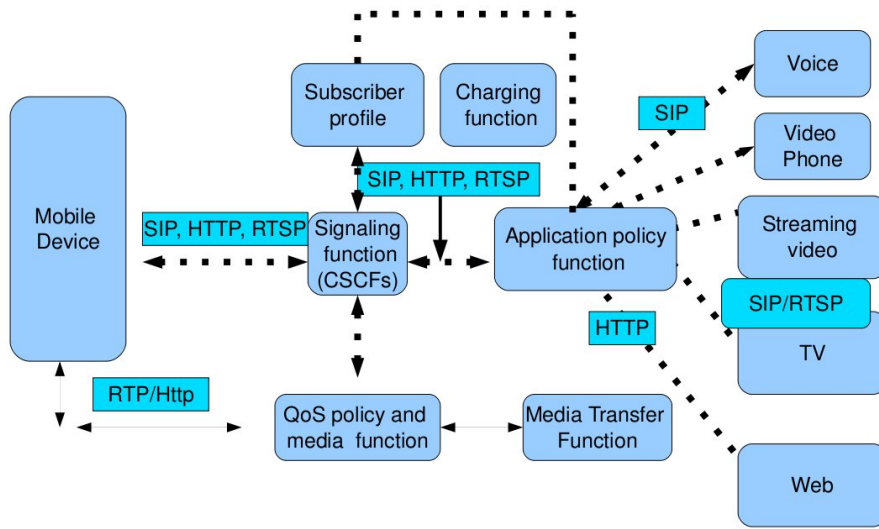


Figure 2.4: The Enhanced IMS Architecture[3]

This architecture directly employs the use of RTSP for performing trick-play. It represents an attempt to solve the problems associated with both parallel protocols architectures and complex protocol-to-protocol translation systems that cause delays. A strong advantage of this architecture is that it allows for unified proxy management of RTSP, SIP and HTTP protocols through the use of the APF. The authors do not present any details on the actual design and implementation of the APF. The APF, serving as a proxy, between incoming SIP and RTSP requests and the (signalling and media) servers may prove to be a largely complex design. It is unclear which of the protocols used will be in charge of session authorisation, control and description. As a result, there may be conflicting functional roles between the protocols.

The authors do not present the implementation and evaluation of both architectures. They do not directly deal with the impact of integrating various protocols on improving the user video interactivity experience, interactive latency and system scalability.

From the literature above, it is clear that the RTSP protocol is a widely used protocol for video interactivity. Due to the SIP-based protocol design of the IMS, the introduction of RTSP into the IMS introduces various challenges. The two TISPAN methods that suggest the incorporation of both protocols do not adequately address the need for a full user interactivity experience that is easily manageable on the (SIP-based) IMS network. SIP is primarily a session control protocol not originally designed for interactivity with media content [25]. There are various studies and implementations that have been discussed that use SIP for video interactivity. The adoption of SIP for video interactivity is encouraged by the flexibility of the protocol. Because SIP is already widely used for controlling VoIP, IM, audio and video conferencing, the use of SIP for video interactivity could provide a framework for a unified protocol design that encourages the proliferation of new integrated services with various interactive features. From the examination of the bulkiness of SIP protocol messages, the SIP design does not lend itself to being more efficient or providing lower latency than RTSP. Most of the SIP-based interactivity implementations discussed do not assess or directly attempt to minimise SIP video interactivity latency. This study aims to address this.

An associated research problem in video protocol architecture design is how scalable, on the video server system, the protocol signalling platform is. This will be discussed in the next section.

2.2 Scalability in Video Interactivity Protocol Architectures

A complete video server architecture typically processes signalling and media plane functions. This section discusses the impact of media playback requests (from the signalling plane) on the server processing load in the media plane. The literature discussed in this section relates to the second research question:

What is the relationship, if any, between a video interactivity protocol such as SIP (or RTSP) and the video processing load of a video server system?

2.2.1 SIP Signalling impact on Proxy Server Scalability

Several performance evaluations of SIP and RTSP that assess the impact of the protocols on servers have been carried out. Nahum et al [36] present an evaluation of SIP proxy server performance over the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) protocols. This evaluation does not specifically examine the impact of SIP video interactivity on server performance, but it provides insight regarding the impact of authentication and transport (i.e. TCP and UDP) protocols on server performance. The evaluations are carried out with and without authentication requests on proxy servers. The authors make use of Open SIP Express Router (OpenSER) proxy servers implemented on the Redhat Linux platform to perform their evaluations. A SIPp SIP requests generator is used for sending requests over a Gigabit Ethernet LAN. The evaluations show that SIP authentication reduces the proxy server processing performance by a factor of four (i.e. increases the processing load of the proxy server by the same factor). Another result from the evaluations is that TCP reduces processing performance by 65 percent (without authentication). It is clear from these evaluations that both authentications and the TCP protocol decrease the performance of SIP proxy servers. The use of authentication requests and the TCP protocol is however unavoidable in the context of Session Control Function (SCF) and the Media Control Function (MCF) used in the IMS video architecture. All video sessions need to be authorised (through user authentication), and the TCP protocol is stipulated for media playback control protocols by the IETF (See Section 2.3). Furthermore, a study done by Ram et al [37] showed that the server design has more impact on the performance of SIP over UDP and TCP than the transport protocols themselves. Ram et al used the same OpenSER platform and, after performing some server TCP proxy server processing optimisations, showed that SIP performance over TCP can be made sufficiently comparable to that of over UDP. Ram et al did show that the performance of SIP over TCP can be significantly improved by using multi-threading techniques, but they did not conclusively prove that SIP over TCP can be made to perform better than over UDP.

These studies, while providing information on the different parameters (i.e. authentications and transport protocols) that influence the performance of SIP proxy servers, do not however provide a comprehensive analysis of how SIP requests impact on the processing of various media requests (i.e. media processing load). What this research study aims to achieve is to analyse of the relationship between SIP-based video interactivity requests and the media processing load⁴.

⁴It is assumed in this study that the proxy server handling SIP requests will co-exist with the video server on the same physical entity.

2.2.2 Interactivity Signalling impact on Video Server and Bandwidth Scalability

The aim of this discussion is to assess whether any link exists between how bandwidth scalability mechanisms deploy or scale video interactivity requests and the video processing load (as defined in Equation 1.2) experienced on the media servers.

Dey-Sircar et al [27] propose two schemes for scaling video interactivity requests. Both schemes aim to optimise the amount of bandwidth available to service interactive requests. Requests such as Fast-Forward and Rewind (FF/Rew) are assumed to require more bandwidth and therefore more processing. The authors argue that FF/Rew requests always involve an increase in the frame rate (normally 30 frames/second) and this automatically increases the amount of data transferred per second. This, it is argued, therefore results in an increase in video server processing and bandwidth. They propose two schemes for efficient bandwidth usage. The first scheme queues requests (i.e. Fast-Forward and Rewind) if the bandwidth demand is larger than the load. The second scheme reduces the frame resolution of the video when bandwidth is not available. The results of the simulation and analysis show that bandwidth is saved and more users are supported using both schemes. This research only addresses bandwidth scalability and not necessarily the reduction of the video server load. The significance of this study is that the authors attempt to highlight a link between bandwidth and video server scalability⁵ in the constant retrieval of video frames from the storage disks.

A VoD system may generally have high bandwidth scalability (labelled 'HL' in Figure 2.5) with no or very little interactivity. Many multicasting designs⁶ that make use of the Internet Group Management Protocol (IGMP) [28] to channel a video stream to many users fall into this area of the graph.

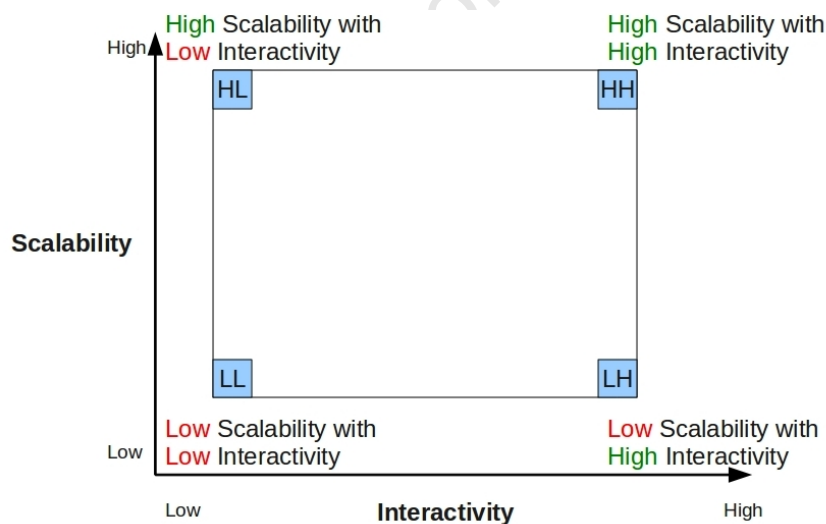


Figure 2.5: Scalability versus Interactivity

Split-and-merge [29] and the Unified VoD architecture [38] techniques are bandwidth scalability and multicasting designs that also aim to provide interactive features. They rely on the behavioural model of the Zipf distribution. According to the Zipf distribution model, many users are likely to

⁵'Scalability' in this sub-section only refers to 'bandwidth scalability'. 'Video server scalability' refers to the video processing load as defined in Equation 1.2.

⁶Multicasting refers to the process of streaming the same video to multiple users at the same time in order to reduce the processing load on the media servers serving multiple users.

access (or interact) with the same content. This model performs poorly when a greater variety of content is accessed by users. This results in low bandwidth scalability with high interactivity (labelled 'LH' on the graph). The other examples of such a VoD system are the download and content-caching techniques. The download technique is shown in Figure 2.6. It allows the video client select the content and to send a video request to the server. The video is then downloaded to the user. Interactivity is allowed after the full length video has been downloaded. The content-caching technique is shown in Figure 2.7. This technique exploits various video buffering/caching mechanisms by essentially downloading small streams from the server and creating the effective experience of 'live' streaming. Most of the content is buffered locally before play.

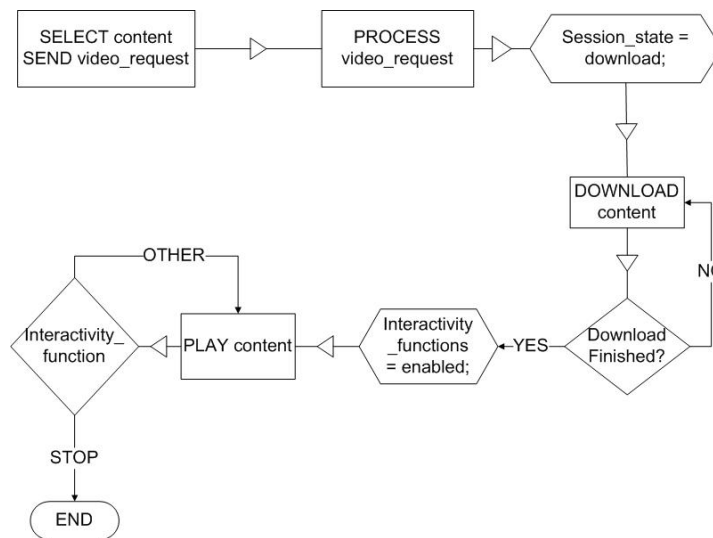


Figure 2.6: Flow Diagram - Download and Play

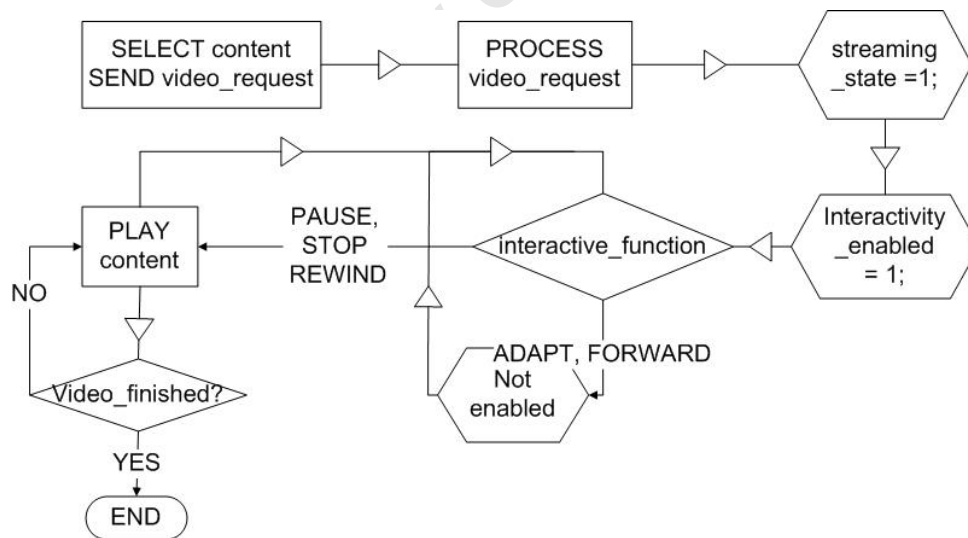


Figure 2.7: Flow Diagram - Content-caching

The main limitation of both of these techniques is that the user may only be interested in specific scenes. This can lead to bandwidth wastage. This often happens when many video sessions are aborted. The wastage of bandwidth when using such techniques was demonstrated in the research by Guo et al [39]. With regard to the video server load, the download and content-caching techniques allow for the lowest direct interactivity demand per session on the video server.

These techniques therefore achieve the reduction of the video server load by reducing interactivity demands directly processed on the video server.

If video interactivity has no or little impact on bandwidth scalability, then the system performance will be close to the area labelled 'HH' on the graph. The technique proposed by Dey-Sircar, achieves a performance close to this with full interactivity. The evaluation of the technique shows savings on bandwidth even though it does not convincingly guarantee any savings on the video server processing demands.

It is clear from the multicasting techniques discussed here, the Internet scalability platforms discussed in this section and the research by Guo et al that there exists a trade-off between bandwidth scalability and video server scalability. For bandwidth to be saved, multicasting split-and-merge techniques are effective. For providing scalability to the video server, download and content-caching techniques are effective. Multicasting techniques limit interactivity with the content, and Internet video architectures limit interactivity with the video server (media retrieval is the only direct interaction with the video server). For both techniques, however (with the exception of the advanced video server design by Dey-Sircar), interactivity has to be reduced to achieve either bandwidth or video server scalability.

Even though Internet based video architectures reduce direct interactivity with the video server, the measurable impact of direct interactivity on video server performance is not clear. One of the factors that make this hard to investigate is that most modern video scalability designs focus exclusively on bandwidth scalability than on video server scalability. It is therefore still not clear at this stage whether increased interactivity (from the signalling requests sent to the video server) directly results in increased video server processing load. It is this study's aim to investigate this further.

An associated research problem in video control protocol design is how the protocol signalling platform allows for the development of future interactivity requests. This is discussed in the following section.

2.3 Platform for Future Video Interactivity Development

The literature discussed in this section relates to the third research question: *Which protocol design, that makes use of either the SIP or the RTSP protocol, can provide an adequate framework for future development of video interactivity requests?*

In order to reduce future development costs, current video interactivity protocol designs need to incorporate scope for future development of video interactivity requests. This section discusses the literature on future interactivity and various designs that aim to provide scope for the development of future interactivity requests. Various requirements and stipulations from ITU-T and IETF standards on video interactivity platforms will also be examined.

Interactive VoD systems have been an active research topic since the early 1990's. The early systems primarily focused on the design and implementation of VCR trick modes over VoD architectures. Little et al [40] discuss various technological features and considerations for a truly interactive VoD service (i.e. interactive video games, interactive news telecast, distance learning television and interactive advertising). The study identifies generic requirements of resource reservation, load balancing, and QoS enforcement for supporting these interactive features.

Amongst various theories (outlined by Bouwman et al [41] and Altgeld et al [42]) that suggest different drivers for IPTV services, one driver for IPTV that is prevalent in the literature is the interactivity feature of IPTV. Shin [43] performed a quantitative study on 452 prospective IPTV users' expectations (i.e. different age groups, and technical competence levels), used the Technology Acceptance Model (TAM) [44] and a method of logistic regression [45] to evaluate the results of the surveys. The research concluded that, among other things, a new paradigm of interactivity is expected in IPTV services. The research observes that the demand for interactivity in future IPTV services does not necessarily mean more interactivity but primarily indicates the need for a differently customised, user-friendly and more enhanced interactivity experience (compared to the traditional TV experience). The new paradigm of interactivity expected in IPTV is linked to the desire of users to have more service control and communication possibilities that are often not possible in traditional TV services but are commonly experienced over the Internet. From this research, it is clear that new interactivity features would include features such as: content selection features, content manipulation abilities as well as other IPTV integrated and interactive experiences (e.g. receiving a voice call over a TV screen).

In light of this, protocol architectures are required that support the implementation of these interactive features. These architectures also need to provide a framework for the development of future interactive features. In the following subsections, requirements from various standards for the design of interactivity architectures will be discussed. This will then be followed by a look at various designs that attempt to provide scope for future requests.

2.3.1 Evaluating Quality of Experience for Video Interactivity

Future interactivity requests need to guarantee a certain level of Quality of Experience (QoE). QoE is a measurement of the service quality level as experienced by the user. QoE is therefore different from Quality of Service (QoS) as the latter puts more emphasis on the network performance than on the user experience. Kilkki [46] defines QoE as a concept that analyses the ecosystem of telecommunications' network quality assurance, business models and user behaviour models. The QoE metric can be used as a benchmark test for video interactivity solutions. QoE metrics are required to measure the level to which a service will likely be accepted or rejected by prospective users.

Pereira[47] suggests three levels of measuring the user quality of experience:

- Service degradation experience e.g. latency
- Service features
- Service quality e.g. perceptual video quality

This presents a qualitative overview of measuring QoE. It means that the parameters of network performance (i.e. latency), service functionality and the user subjective experience play a role in achieving QoE. In future video interactivity architectures, it is therefore important that both latency and providing more functionality as required and preferred by the user are adequately addressed. ITU-T G Series has a recommendation standard for measuring QoE. The ITU-T standard defines QoE as "the overall acceptability of an application or service, as perceived subjectively by the end-user" [48]. The standard specifically focuses on the definition of user requirements for QoE in IPTV services. The QoE requirements are defined from the user perspective and are agnostic to network deployment architectures and transport protocols. The standard provides QoE requirements for video, audio, text, graphics, control functions and meta-data. On IPTV content control functions,

the standard specifies that trick modes *should be guaranteed* in a VoD service. The ITU-T standard does not provide exact values for measuring the service experience and latency. It is clear that these variables will ultimately depend on the system design and the user requirements, trick latency is simply defined as the time between when the interactivity request is sent and when it is visibly executed on the video screen. According to the standard, trick latency is required to be *sufficiently low* to meet the user's requirements.

From the definitions detailed above, it can be observed that QoE comprises an evaluation of the quality of a service that includes the subjective user service experience. When QoE is applied to video interactivity, this would mean that various interactive features and the latencies associated with each interactivity request will be the most important variable for ensuring high QoE scores.

QoE is a relatively new concept that still requires further research. The QoE definitions discussed above will be used as a guideline for assessing the QoE of video interactivity designs discussed in this study.

2.3.2 Video Interactivity Requirements

The IETF has developed a set of guidelines for the development of future video interactivity protocol architectures. The IETF draft on Media Control Requirements lists the following requirements [26] [34]:

1. REQ-MCP-01 - The control protocol will enable one or more Application Servers to control a media server.
2. REQ-MCP-02 - The protocol must be independent from the transport protocol.
3. REQ-MCP-03 - The protocol must use a reliable transport protocol.
4. REQ-MCP-04 - The application scope of the protocol shall include Enhanced Conferencing Control and Interactive Voice Response.
5. REQ-MCP-05 - The protocol will utilise an XML markup language.
6. REQ-MCP-06 - A Media Server should be application/service independent. It should be possible to have a many-to-many relationship between Application Servers and Media Servers that use this protocol.
7. REQ-MCP-07 - Media types that are supported in the context of the applications shall include audio, tones, text and video.
8. REQ-MCP-08 - The protocol should allow, but must not require, a media server resource broker or intermediate proxy to exist between the Application Server and Media Server.
9. REQ-MCP-09 - The solution must enable one control channel between an AS and MS, and shall allow for the support of multiple channels.

These requirements can be applied to the video interactivity protocols (SIP and RTSP) already discussed above. The first requirement (REQ-MCP-01) ensures that users accessing media content from different services and/or applications can be able to access content without the need to subscribe to a particular VoD service. The second requirement (REQ-MCP-02) ensures that the protocol used is not vertically integrated with the media format or the transport protocols. This

allows for interactivity signalling processing to be managed and processed separately from media and transport processing. The third requirement (REQ-MCP-03) concerns the use of a reliable transport protocol such as TCP for ensuring high service reliability, which impacts on the service QoE. The fourth requirement (REQ-MCP-04) concerns the use of a media control protocol for functions beyond simple trick-play. This ensures that a unified interactivity system can be achieved within a singular protocol framework that can handle a variety of interactive features. The fifth requirement (REQ-MCP-05) suggests the capability of transporting XML description content in order to enhance the functional capability of the protocol. The sixth requirement (REQ-MCP-06) stipulates that the media server handling the requests from the user should not be statically linked to a particular service. This is to allow for enhancements to the media content and the description of the content that will not require modifications on the application servers. The seventh requirement (REQ-MCP-07) stipulates the support of all media types by the media control protocol. The eighth requirements (REQ-MCP-08) suggests the introduction of a proxy between a media server and the application server. This is to allow for the monitoring and management of media requests from the application server to the media server. The ninth requirement (REQ-MCP-09) stipulates the singularity of a control channel while encouraging the multiplicity of data channels between the media server and the application server.

These requirements provide a framework for the design of an interactivity architecture that allows for future interactivity enhancements and developments. The interactivity architectures assessed in this study will use these requirements as a rule of thumb for determining their effectiveness.

The following subsection assesses video interactivity designs that attempt to provide a framework for future interactivity development.

2.3.3 Current Video Interactivity Designs

TISPAN Method 1 and 2 (discussed in Section 2.1.1), make use of the RTSP protocol for all video interactivity operations (i.e trick-play requests). Given that the RTSP protocol currently only supports a limited set of interactivity commands, these methods do not provide the most effective platform for future video interactivity development.

Sivasothy et al (discussed in Section 2.1.2) make use of the SIP protocol for handling interactivity requests. This study does not provide a discussion on how the proposed SIP-based architecture could be used to provide an enabling platform for the development of interactivity features.

Khan et al (discussed in Section 2.1.3) do promote the use of RTSP, SIP and HTTP protocols. This provides larger scope for the development of future interactive requests. The limitation of these architectures, as already discussed in Section 2.2, is that the management of multiple profiles may prove to be cumbersome and security vulnerable.

Muntean et al [49] present one of the most recent designs for interactivity in modern IPTV systems. They propose the *iPersonal* framework for personalised IPTV entertainment that includes video adaptation modes and an adaptation framework that enforces QoE constraints on various interactive requests. The *iPersonal* IPTV server includes a Viewer Profiler (VP) that handles viewers' interests, a Concept Model (CM) that hierarchically organises media content in different categories, a QoE Model (QoEM) that makes personalised suggestions related to the media characteristics, and an Adaptation Engine (AE) that performs media adaptation. The *iPersonal* IPTV framework aims to improve on current designs that exclusively focus on Electronic Program Guide (EPG) design, or on user context-aware interactive content without incorporating network conditions and demands. The model of integrating interactivity features with QoE and network

conditions, proposed by the *iPersonal* design, is necessary for future IPTV interactivity designs. It provides a unified framework for providing interactivity needs while also ensuring that the network is not overloaded with increased demands from individual users. A full discussion on the implementation and evaluation was however lacking in the paper. A closer assessment of the design shows that it involves a number of processing entities (i.e. CM, EPG and AE), and it is not clear how these different entities will be integrated to provide an effective enabling platform for the proliferation of new interactivity requests. The underlying video control (signalling) mechanisms were not discussed. The authors also did not highlight possible implications of the proposed design on latency for each interactivity request.

Other possible future interactivity designs could include the use of protocol description formats. SIP and RTSP can be used alongside protocol description formats such as the Session Description Protocol (SDP) [50], the Extensible Markup Language (XML) [51], and the MPEG-21 Digital Item codec [52]. Each of the description formats can be used to provide more detailed video data format information such as the video bit rate. An example of an XML based interactive implementation is provided by Simoes et al [53].

In this section, various requirements for future interactivity architectures have been discussed. It has been noted that ITU-T trick-play requirements do not stipulate exact performance values for measuring QoE. The IETF video interactivity guidelines provide a comprehensive and non-restrictive framework for designing a protocol architecture that provides scope for future interactivity features. The complex design framework proposed in the *iPersonal* framework needs to be further developed and implemented further in order to fully assess the viability of the framework.

2.4 Chapter Summary

The first aim of this study is to find a SIP design framework that results in an easier implementation to achieve lower video interactivity latencies (as defined in Equation 1.1). The literature review in this chapter aimed to discuss various literature that provide different designs and implementations of SIP and RTSP protocols for video interactivity. The use of SIP for video interactivity has been found to be a recent development that has not been fully investigated. Despite various designs that aim to integrate the use of both protocols in video on demand architectures, none of the designs in the current literature provide a comprehensive comparative evaluation of SIP and RTSP protocols. As a result none of the designs can be used to comparatively assess both protocol latency and complexity of implementation of SIP video interactivity.

The second aim of this study is to investigate the nature of the relationship between interactivity and scalability. The literature looked at various scalability mechanisms. The various investigated literature primarily focuses on bandwidth scalability techniques and lacks a conclusive analysis on the relationship between video interactivity and the video server load. This study will aim to investigate this further and propose a platform that aims to minimise the impact of video interactivity on the video server load.

The third aim of this study is to find the protocol framework that allows for future development on interactivity requests. The literature review discussed the *iPersonal* framework which is a design that incorporates the use of various component engines in order to enhance the interactivity experience. Underlying video control signalling mechanisms were not incorporated into the *iPersonal* framework design. A protocol framework that makes use of established video control protocols such as SIP or RTSP is required and will be investigated in this study.

Chapter 3

The IMS-based Three-layered Video Interactivity Framework

In the previous chapter, it was established that none of the current designs are suitable for SIP-based video interactivity. It was also found that existing frameworks that are built for extensibility, such as the *iPersonal* framework, are not fully integrated with established video control protocols such as SIP or RTSP to enable further developments on video interactivity. An extensible protocol framework that is integrated with RTSP or SIP is therefore required.

In light of this, this chapter aims to achieve the following:

- Propose a framework for the implementation of SIP-based interactivity. The proposed framework will aim to reduce the delay incurred when transmitting and processing a SIP-based video interactivity request. This delay is measured from sending a request to the video server to receiving a reply indicating that the request was processed. In addition to this, the proposed framework will also aim to provide a platform for the development of future video interactivity requests.
- Propose a hypothesis regarding the nature of the relationship between interactivity and the video server load. The hypothesis proposed in this chapter is: *If more interactivity requests lead to more processing load on the video server, these requests (for video playback) need to be rationed, through the use of user service profiles, in order to avoid server over-loading.* This hypothesis will be tested further using the evaluation platform implemented in the following chapter.

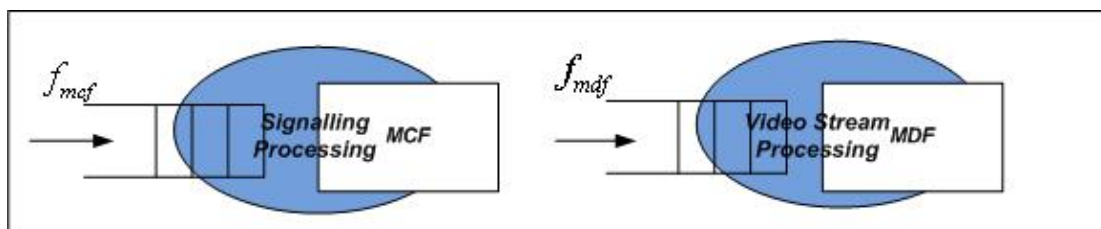


Figure 3.1: Interactivity and Video Server Load Overview

Figure 3.1 shows an overview of the video interactivity operations in an IMS video architecture. The f_{mcf} variable represents the frequency of interactivity signalling requests from the user to the video server, directed at the Media Control Function (MCF). The MCF processes a request and

forwards it to the MDF. The f_{mdf} variable represents the frequency of requests received at the Media Distribution Function (MDF).

The MDF performs the media processing as per instructions received from the MCF. The queues next to the MCF and the MDF are assumed to be proportional to the frequency of the requests and are shown to indicate the load of both entities. The aim of the design in this chapter is to reduce f_{mdf} , such that $f_{mdf} < f_{mcf}$, in order to reduce the processing load on the MDF with the ultimate goal of reducing the total video server load.

This Chapter is divided into two main sections. The first section will discuss the requirements of a video interactivity framework. These requirements will be guided by IETF and ITU-T standards discussed in the previous chapter. The second section will discuss the proposed Three Layered Video Interactivity Framework (TLVIF) that makes use of three composite layers for media processing, processing video interactivity and service management.

3.1 Requirements of the Video Interactivity Architecture

In this section, requirements of the video interactivity architecture are outlined.

3.1.1 Metrics for the Design

Video interactivity latency primarily refers to the latency from the transmission and processing of a request. Any request that is sent to the video server, such as 'pause', is transported through a protocol such as RTSP. This request is received at the video server and then processed (i.e. a video stream is paused), and whilst a signalling reply is sent back to the client, the video client will also reflect the video stream being paused. Latency is then a measure of the time between when the user sends a request and when the request is executed on the client video player. Latency therefore includes components of both transmission latency and processing latency as per Equation 1.1.

The video server processing load specifically entails the amount of processing load contributed to the media component (i.e. the MDF) of the server by the processing of video streams during video interactivity processing.

3.1.2 Design Requirements

In view of the literature discussed in the previous chapter, the video interactivity framework will meet the following criteria:

- Low latency. Although the ITU-T standard provides no accurate values, SIP video interactivity latency should at least be expected to be in a similar range to that of RTSP latency (typically less than 1 second). The SIP-based architecture proposed by Sivasothy et al [2] requires major modifications to the SIP stack and the authors do not present any results on the actual latencies experienced. Khan et al [3] primarily makes use of translation proxies that add delays. The proposed design should aim to provide reduced video interactivity latency.

- The platform should reduce the MDF processing load (defined in Equation 1.2), due to interactivity while allowing full interactivity features. Most video download techniques off-load the video interactivity processing load from the server to the client. The aim of the proposed design will be to allow live interactivity features at the video server while also reducing the video interactivity processing load (at the MDF).
- A video interactivity framework that supports a variety of video interactivity features to be developed should be incorporated into the design¹. The proposed design should aim to use one of the established protocols, RTSP or SIP to implement various interactive features. The iPersonal framework, which also makes use of the XML and is application independent, does not however provide any underlying transport protocol mechanisms for how these features will be implemented.

3.2 The Proposed Three Layered Interactivity Design

The proposed design essentially comprises:

- The exclusive use of the extensible SIP protocol for video interactivity
- A user profile framework in the MCF to reduce the video processing load at the MDF

The requirements of the proposed design as set out above are: low latency, a framework for the development of interactivity functions and increased capacity for handling video interactivity requests.

The TISPAN platform provides a standardised IMS IPTV platform that is widely recognised and adopted by industry and is widely used in IMS research as seen in Sivasothy et al [2], Cuevas et al [54], Al-Hezmi et al [55], Friedrich et al [56, 57], and Chatras et al [58].

Figure 3.2 shows the TISPAN IMS-VoD architecture².

UE: The IPTV enabled UE initiates the IPTV control and media signals, and displays the corresponding information to the user. The user interaction with the UE allows for selection of program, content, and service descriptions, such as content guides for broadcast and VoD services. This UE interface will also include video interactivity buttons such as PAUSE, PLAY, FORWARD and REWIND in order to control video streams.

IPTV Service Supporting Function (IPTV SSF): defines and enables common functions which could be supported or used by other IPTV services or applications. This is especially useful for the development of a common interactivity framework across all IPTV services.

Service Control Function (SCF): the SCF authenticates and authorises users to subscribe to the IPTV service and content. The SCF is a functional entity that provides IPTV service logic and the functions required to support execution of such logic.

Home Subscriber Server (HSS) and User Profiles (UP): user data that are involved in providing IPTV services. It provides functionality for authentication, authorisation, and signalling for

¹This relates to the IETF set of requirements, discussed in the previous chapter, that stipulate independence from the transport layer; the use of the protocols in applications such as Interactive Voice Response; the use of XML or SDP; application independence; support for a variety of media types; and a single video interactivity (playback control) channel

²It must be noted that this Figure only shows already existing components in the TISPAN standard, the new components which are part of the proposed solution are discussed from Section 3.2.1.

core, is used in the proposed solution.

3.2.1 A SIP-based Video Interactivity Architecture

The Method line in the Header of the SIP message can be used to implement a series of video interactivity methods as in the design by Sivasothy et al [2]. The advantage of this method is that of quick processing since the signalling server needs to only look at the header. The rest of the SIP message body does not need to be processed. The disadvantage of this method is that SIP-based video servers will need to be adapted to recognise and execute new SIP methods. In order to allow for the seamless introduction of new interactivity features, disruptions on the current SIP protocol design have to be minimised by avoiding changes on the SIP header.

The proposed design differs from the SIP-based video interactivity design proposed by Sivasothy et al [2] in that instead of a header line being used an attribute line, `a=siptrick:`, in the SDP body is used. The proposed design also differs from the Extended IMS Architecture proposed by Khan et al [3] in that it avoids the introduction of a translation proxy for effecting RTSP requests from a SIP message.

The proposed SIP-based architecture for IMS-based video interactivity is shown in Figure 3.3.

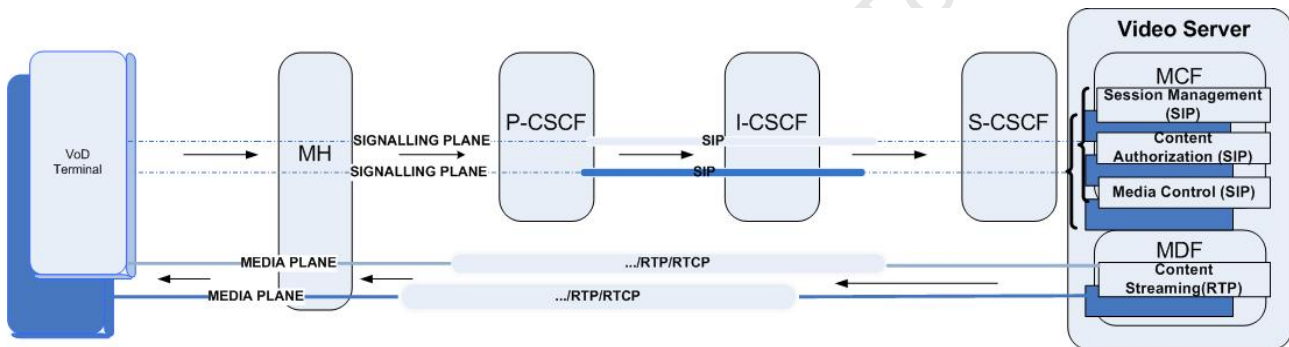


Figure 3.3: SIP-based Video Interactivity Framework

The SIP⁴/SDP attribute line for interactive functionality is executed at the MCF. The proposed SIP-based video interactivity (that directly interacts with the MCF through the IMS core) therefore primarily provides an improvement in simplicity (compared to the design by Sivasothy et al [2]) and a slight improvement in latency (compared to the design by Khan et al [3], which adds delays through the use of a translation proxy).

The IMS core network entities that will be traversed for the SIP signalling requests are shown. These entities include a mobile host station (MH), the Proxy Call Session Control Function (P-CSCF) server, the Interrogating CSCF (I-CSCF) server, and the Serving CSCF (S-CSCF) server. The media plane, which works independently from the signalling plane, uses the RTP/RTCP protocol for video transmission. The figure also shows the server functions performed at the video server (i.e. MCF + MDF). These functions include: session management, content authorisation and media control. The signalling functions will be contained in the MCF. The media processing and transmission functions will be in the MDF. The focus of the implementation of this design will be on the media control functions.

⁴SDP is used to implement the interactive features, but for readability 'SIP/SDP' is intentionality left out even though it is implied.

The actual value of the latencies experienced and the impact of the SIP-based video interactivity design will be evaluated in the following chapters. The SIP message body structure was modified as shown in Figure 3.4. The attribute line siptrick:pause is shown that is used for SIP-based interactivity requests. Similarly, SIP-based interactivity requests for rewind, forward, stop and play were implemented using siptrick:rewind; siptrick:forward; siptrick:stop; siptrick:play attribute lines respectively. The SIP/SDP details implementation are provided in Appendix B, Section B.3.2.

```
(SIP Header)
SIP/2.0 200 OK
Via: SIP/2.0/UDP
...
(SDP Body)
v=0
...
a=siptrick:pause
```

Figure 3.4: The Proposed Modified SIP Message

SIP interactivity messages traverse the IMS core as shown in Figure 3.5. A SIP 'pause' message is shown traversing the IMS core and then forwarded to the Media Control Function (MCF), where it is processed and a signal is sent to the MDF to pause a video stream.

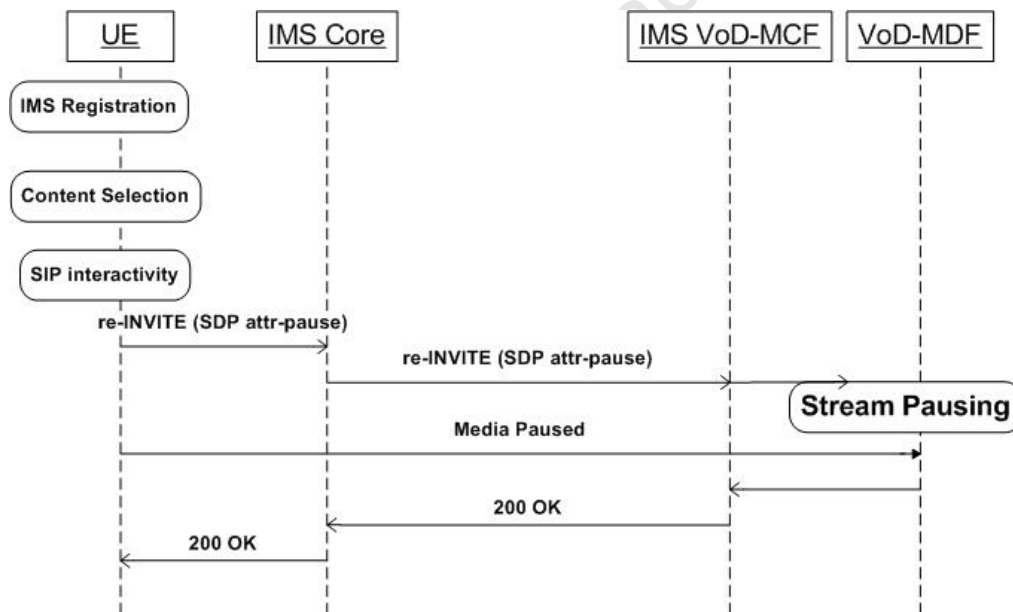


Figure 3.5: SIP-based Video Interactivity Sequence Diagram: Pause

Similarly, in Figure 3.6, a SIP 'rewind' message is shown that traverses the IMS core and then forwarded to the Media Control Function (MCF), where it is processed and a signal is sent to the MDF to rewind a video stream⁵. It must be noted that for each 'rewind' request sent by the client, a corresponding 'rewind' process will be initiated at the MDF. For each of the 'rewind' signals received at the MCF, the playback rate will therefore be increased exponentially by a multiple of four in the negative direction.

⁵The implemented algorithm in the 'rewind' implementation rewinded the video backwards at four times the playback rate. This continues until the user presses 'play' to replay the video at the normal playback rate.

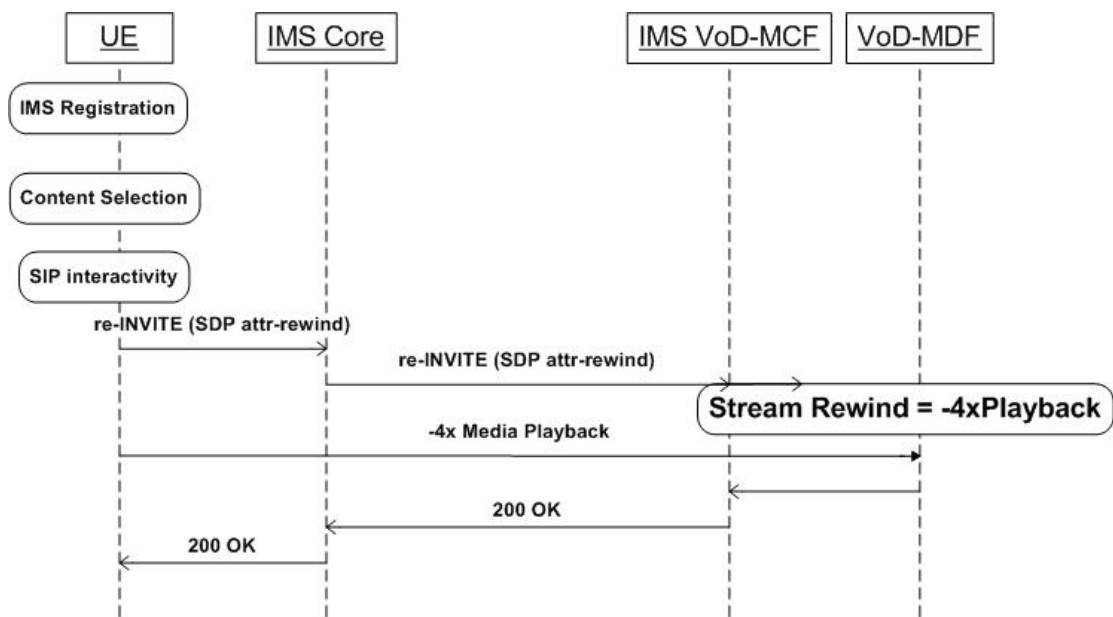


Figure 3.6: SIP-based Video Interactivity Sequence Diagram: Rewind

The proposed design could incur high latencies due to many IMS core entities (P-CSCF and S-CSCF) that need to be traversed before reaching the MCF. The high latencies experienced using the SIP protocol compared to RTSP over the IMS will be demonstrated in the evaluations carried out in the following chapters.

3.2.2 A Three-Layered Video Server Architecture

To minimise the impact that interactivity requests have on the video server, users will be allowed to specify (through service profiles) whether they would like to have a fully interactive session that allows any number of requests or a reduced interactivity session that only allows a limited number of interactivity requests to be processed. Some profiles will allow users to send a limited number of video interactivity requests to the video server thus reducing the possibility of server over-loading. This will ensure a service level scheduling and management of the interactivity load on the video server while also ensuring that the service experience conforms to the user expectations. To make this possible in a real world scenario, a user can be charged (or even given service credits) for choosing certain service profiles.

This solution is adopted in the proposed design for three reasons. Firstly, it meets the user QoE expectations by executing requests that the user subscribed to. Secondly, the MDF video processing load is expected to reduce when some requests from the user are not forwarded from the MCF to the MDF. Thirdly, the MCF does not need to rely on a feedback mechanism from the MDF in order to decide whether to process a request. This simplified approach to reducing the video server load has minimal impact on both the latency of processing requests and the complexity of the design⁶.

The three layered architecture for IMS-based VoD services for the reduction of the video server

⁶An overload signal could be sent to the MCF when the server is starting to overload. The advantage that this design offers is to obviate the need for continuously polling the MDF for resources when deciding whether or not to admit a video interactivity request. As discussed, the main decision logic for processing a request is at the MCF, this logic primarily makes use of the user profile information and is largely independent of MDF signals.

load ⁷ is shown in Figure 3.7.

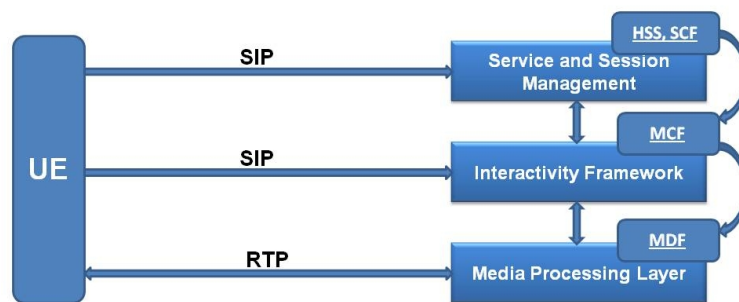


Figure 3.7: Proposed Three Layer Interactivity Framework

Session Management Layer

Session Management layer (SML) is responsible for authentication, authorisation as well as management of various service requests. This layer is thus responsible for executing service activation functions that enable video stream set-up, integration with other services (e.g. VoIP over TV) and service accounting (e.g. billing). Advertisements will be managed at the service management level. The time scheduling of various video streams can also be performed at this level. This layer therefore manages all the clients' sessions and makes sure that the session quality is to the level of the user subscriptions. The high level scheduling of video streams involves the possible prioritisation of particular some clients' requests over other clients' requests. This particular functionality is however not implemented in the proposed solution and it falls outside the scope of this study.

As per TISPAN and ATIIS IPTV standards⁸, the SML layer is hosted at the IPTV Service Control Function(IPTV SCF). The IPTV SCF supports the service and session requests from different subscribers.

To ensure and enforce fair allocation of resources to VoD subscribers user profiles will be used for different VoD packages. These user profiles are stored in the HSS. The decision logic for allowing and rejecting requests based on the user subscription profile is implemented on the MCF (which retrieves profile information from the HSS).

Figure 3.8 below shows an extended User Profile Server Function (UPSF) which will be used to manage interactivity. The TISPAN based architecture [1] has variables for keeping track of Bookmarks, Broadcast sessions, scheduled Network-Private Video Recorder (N-PVR) videos, and available VoD content⁹.

The programme or movie ID and the media delivery status are one of the variables stored to keep track of various videos. These variables have been extended to include interactivity variables. The added variables are: TPAAllowed - which indicates what trick modes are allowed; VadaptAllowed - which checks if video adaptation is enabled; Tmodestatus - which is a boolean variable indicating whether any interactive features are allowed; contextEnabled - which is boolean variable indicating whether various user service contexts (such as HOME, when the user is accessing a video

⁷This is achieved through the use of MCF-based profiles' mapping framework. The mapping framework is simulated in terms of the selective processing implemented in the evaluation platform discussed in the following chapter.

⁸See Appendix A

⁹The TISPAN standard uses the 'CoD' label to include all 'Content-on-Demand' audio and video content. Of particular interest to this study is the service description parameters for a CoD.

service from a High Definition Television (HDTV) at home, or WORK when the user is accessing the video service from a handheld device at work) are enabled or not.

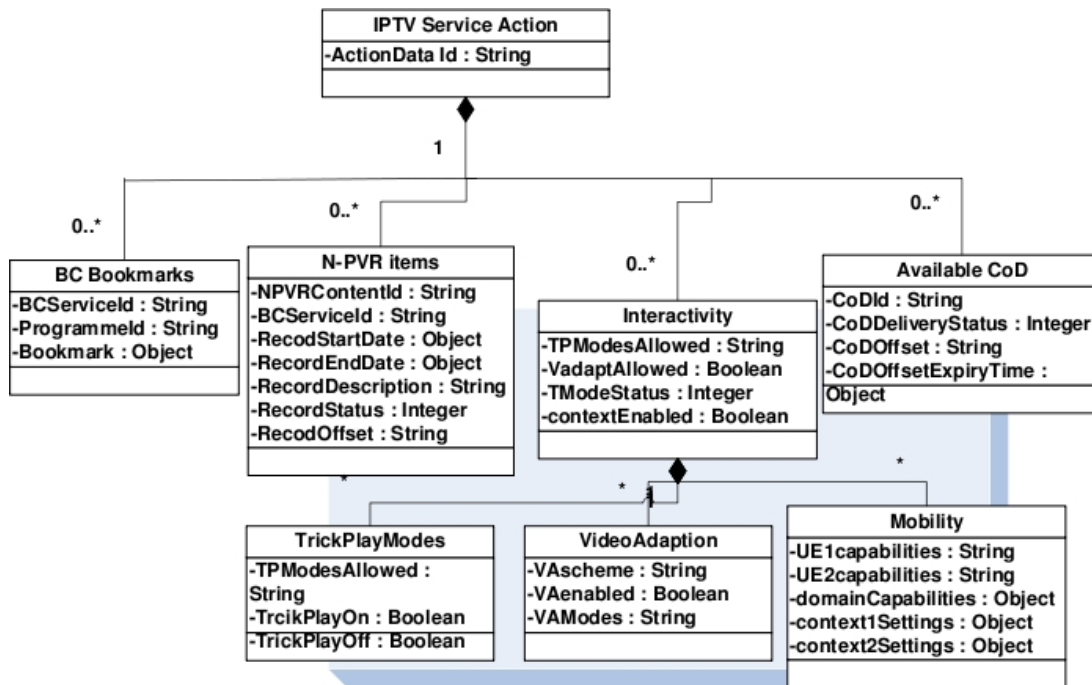


Figure 3.8: The TISPAN User profiles with the Proposed 'Interactivity Profile' Extensions

This layer will be hosted at the IPTV Service Control Function and will therefore be accessed through the ISC interface (as indicated in Figure 3.2), and the user will specify the video session parameters in a SIP/SDP message. The SCF will be able to directly relay the client service profile information to the MCF without involving the IMS core, as allowed by the TISPAN IPTV standard.

Interactive Framework Layer

The Interactive Framework Layer (IFL) is a layer, at the MCF, which is the main decision logic for TLVIF. This layer handles video interactivity requests that directly alter the video streaming session. As shown in Figure 3.9, a received interactivity request is evaluated on whether it is allowed based on the user subscription profile. If this request is allowed this layer will both process the incoming SIP request and forward the request to the media processing and streaming layer (hosted by the MDF). A SIP 200 OK response will then be sent. If this request was not allowed a 400 Bad Request response is sent to the client. The IFL does not proceed to send the request to the MDF if the request was not allowed.

The interactivity profile variables include information about comprehensive call information such as which interactivity functions are allowed or disallowed at the server. For every client interactivity request, the Media Control Function (MCF) will evaluate whether the request is allowed as per the user profile information from the HSS and the service subscription information obtained from the SCF. The decision logic therefore occurs at this level. All requests received from the SML will be examined at this layer (which is in the MCF) to be forwarded to the MPL.

Through the use of the interactivity profile information of each user, this layer will be able to curtail the amount of resources used and improve security at the MDF by reducing the total number of interactivity requests processed (that could overload the MDF).

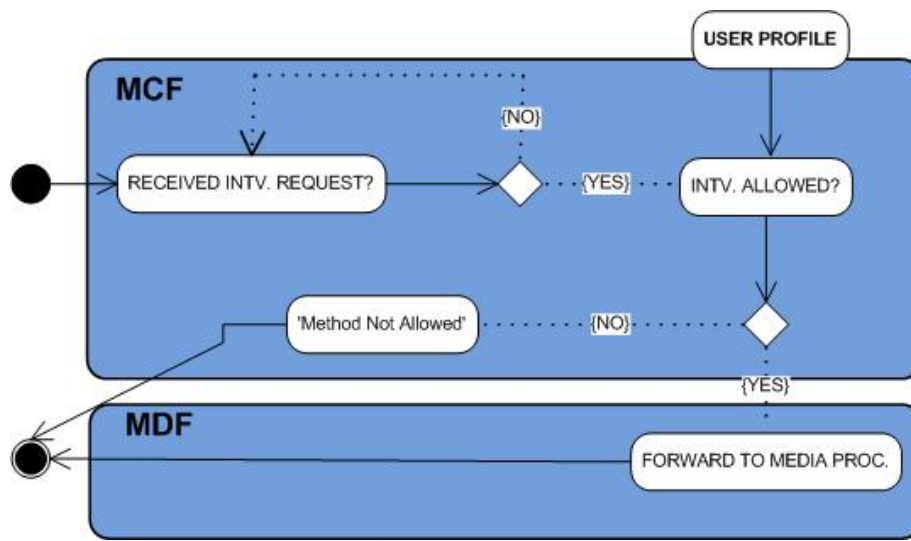


Figure 3.9: Interactive Framework Layer

Consider the following example. Assuming a directly proportional relationship between the amount of video interactivity requests received at the MCF and the overall video server processing load (MCF + MDF) without the proposed TLVIF, five users attempting video interactivity with the video server with full interactivity enabled will cause a certain processing load in the video server. When only 2 out of 5 users are allowed with full interactivity, and the rest only have 'pause' and 'play' enabled a defined (i.e. reduced) number of times, the overall load of the MDF is decreased and thus server over-loading is less likely to occur.

Media Processing layer

The Media Processing Layer (MPL) is on the media plane, the main function of this layer is to perform unicasting of video streams to video clients. The other function of this layer is to keep track of different interactive states of a video stream and video frames. A video stream can be in the following states:

1. PLAYING, to indicate that the current stream is currently being played
2. PAUSED, to indicate that the current stream has been paused
3. STOPPED, to indicate that the current stream has been stopped
4. STREAM_POSITION works alongside the above states to indicate the exact position of the video during the PLAYING, PAUSED or STOPPED states.

The video frames can set according to the following parameters:

1. Video frame size, to video screen size of the video
2. Frame rate, to indicate the rate at which the video frames are being encoded or decoded
3. A combination of 1 and 2, depending on the settings of the video service context variable (as explained in the UPSF profile in Figure 3.8)

This layer ensures that every requested change in a video stream state or frame is processed as quickly as possible. This is ensured by the highly efficient gstreamer pipeline modules that are used in the implementation of the video server. Figure 3.10 shows how the MPL layer will keep track of each of the user unicast streams and how it will process various requests.

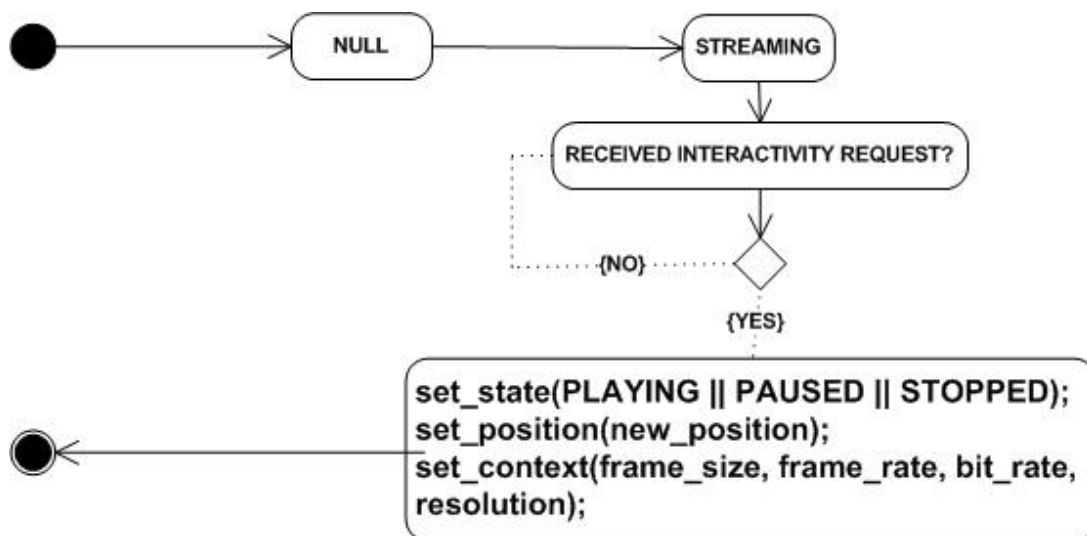


Figure 3.10: Media Processing Layer

The figure above shows that when a video stream starts, the state of the video session is changed from NULL to STREAMING. Once in the streaming state, the MPL layer is then ready to receive requests for stream manipulation from the IFL layer. It must be noted that no decision logic will occur at this level, all requests received from the IFL will be processed (the IFL layer is responsible for examining whether requests will be forwarded to this layer or not). The implementation details of this layer are further provided in Appendix B, Section B.4.

TLVIF Signalling

The framework ensures that user profiles, the interactivity experience and the media processing are modularised in order to provide scaling of the video server by limiting video interactivity requests (depending on the user service profile).

In Figure 3.11, after the User Equipment (UE) registers with the core, a video on demand session is initiated by sending an invite to the IPTV application server (in this case the SCF). The SCF then sends the latest VoD content to update UEs Electronic Program Guide (EPG) which allows the user to perform content selection.

The client navigates through the available VoD content and then chooses a particular content of interest by sending an INVITE request to the SCF to set-up a VoD session. The SCF searches for the appropriate MCF and initiates the media session. The selected MCF collects the necessary client details such as the user interactivity profile information from the SIP call. A VoD session is then initiated and an OK message is send to the UE.

To perform a video pause, the client sends a SIP re-INVITE request to the MCF. The request is examined at the MCF by the IFL (as described above). The IFL uses the user profile information from the SCF to decide if a video pause is allowed. If a video pause is allowed, the IFL (in the MCF) sends a procedure call to the MPL (in the MDF, with 'pause' as the video interactivity request parameter). The MDF executes the request and sends a procedure completion status to

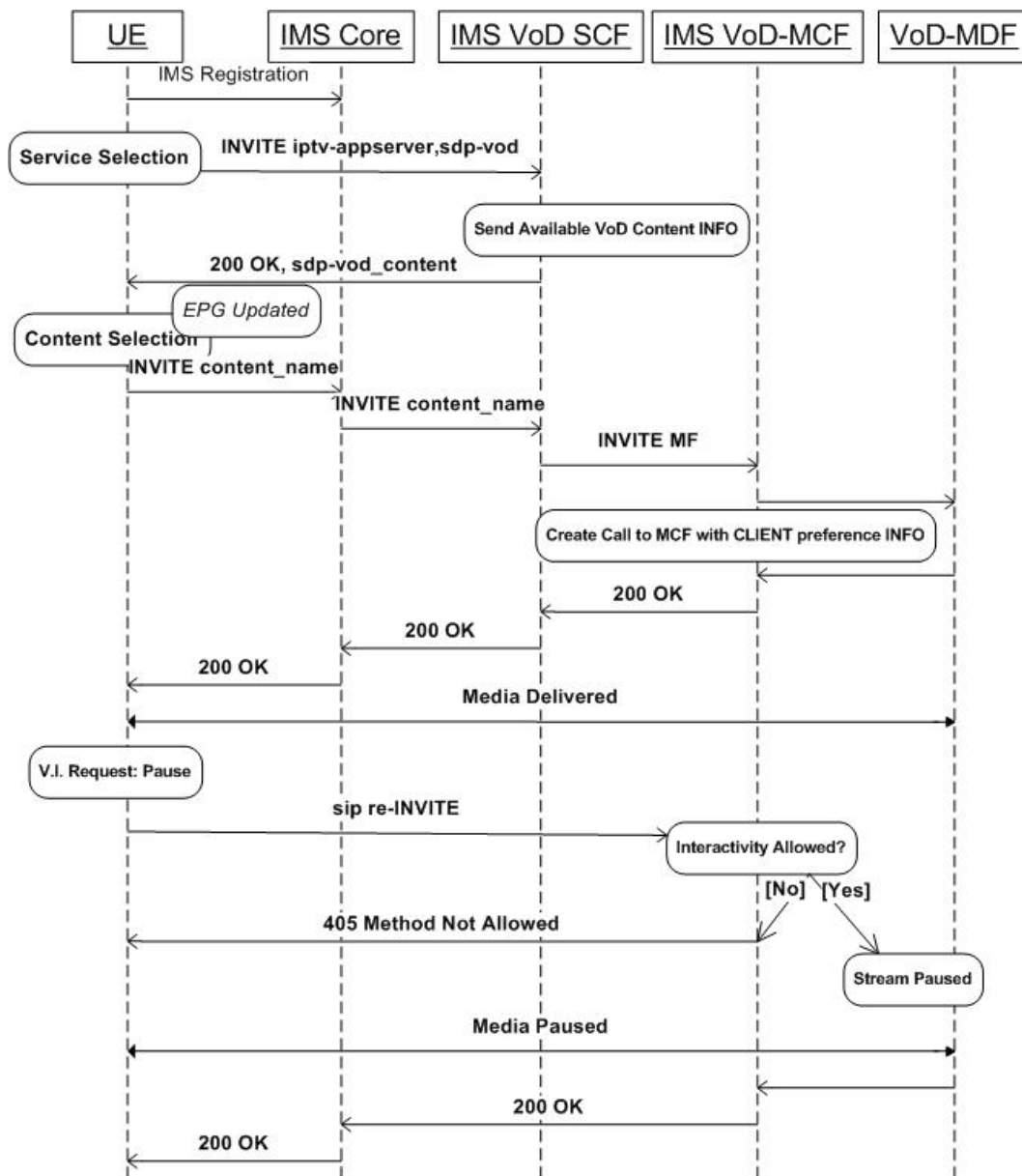


Figure 3.11: Sequence Diagram for the TLVIF

the MCF. If 'pause' is not allowed, a SIP response message indicating that the operation is not allowed is sent to the UE. The MDF procedure is not invoked.

In Figure 3.12, for a 'rewind', the client sends a SIP re-INVITE request to the MCF. The request is examined at the MCF. The MCF uses the user profile from the SCF to check if 'rewind' is allowed as per the user service profile (as shown in the algorithm presented in Figure 3.9)¹⁰. If 'rewind' is allowed, the client request is forwarded to the MDF for execution, with the playback rate set to four times the normal playback rate in the negative direction. If 'rewind' is not allowed, a SIP response message indicating that 'rewind' is not allowed is sent to the UE. A forward request is also shown in the figure.

¹⁰The implementation of this for the evaluation platform is discussed in the following chapter.

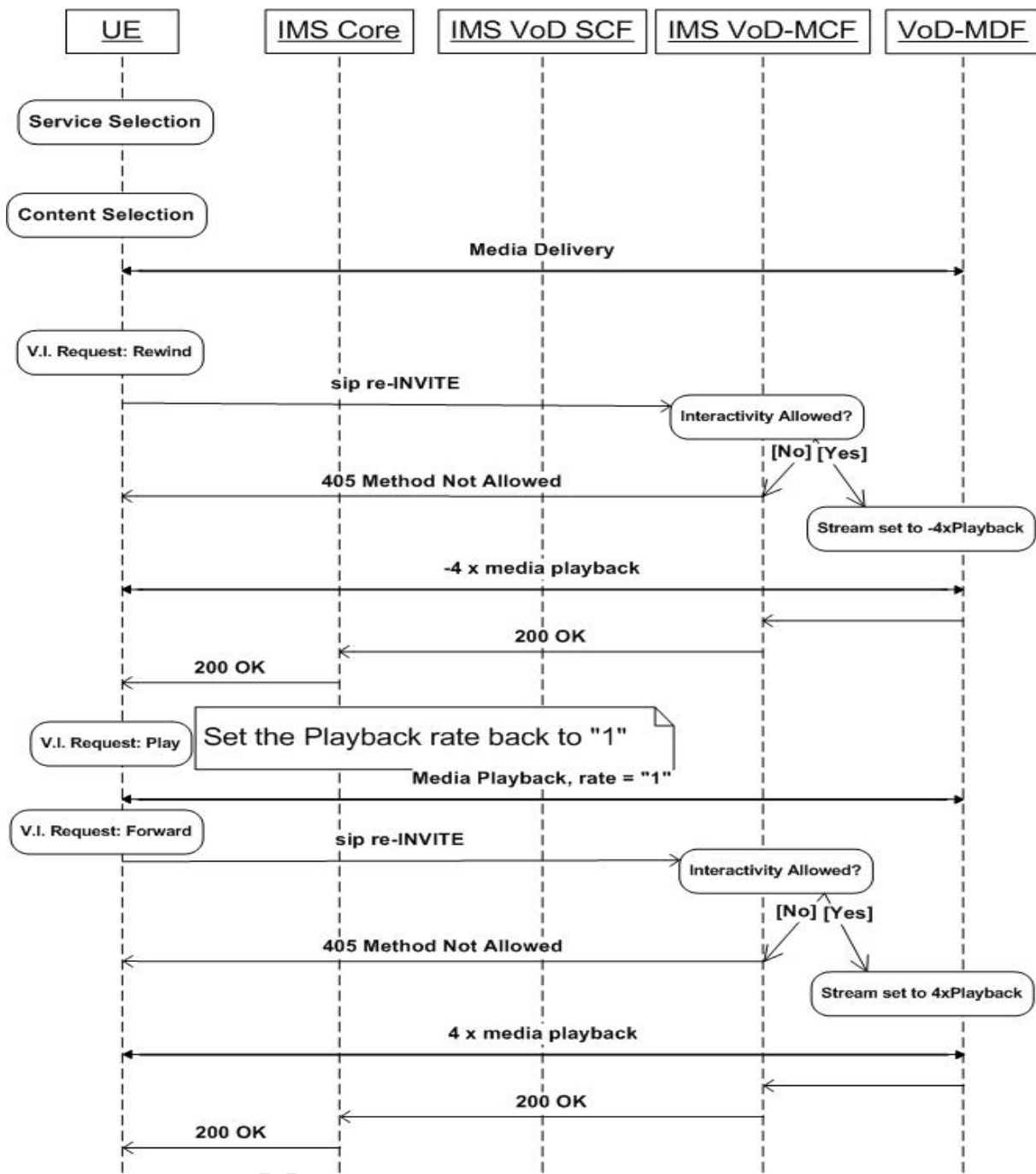


Figure 3.12: Sequence Diagram for the TLVIF: Rewind, Forward

3.2.3 A SIP-based Platform for Interactivity Development

The interactivity framework links session parameters (obtained during video session set-up) with interactivity profiles (by means of SIP call information and video session states).

In order to allow for the development and proliferation of new interactivity features, the proposed framework uses the approach of employing SIP/SDP for the development of future video interactivity requests. This approach is chosen as it leverages the flexible design of the SIP protocol. The video interactivity functions that are supported by the proposed system are shown in Table 3.1. All these functions are implemented by using the `a=siptrick:x` attribute, where `x` is the video interactivity function.

Various new requests such as video adaptation can be performed simply by adding a new attribute

Interactivity Function	Process	Protocol Used
SETUP	Send session setup request to the SCF	SIP/SDP
PAUSE	Send PAUSE request to the MCF	SIP/SDP
PLAY	Send PLAY request to the MCF	SIP/SDP
FORWARD	Send FORWARD request to the MCF	SIP/SDP
REWIND	Send REWIND request to the MCF	SIP/SDP
STOP	Send STOP request to the MCF	SIP/SDP
ADAPT	Send video adaptation request to the MCF	SIP/SDP
SCENE-SEL	Select scene selection request to the MCF	SIP/SDP
PLAY-RATE	Change the playback speed	SIP/SDP
SLOW-F/RW	Reduce forward / rewind rate	SIP/SDP
FAST-F/RW	Increased forward / rewind rate	SIP/SDP

Table 3.1: Video Interactivity Functions

or modifying existing ones as shown in Figure 3.13.

```

(SIP Header)
SIP/2.0 200 OK
Via: SIP/2.0/UDP
...
(SDP Body)
v=0
.
.
.
.
m=video 38080 RTP/AVP 34
a=rtpmap:34 H263-1998/90000
b=AS:128
a=framesize:34 -
a=recvonly
a=siptrick:adapt
a=framerate:20

```

Figure 3.13: The Video Adaptation SIP Message

As shown in the figure, only the `a=siptrick:adapt` and the `a=framerate:20` attribute are new. The rest are existing attributes that are part of the Session Description Protocol (SDP) design[50]. The newly added attributes will be used to indicate to the video server that the client wants to change the framerate to 20 frames per second.

Figure 3.14 shows a sequence of a new interactivity request, that allows for scene selection, being processed. For all new video interactivity requests, the MCF needs to be updated to handle the new requests and the client should be allowed (through the interactivity profile) to send the request. Other attributes can be added to effect various changes in the video signal. This solution is an improvement on the iPersonal framework proposed by Muntean et al [49] in that it implements

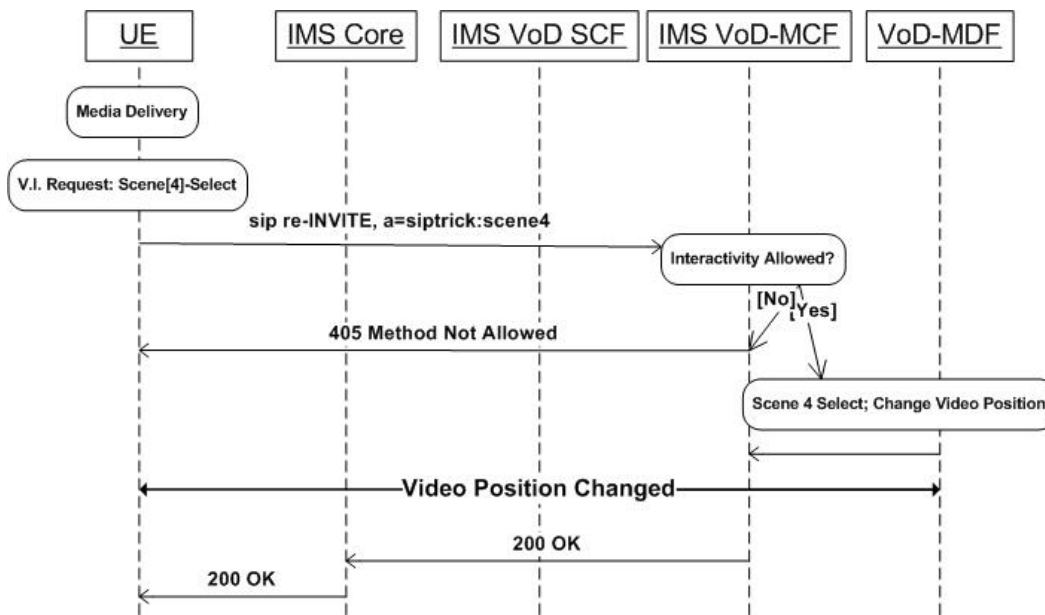


Figure 3.14: Sequence Diagram for New VI's in the TLVIF

how signals are sent to the Adaptation Engine¹¹ using SIP signalling.

3.3 Chapter Summary

This chapter has presented the proposed Three-Layered Video Interactivity Framework (TLVIF). The proposed design differs from the SIP-based video interactivity design proposed by Sivasothy et al [2] in that instead of a header line being modified, only an extra attribute `a=siptrick:` is added to the SIP message structure to effect media playback.

Video Interactivity Solution	Author	Interactive Latency	Interactive Development	Video Server Load
Method 1,2	Riede et al	RTSP latency	Uses both SIP and RTSP	None
Enhanced IMS	Khan et al	Translation proxy latency	Uses SIP	None
Extended IMS	Khan et al	RTSP latency	Uses both SIP and RTSP	None
Unified SIP	Sivasothy et al	SIP header latency latency	Uses SIP	None
TLVIF	Proposed	SIP/SDP latency	Uses SIP and SDP features	Introduces the Interactivity profile to reduce server load

Table 3.2: Proposed vs. Existing Solutions

The SIP/SDP body has been employed to implement a variety of new attributes for various interactivity requests.

¹¹See Section 2.3.3

In order to reduce the server processing load, every video interactivity request is examined in the proposed framework. The requests are examined at the MCF. If a video interactivity is allowed, a procedural call is sent to the MDF for execution. If the request is not allowed, a SIP response message indicating that the interactivity request is not allowed is sent to the UE.

Table 3.2 shows how the proposed solution differs from the designs discussed in Chapter 2. It is clear that one major advantage of the proposed solution is that of providing a design that reduces the MDF processing load. The proposed design also allows for SDP-based future video interactivity development.

The actual video interactivity latencies experienced in the proposed (SIP/SDP-based) solution need to be evaluated. The impact of the interactivity load on the video server as well as the proposed TLVIF will be examined on an evaluation platform in the following chapters.

University of Cape Town

Chapter 4

Architecture of the Evaluation Platform

The aim of this chapter is to present an evaluation platform that will be based on the design discussed in the previous chapter to allow comparisons between latencies on the SIP and the RTSP environments as well as to investigate the hypothesis on the relationship between video interactivity and the video server processing load proposed in the previous chapter.

In this chapter, we will ascertain the validity of the proposed three-layered video interactivity platform in order to evaluate the latencies in the two environments referred to above.

4.1 Requirements of the Evaluation Platform

Firstly, the evaluation architecture should be implemented with little complexity to allow for thorough evaluation and repeatability of the evaluation results. The evaluation should together with widely available tools provide detail on the processes and procedures used to implement the design.

Secondly, the architecture should be TISPAN IMS VoD standard compliant to allow for IMS compliant implementations of RTSP- and SIP-based video interactivity, where the assessment of the impact of both protocols on the video server load is possible.

4.2 Decision on Test-bed Implementation and Tools Used

The following evaluation methods were considered for the evaluation platform:

- Test-bed set-up. A test-bed environment includes implemented components of the IMS core, the SIP stack, the RTSP stack as well tools for evaluating performance (i.e. the processing load and latency). Test-beds are a realistic and robust method of evaluating the components of a network platform. Open test-bed platforms do allow for easy implementation and transparency of testing processes and results. Test-beds also allow for video server performance to be easily measured on Linux based test-bed platforms using tools such as `/proc/cpuinfo/`. A test-bed evaluation platform was chosen due to the availability of the TISPAN IMS standard OpenIMS test-bed developed by Fraunhofer fokus. This meant that development time could be reduced. The current UCT IMS test-bed, which includes an IMS Client, allows for the convenient evaluation of the performance of the already implemented

SIP signalling functionality. The OpenIMS platform was developed on linux, this allows for the use of linux-based tools such as *Dstat*. *Dstat* allows for measurement of the processing load on the workstation. The C-based OpenIMS platform also allowed the use of a tool such as *gettimeofday* that allows for measuring latency.

- Network Simulation. Network emulators simulate the conditions and properties of an existing network to assess various performance metrics. This involves the use of software tools that employ the use of a mathematical model and a network emulation to model a real world system. Network emulators are useful in that, even though simulated tools instead of real network tools are used, the actual performance and results of the simulation can be designed to be very close to that of a real world environment set-up. With regards to the first requirement stated above, emulators do allow for repeatability of results as most of the environment is in software code which can easily be reproduced on most computing platforms. *NS-2*, for example, can limitedly act as full emulation platform for the IMS as currently its main modules are concerning the physical and data-link layer environment. Given the availability of the already existing evaluation platform, and the significant amount of development required to produce a Network Simulation, a test-bed evaluation platform remained a better option. Examples of network simulators are OPNET and *Network Simulator 2 (NS-2)*.
- Some tools for evaluating the video processing load and the impact of transport protocols, such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP), on server performance have been proposed. Lee et al [59] present a toolkit for the evaluation of RTSP based streaming servers. They propose the use of the PseudoPlayer and the PseudoMonitor for assessing RTSP server performance. This toolkit could not easily be appropriated for use in an IMS environment.

The existing test-bed platform was adopted and enhanced to address the requirements of this research.

4.3 Test-bed Software Overview

For the evaluations carried out in this thesis, the Fraunhofer Fokus Open IMS Core (FFOIMS) and the UCT IMS test-bed (UIMS) were used. The FFOIMS is an Open IMS test-bed platform developed by a team of Engineers at the Fraunhofer Fokus NGN group (FFG) in Germany. The FFOIMS is based on the design of Open Sip Express Routers (OpenSER) initially developed by the OpenSER team. The OpenSER design includes SIP registrar servers, SIP redirect servers, SIP proxies and SIP application servers [60]. The main modifications made by FFG to the Open SER design was the introduction of the IMS Home Subscriber Server (HSS), the IMS compliant SIP proxies (CSCFs) and IMS compliant SIP signalling. The FFOIMS implementation does not include an IMS compliant video server solution.

The Open IMS Core (shown in Figure 4.1) is an Open Source implementation of the IMS Call Session Control Functions (CSCFs) and a light weight Java-based Home Subscriber Server (HSS). The Open Source MySQL database was also used in the HSS [61, 5]. The advantage of using the Open Source software is that since the source code is widely available, the design of the test-bed platform can easily be modified to accommodate new features that may be required.

The UCTIMS IPTV (UCTtv) server will be used. The UCTtv was developed at the Communications Research Group (CRG), University of Cape Town [30]. UCTtv is based on the VideoLAN

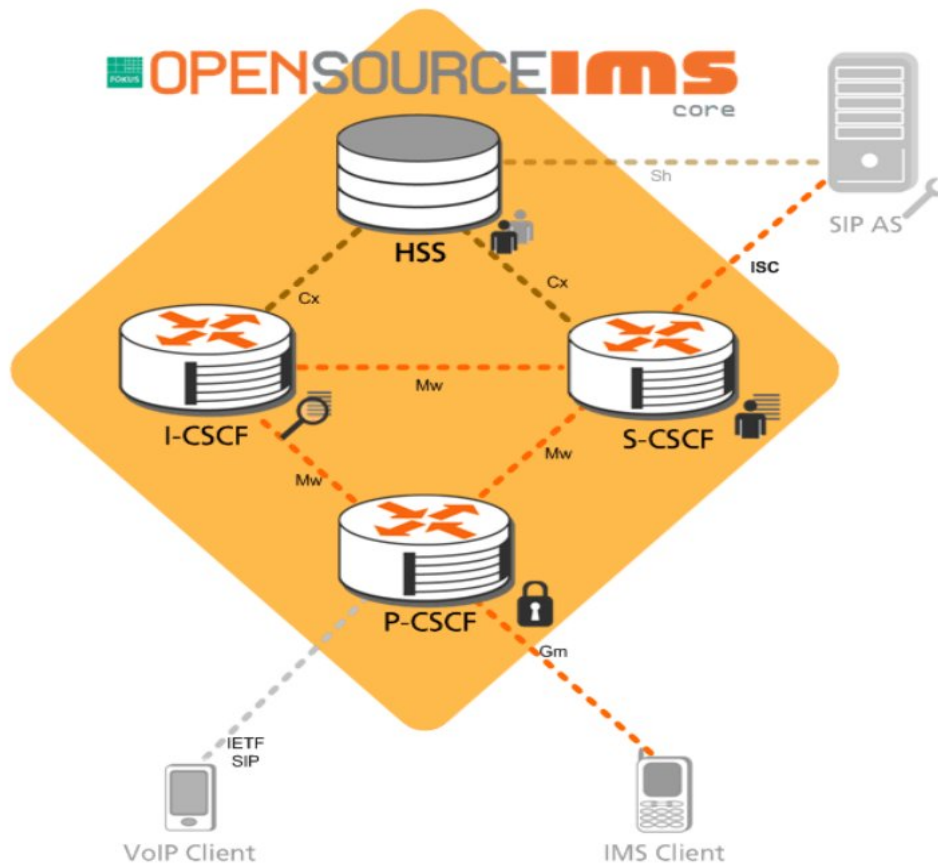


Figure 4.1: Fokus Open IMS Core Project [5]

project maintained by the VideoLAN Organisation [62]. VideoLAN is a free and open source cross-platform multimedia player and server framework that plays most multimedia files and supports various streaming protocols. The VideoLAN project supports control of streams by the RTSP protocol. The UCTtv server provides basic features such as handling RTSP requests and initiating RTSP sessions with third-party media servers [63].

The UCT IMS Video Client (UCTvc) was also developed alongside the UCTtv server. UCTvc was based on the UCT IMS Client earlier developed in the CRG. The UCT IMS Client was developed to provide a user friendly client interface with the FFOIMS based on IMS compliant SIP signalling [30]. As an extension to this, UCTvc included new features of a video interface as well for RTSP-based video session control.

To implement the three-layered video interactivity architecture several UCT IMS test-bed components were used. Most components were modified to suit the requirements of the evaluation. These modified components will be explained in the sections below.

4.4 Test-bed Software Developed

This section outlines some of the software developed or modified for the purpose of this study.

4.4.1 UCT IMS Client

Software from the UCT IMS test-bed project was modified to accommodate the evaluations that need to be carried out in this study. Section B.3 in Appendix B shows the contribution of SIP-based video interactivity implementation on the UCTIMS client.

IPtv VoD Interface

Another tab was added to the menu bar to allow for selection of movies. The menu items bar is shown in Figure 4.2 that has buttons for Movies Channel, Documentaries Channel, Youtube Channel and TV Series Channel. A user can click on one of the buttons to select a channel request. After a channel is selected, a video can then be selected by clicking on one of the videos displayed. The channel stream is enabled by clicking the 'Call or Answer' button.

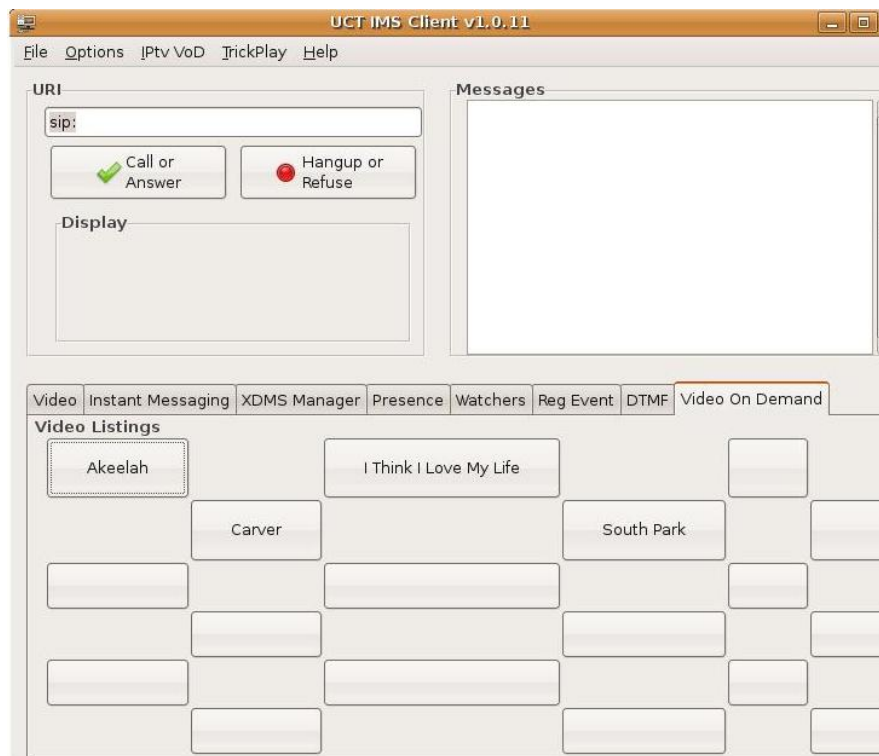


Figure 4.2: The UCT IMS CLIENT

Trick-play Interface

An extra menu item was added to the menu bar to allow for trick-play triggering. The menu has buttons for PLAY, PAUSE, FORWARD and REWIND. A user can click on one of the buttons to activate a trick-play request. Figure 4.3 shows the implemented interface for requesting trick functions.

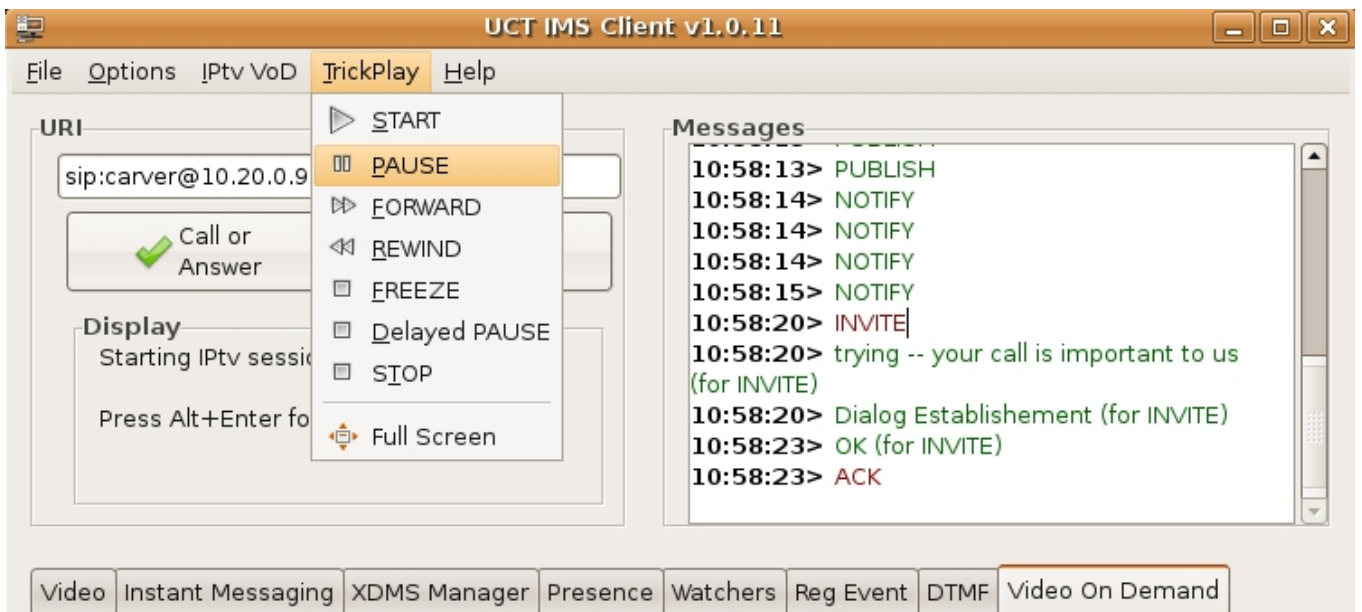


Figure 4.3: Trick-play Interface

SIP-based Interactivity Signalling

The eXoSIP [64] SIP stack and a bare-bone implementation of RTSP signalling¹ were used to perform video interactivity signalling. The methods added were:

- `on_trickfunction_1_activate` for handling SIP and RTSP 'play' requests from the client interface
- `on_trickfunction_2_activate` for handling SIP and RTSP 'pause' requests from the client interface
- `ims_call_reinvite_trick()` for creating an SDP for SIP-based video interactivity

The signalling methods modified were:

- `rtsp_PLAY()` for sending RTSP 'play' to the server
- `rtsp_PAUSE()` for sending RTSP 'pause' to the server
- `common_exosip_handler()`, modified to handle SIP interactivity responses from the server
- `vod_invite_SCF()` for inviting the SCF-MCF server

Figure 4.4 shows program files used to implement SIP-based interactivity. To start a session, a SIP re-INVITE signal is sent from the UCT IMS CLIENT interface (captured in `callbacks.c`) to the media server. To pause a session, the `on_trickfunction_2_activate` function is triggered with the PAUSE button from the client interface. Similarly, to resume a session the `on_trickfunction_1_activate` function is triggered. Following this, a full SIP re-INVITE message is packaged and sent through by the use of `eXoSIP_call_send_request` method. Figure 4.4 also shows corresponding RTSP methods: `rtsp_PLAY()` and `rtsp_PAUSE()` methods, which are called from the `rtsp.c` file.

¹Partially implemented as part of the UCT Advanced IPTv by Robert Marston

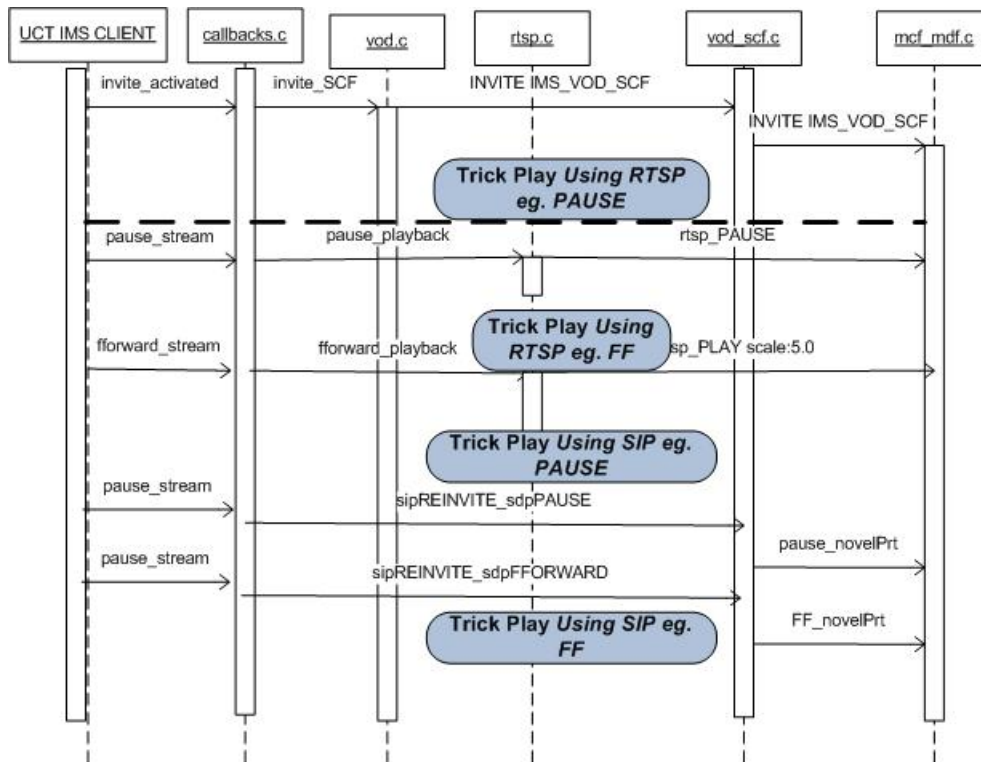


Figure 4.4: Interactive Platform

The software code for these methods is provided in Section B.3.1 in Appendix B. These methods are similarly passed on to the `on_trickfunction_x_activate` methods and follow the same procedure for sending the requests to the media servers.

4.4.2 TLVIF Layers Implemented

In terms of the proposed three-layered framework, the implementation of the TLVIF video server in the evaluation platform includes both the Interactive Framework Layer (IFL) and the Media Processing Layer (MPL). The IFL is implemented on the UCTtv server, as part of the MCF. The MPL is implemented on the UCTtv, as part of the MDF. The video server used for the evaluation platform therefore had both the MCF and the MDF entities implemented.

The implementation of the Session Management Layer (SML) was not completed. The reason behind this is that the SML required the use of a complete IMS compliant Session Control Function (SCF) entity and an enhanced Charging Platform (CP) that will implement the service profiles inherent in the TLVIF framework. The actual design and implementation of both the complete IMS compliant SCF and CP platform was however outside the scope of this study as a complete evaluation of the proposed TLVIF framework did not necessarily require the use of a fully functional SML layer. The main objective of the TLVIF was to reduce the video processing load on the video server through limiting the number of video interactivity requests forwarded to the MPL. This was demonstrated through an emulation of a function in the MCF (i.e. IFL) that decides whether a request should be forwarded. This emulation will be carried out on the test-bed. The functions of the service profiles in the HSS and the SCF that restrict certain users from executing several video interactivity requests are emulated in the evaluation platform. The tests carried out to emulate the TLVIF are explained in Section 4.7 below.

4.4.3 The TLVIF Video Server

One of the modifications of the original UCTtv server implementation was the use of the `gstreamer` framework for video interactivity processing, instead of the initial VLC-based video streaming platform. The VLC-based video streaming platform was not used as it does not support SIP-based signalling. The `gstreamer` framework contains basic modules that can be modified to support both SIP-based signalling

More details on the new media processing functions (i.e. `g-streamer` based forward/rewind/pause functions) implemented are provided in Section B.4.

Method Name	Function	File name
<code>get_sip_events</code>	Receives and processes SIP interactivity requests from the video client	<code>uct_mcf.c</code>
<code>add_stream</code>	SETUP-Initialise video channel using <code>gstreamer</code> framework	<code>uct_ims_mdf.c</code> (calls <code>initialize_channel</code> in <code>video.c</code>)
<code>play_stream</code>	PLAY-Processes play requests from the video client	<code>uct_ims_mdf.c</code> (calls <code>play_client</code> in <code>video.c</code>)
<code>pause_stream</code>	PAUSE-Processes pause requests from the video client	<code>uct_ims_mdf.c</code> (calls <code>pause_client</code> in <code>video.c</code>)
<code>rewind_stream</code>	REWIND-Processes rewind requests from the video client	<code>uct_ims_mdf.c</code> (calls <code>rewind_client</code> in <code>video.c</code>)
<code>forward_stream</code>	FORWARD-Processes forward requests from the video client	<code>uct_ims_mdf.c</code> (calls <code>forward_client</code> in <code>video.c</code>)
<code>stop_stream</code>	STOP-Processes video stop requests from the video client	<code>uct_ims_mdf.c</code>

Table 4.1: Implemented TLVIF Server Methods

The UCTtv server was modified to handle SIP-based video interactivity. The methods implemented on the TLVIF server with the respective new as well as modified files are shown in Table 4.1. Most the methods explained were the software development contributions by the author to the UCTtv server in this study, with the exception of the `get_sip_events`, `initialize_channel` and `add_stream` methods, which already existed in the initial UCTtv implementation.

4.5 Test-bed IP-CANs

To support bandwidth intensive video streams, VoD clients require IP-CANs (IP Connectivity Access Networks) with large amounts of bandwidth. To ensure support for efficient interactive video streaming, the two IP-CAN networks used in the evaluation testing were WiMAX and Ethernet. Both of these networks support at least 10Mbps data rate. Other alternative IP-CANs such as 3G, ADSL and WiFi were not considered as they support insufficient bandwidth rates for interactive video streaming.

WiMAX

The WiMAX (802.16d) IP-CAN from BreezeMAX was used in the evaluation testbed. The WiMAX IP-CAN comprises of a Micro Base Station (MBS) that is linked to the backbone network. The MBS then broadcasts microwave signals to the Subscriber Station (SS) OutDoor Units (ODUs). SS ODUs are linked to the user equipment through the SS InDoor Units (IDUs) (shown in Figure 4.5). Figure 4.5 also shows the edge (Egress and Ingress) and core (Interior) routers of the WiMAX network.

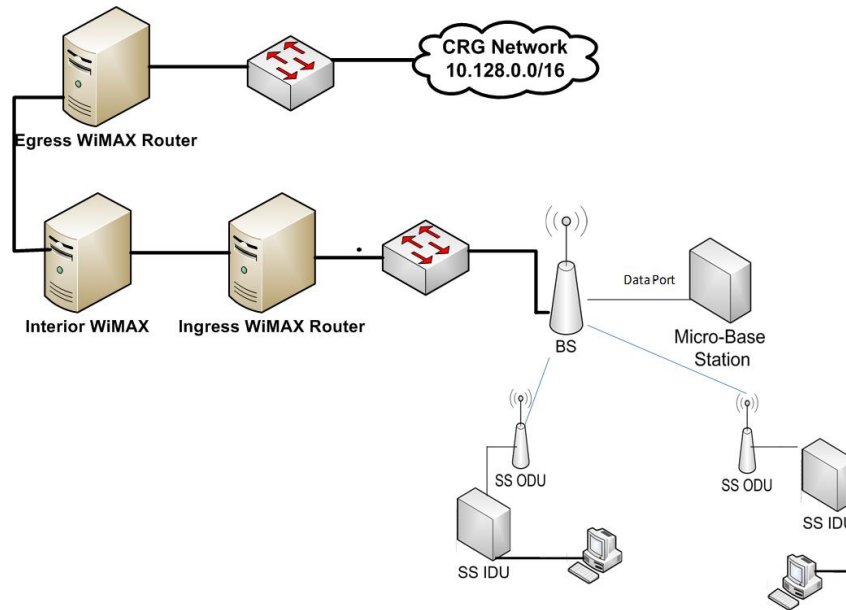


Figure 4.5: WiMAX Test-bed

IEEE 802.11 - Ethernet

A Local Area Network (LAN) with 100 Mbps Ethernet (IEEE 802.3) link speed was also used during the testing.

4.6 Setting Up the Evaluation Environment

4.6.1 SCF-MCF-MDF and The Open IMS Core

The IMS SCF and the IMS MCF were added to the HSS using the HSS web interface shown in Figure 4.6. In the configuration, both the SCF, MCF and the MDF were co-located in the same workstation. The reason for the co-location of the two entities was that no additional functionality was required to be implemented in either of the entities. The TISPAN IMS-IPTV standard does not provide a specification for an interface between the MCF and the MDF. In this implementation, a simple local procedural call is used to forward requests from the MCF to the MDF. When the called procedure is finished (i.e. 'forward' has been executed), the control is returned to the MCF. Although this implementation is not flexible, it is adequate for the evaluations required in this study.

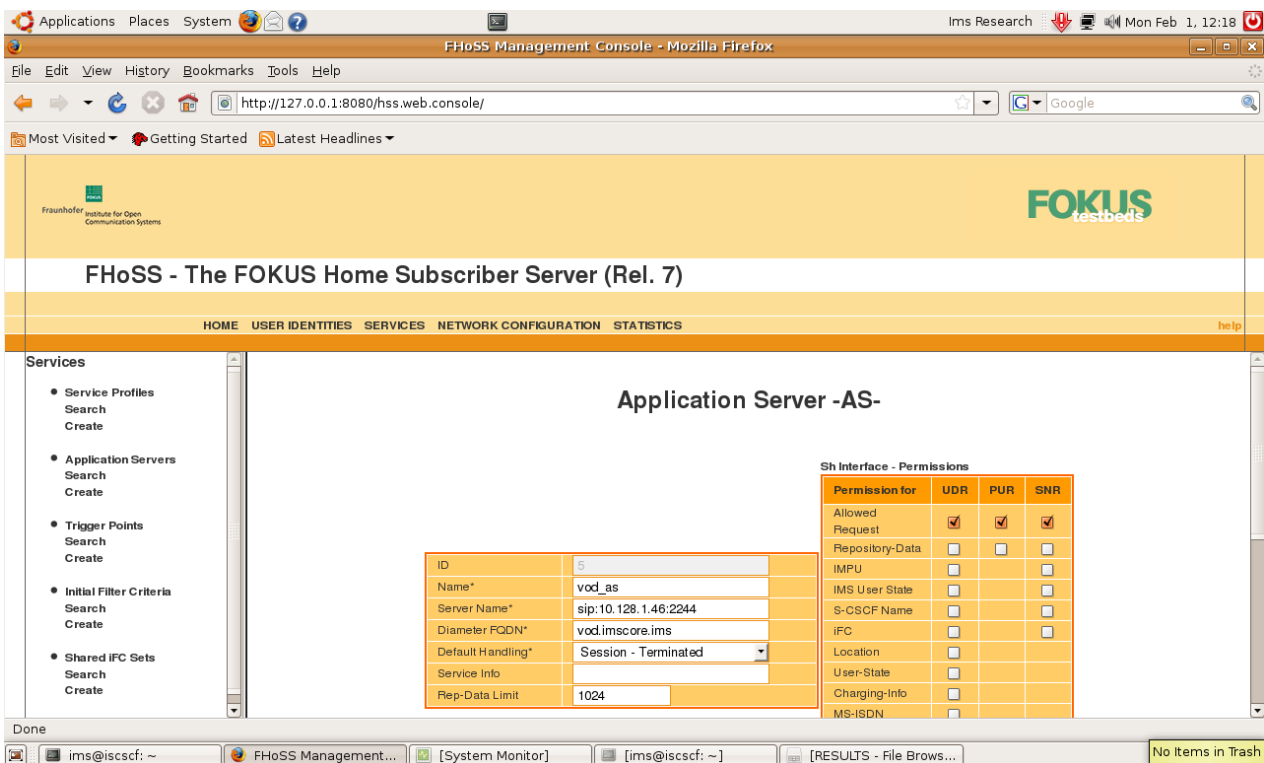


Figure 4.6: HSS Interface

Figure 4.7 below shows the IMS test bed with all the IMS Call Session Control Functions (CSCFs) physically co-located on the same workstation. The HSS has a separate workstation.

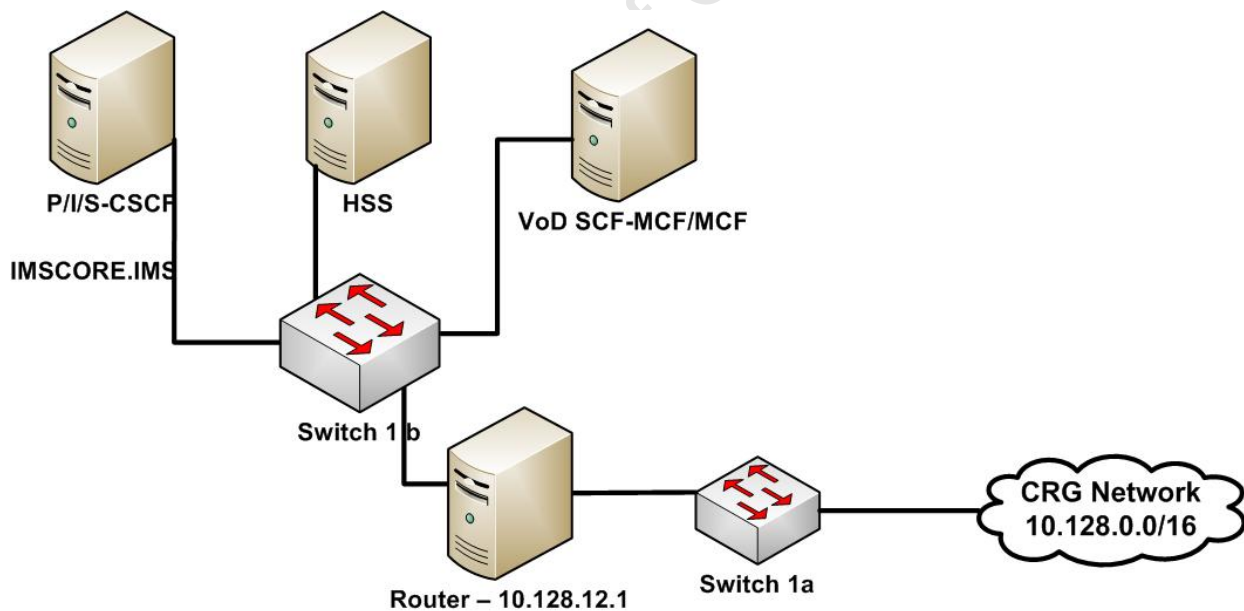


Figure 4.7: Open IMS Test-bed

4.6.2 Setting up the SCF-MCF-MDF

The SCF-MCF was set to use port 2244. This port is used for both initiating a video session as well for sending SIP-based video interactivity requests. The RTSP port number, for RTSP-based

video interactivity, was set to 9554. The SCF-MCF audio and video ports were set to 2222 and 2223 respectively.

4.6.3 Hardware Specifications

Specifications of the equipment used are provided in Table 4.2.

	HSS Workstation	Open Core CSCFs Workstation	VoD SCF-MCF Workstation
Processor	Intel Pentium Dual CPU @1.8GHz	Intel Pentium Dual CPU @1.8GHz	Intel Pentium 4 CPU @2.40GHz
CPU (MHz)	1203	1203	2412.351
Cache size (kB)	1024	1024	512
RAM size (kB)	1027052	1027004	515580
Operating System	Ubuntu 8.04.1 Hardy	Ubuntu 8.04.1 Hardy	Ubuntu 8.04.1 Hardy
OS Kernel	2.6.24-19-generic	2.6.24-19-generic	2.6.24-19-generic

Table 4.2: Software and Hardware Specifications

4.7 Setting up Measurements

The explanation of how latency and the video processing load were measured is provided below.

4.7.1 Latency Measurement

The timers for tracking the time from which a request is sent from the video client to the video server to when the reply received at the video client were implemented on the video client. The contribution of the video server processing to the latency was assessed by activating and deactivating the MDF functionality.

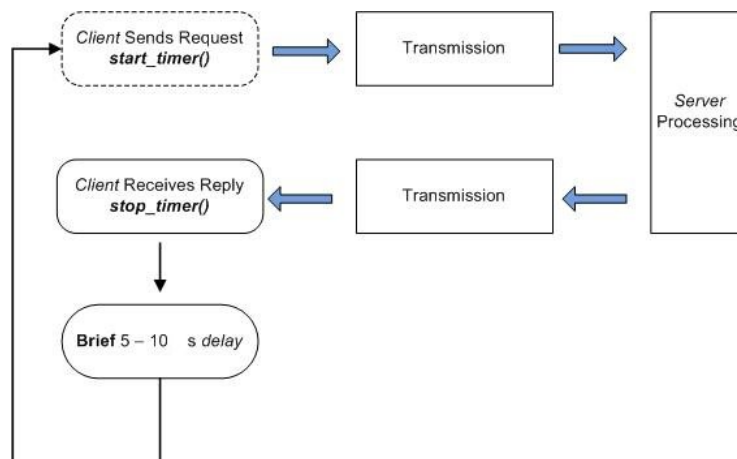


Figure 4.8: Latency Measurements

Figure 4.8 depicts an overview of latency measurements. The user first initiates a video session on the IMS core using SIP. After a session is established, the 'pause' button on the interface is used to pause a video, after which the request is sent to the video server and a reply is sent to and received at the video client. After the reply has been received, a delay is enforced (5s or 10s, chosen arbitrarily), before the next reply is sent. The setup was then automated to continue sending the next request without any manual intervention for a period of time. The delay is enforced between requests to avoid server over-loading which could compromise the measured latency results².

Figure 4.9 provides more detail on how the measurements were performed, To evaluate the RTSP-based video interactivity latency, the gettimeofday C method was used. Alternating PAUSE and PLAY requests were sent. This was achieved by sending a request using the callback.c function corresponding to the video interactivity request. The callback.c function is responsible for sending requests to the MCF.

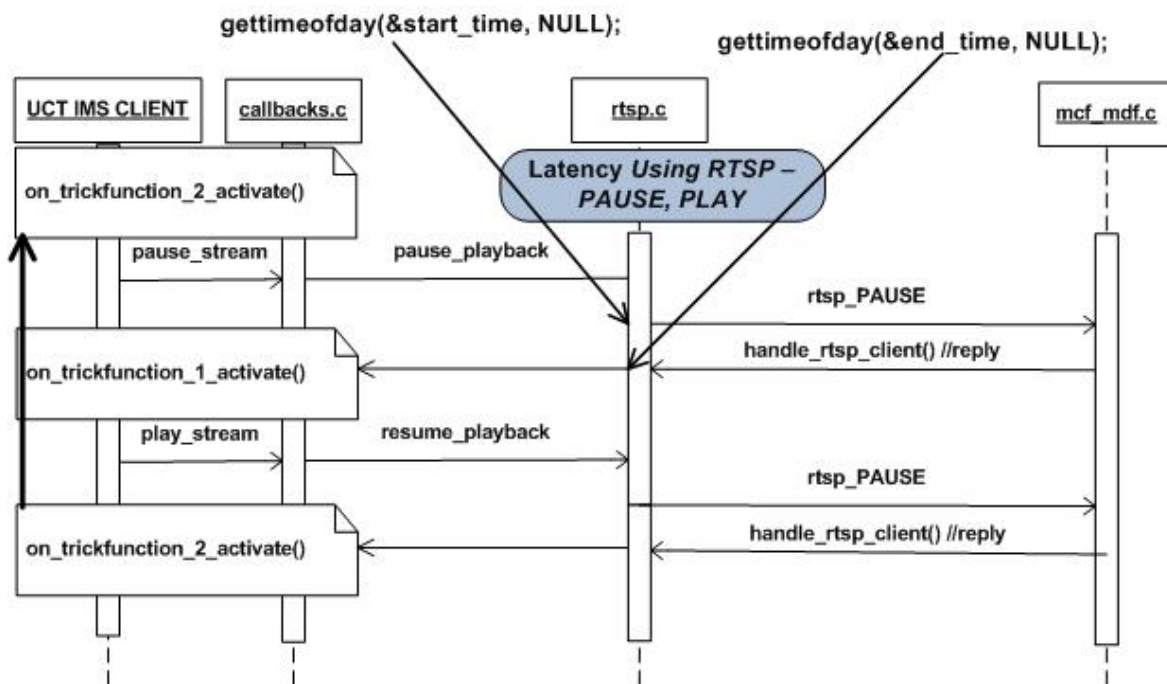


Figure 4.9: Measuring RTSP Latency

When the user presses 'pause' on the client interface to pause the video session, the corresponding `callback.c` function, `rtsp_PAUSE()`, is called and the 'pause' request is sent to the MCF. The MCF responds with '200 OK' and the `on_trickfunction_1_activate` is called after an RTSP PAUSE '200 OK' reply is received from the MCF. The `on_trickfunction_2_activate` function is similarly called after an RTSP PAUSE reply is received. This arrangement forms an automated loop for alternating RTSP 'play' and 'pause' requests over a period of time.

Similarly, Figure 4.10 shows how the SIP-based video interactivity latency was evaluated. Alternating PAUSE and PLAY requests were also sent. This was achieved by sending calling a `callback.c` function corresponding to the video interactivity function called. The `on_trickfunction_2_activate` is called after a `siptrick:play 200 OK` reply is received. This also creates a loop of alternating requests over a period of time.

²Note that these evaluation tests were not concerned with reduction of server processing load, but to ensure the reliability of latency results, this had to be considered. It must also be noted that these values (5 or 10 seconds) were decided on during the tests were done and were thus arbitrarily chosen.

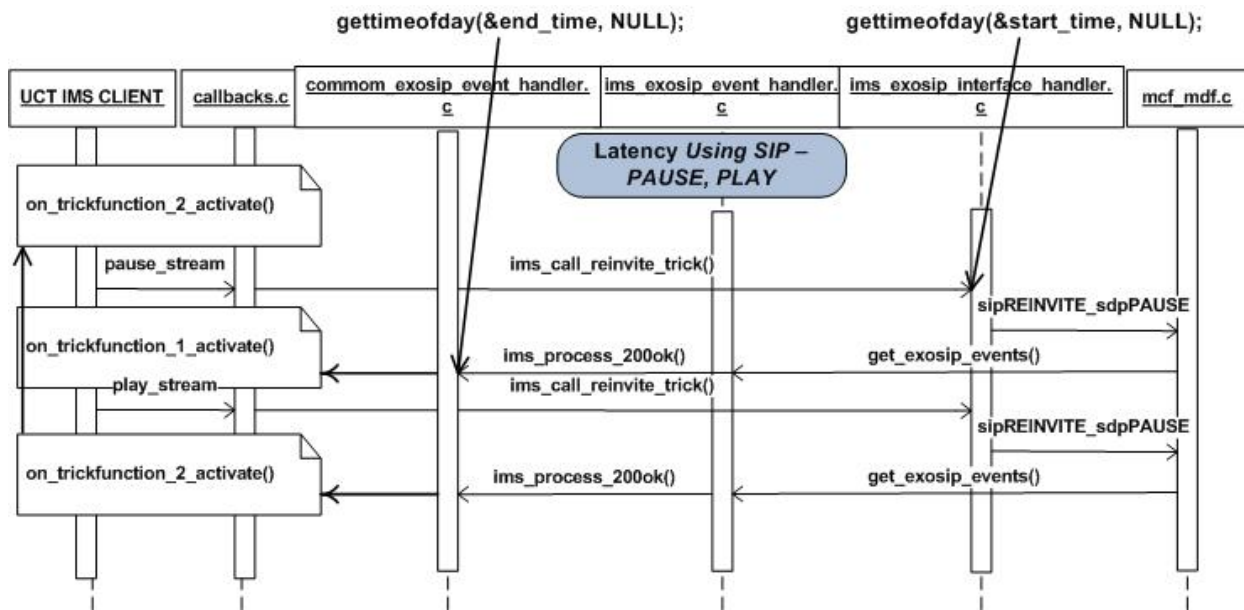


Figure 4.10: Measuring SIP Latency

4.7.2 Server Processing Load Measurement

In order to evaluate the validity of the proposed three-layered framework, a simple design that includes the activation and re-activation of the processing of video interactivity requests has been implemented. The implementation of the SCF-MCF includes an option of receiving video interactivity requests and not forwarding these requests to the MDF (or the Media Processing Layer-MPL).

The emulation of the TLVIF simplistically involves the enabling and disabling of sending requests sent to the MPL (at the MDF) based on random selection determined by a random number generator³. This effectively ensures that the video server handles a number of subscribers some of whom will have a full subscription to all the video interactivity profiles while others will have limited profiles.

Figure 4.11 shows the emulation framework used to implement profile-based video interactivity. The figure is similar to the one shown in Figure 3.9 in the previous design chapter. The only difference between the two architectures is the use of a random number generator to allow or reject certain requests instead of user subscription profiles.

The actual percentage of the requests that are processed is presented in the following chapter. The results will therefore not indicate the individual experience of each of the users (as per their service profiles) but only show the simulated impact of profile-based interactivity on the video server.

The overall result will thus be of limiting video interactivity requests with an aim of reducing the total load on the MDF. This effectively then achieves the objective of assessing the proposed hypothesis of: *If more interactivity requests lead to more processing load on the video server (i.e. the MDF), these requests need to be rationed (or limited) in order to avoid server over-loading.*

Given that both the MCF and the MDF are co-located on the same workstation, the processing load of these entities needed to be separated to show the contribution of each entity. This was achieved by measuring the processing load with and without the video (or MDF) processing. The

³Requests are forwarded or rejected depending on the value of the random number.

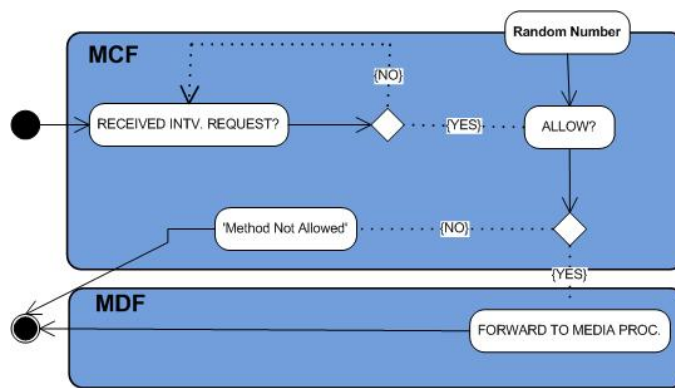


Figure 4.11: Interactive Layer Emulation

procedures followed and the results obtained are shown in the next chapter.

Dstat is a linux-based versatile resource statistics tool for measuring system parameters such as read/write bytes, number of active system processes, and number of sent and received bytes [65]. The Dstat tool was used to measure the server processing load on the video server.

An example output of the *Dstat* command is shown below:

```

"time","total cpu usage",,,,,,"dsk/total",,"net/total",,"memory usage",,,
"date/time","usr","sys","idl","wai","hiq","siq","read","writ","recv",
"send","used","buff","cach","free"
13-03 12:30:17,3.667,2.699,91.835,1.771,0.004,0.024,1411.656,64954.426,
0.0,0.0,117268480.0,149782528.0,210972672.0,49930240.0
  
```

The CPU processing usage (or video processing load) is measured using three variables in the *Dstat* output above: *usr*, *sys*, and *idl*. The *idl* variable indicates an 'idle' pool of processing resources. The typical and maximum value of this variable when there are no active system or user processes is 100. The *sys* variable indicates the amount of processing resources being used for system processes. The typical value of this variable when the video server is not running is 0. The *usr* variable indicates the amount of processing resources being used for user processes. The typical value of this variable is 0 when the server is not running. When the server is running at full load, processing a large number of video interactivity requests, the value of this variable ranges from 70 to 96. The total sum of the processes in the *sys*, *idl* and *usr* is approximately but not necessarily always equal to 100. Some of the processing power may be taken up by waiting queues for various processes in the system.

4.7.3 Bandwidth Measurements

The following tools were also used to perform bandwidth tests carried out in this study:

Iperf. Iperf⁴ is a tool used to measure bandwidth and the quality of the network link. The bandwidth is measured by sending several TCP packets that are then evaluated (e.g. round trip time) to calculate the quality of the network link.

Iperf can be used to measure bandwidth by typing the following command on a server ⁵:

⁴Iperf was developed was *The Iperf Team* and was initially released in April 2008.

⁵The *server* is normally arbitrarily chosen to represent the first station from which bandwidth is measured.

```
#iperf -s
```

To complete a link from the client workstation, the following command is entered:

```
#iperf -c SERVER_IP_ADDR
```

Where SERVER_IP_ADDR represents the server IP address [66], [67].

The transport layer bandwidth for WiMAX (using 'Iperf'), was measured between the Ingress WiMAX router and one of the clients on the SS IDU units shown in Figure 4.5 above.

The Ethernet transport layer bandwidth was measured between the SCF-MCF workstation and a client on the Ethernet network (10.128.0.0/16) shown in Figure 4.7 above.

The results of the bandwidth measurements are presented in the following chapter.

4.8 Chapter Summary

This chapter has primarily discussed the test-bed components and the software tools used for the evaluation. The evaluation platform developed in this chapter meets the evaluation method requirements stated earlier. The platform uses TISPAN compliant interfaces between the IMS core and the video server (MCF-MDF) to achieve compliance with IMS standards. The setting up of measuring latency and the server processing load using existing tools has been presented.

Chapter 5

Evaluation Results and Analysis

This chapter presents the results on video interactivity latencies for video interactivity requests using both RTSP and SIP in the IMS. These results will be compared with other current video interactivity designs by Sivasothy et al [2], and Khan et al[3]. The performance of the proposed three-layered video interactivity platform will also be assessed and compared with current designs. This chapter will also present an analysis, based on the results, on the relationship between video interactivity and the video server processing load (i.e. processing load at the MDF) to ascertain whether there is a reduction in the video processing load.

5.1 WiMAX and Ethernet-based Testbed Environments

The transport layer bandwidth of WiMAX and Ethernet networks were measured. The results of the bandwidth tests presented in Appendix C are summarised below.

Table 5.1 shows the bandwidth estimations at the transport layer level for WiMAX. As can be seen in the table, even though the WiMAX IP-CAN supports speeds of up to 12Mbps, the average transport layer bandwidth experienced in the *iperf* tests was approximately 8Mbps.

Average (Mbps)	7.96
Minimum (Mbps)	7.74
Maximum (Mbps)	8.54

Table 5.1: WiMax Bandwidth Link Estimation

Average (Mbps)	94.89
Minimum (Mbps)	94.00
Maximum (Mbps)	95.80

Table 5.2: Ethernet Bandwidth Link Estimation

Table 5.2 shows the transport layer bandwidth estimations on the Ethernet IP-CAN, and even though the Ethernet IP-CAN supports speeds of up to 100Mbps, the actual effective bandwidth experienced through the *iperf* tests was approximately 95Mbps.

These measured speeds will be used during the latency tests and video processing tests performed throughout this chapter.

5.2 Evaluation of Video Interactivity Latencies by RTSP and SIP

This section will evaluate video interactivity latencies experienced by both RTSP and SIP. In the subsections below, two tests will be performed:

- Latencies by SIP and RTSP without MDF processing (i.e. without the $l_{videoServerProcessing}$ component as per Equation 1.1)
- Latencies by SIP and RTSP with MDF processing (i.e. total latency as per Equation 1.1)

5.2.1 Interactivity Delays by RTSP and SIP protocols - Without MDF Processing

RTSP-based Video Interactivity Results

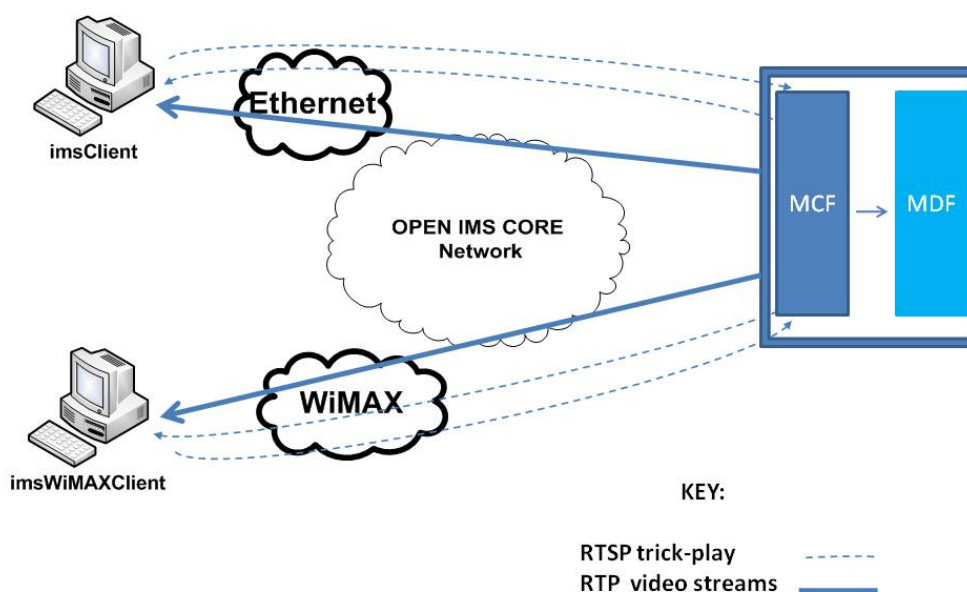


Figure 5.1: RTSP Tests Diagram

Figure 5.1 depicts an overview of the tests that were carried out to measure RTSP latencies¹. About 1000 RTSP requests were sent to the video server. The complete results of the captured RTSP latencies are provided in the accompanying CD-ROM of this document.

¹All video interactivity latencies in this chapter will henceforth simply be referred to as 'latencies'.

Table 5.3 shows the average, maximum and minimum latencies of RTSP requests. The variance of various requests' latency is also shown.

Network Link	Average Latency (s)	MAX Latency (s)	MIN Latency (s)	Variance
WiMAX	0.0376	0.4190	0.0270	0.0011
Ethernet	0.0022	0.0740	0.0000	0.0001

Table 5.3: RTSP Latencies Without MDF

The results show some fluctuations in the latencies captured, in some cases showing a difference of a magnitude of 10. The actual statistical variance is small. There is also a significant difference between WiMAX and Ethernet values. The magnitude of the difference of latencies over WiMAX and Ethernet are shown in Table 5.4. The table shows the percentage change² of the latencies experienced over both networks.

	WiMAX-Ethernet Delay Difference(s)	WiMAX-Ethernet Delay Percentage Change
Min Delay	0.03	100
Max Delay	0.35	82.34
Average Delay	0.04	94.08

Table 5.4: Network Delay Percentage Change for RTSP-based Video Interactivity

The table shows a significant percentage change of over 80 percent. This can be accounted for by the large difference in network transmission speeds. The significant difference in the latency values therefore means that the transmission speed of the RTSP requests significantly contributed to the total latency.

The results above show RTSP latency without video processing (i.e. the MDF excluded). The following subsection will assess SIP latencies and compare them to that of RTSP.

SIP-based Video Interactivity Results

Figure 5.2 depicts an overview of the tests that were carried out to measure SIP latencies. SIP-based video interactivity tests were performed over both WiMAX and Ethernet over the same bandwidth values (explained in Section 5.1 above). It must be noted that the latencies shown in the tables (for RTSP and SIP latencies, without MDF processing) are for both PAUSE and PLAY requests. The difference between the processing of PAUSE and PLAY requests is at this stage insignificant as both requests are packaged and processed in the same manner without the video processing.

Table 5.5 shows WiMAX and Ethernet video interactivity latencies for SIP. SIP latencies have a wide range from 0.285 seconds to 1.84 seconds over both networks even though the variance values show a small statistical variation of the measured values for each network.

The percentage change between delays over WiMAX and Ethernet are shown in Table 5.6. On average SIP WiMAX latencies are marginally larger than Ethernet latencies. Interestingly, this is in

²PercentageChange = $(WiMAX_{latency} - Ethernet_{latency}) / Ethernet_{latency} * 100$

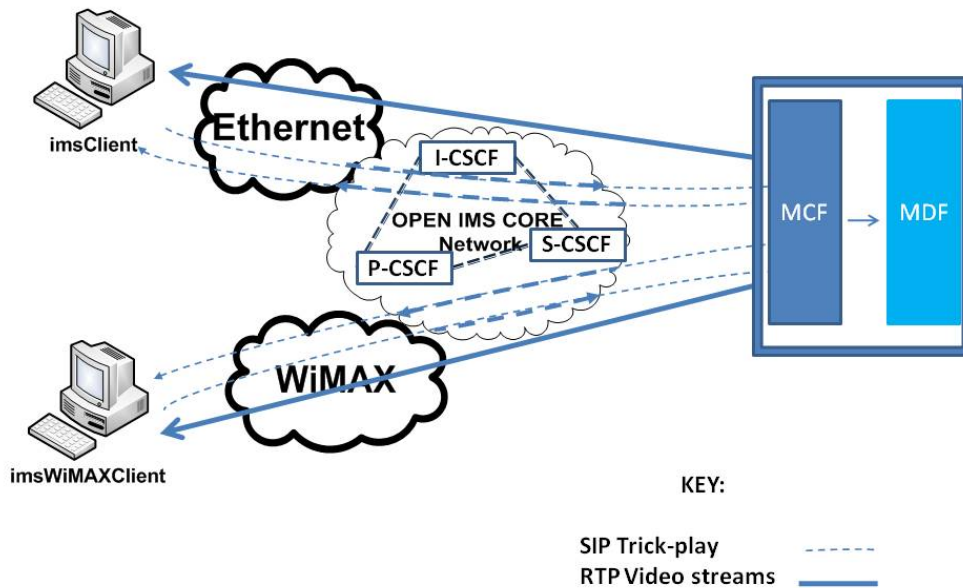


Figure 5.2: SIP Tests Diagram

Network Link	Average Delay (s)	MAX Delay (s)	MIN Delay (s)	Variance
WiMAX	0.4605	1.8420	0.3400	0.0044
Ethernet	0.4076	1.6060	0.2850	0.0023

Table 5.5: SIP Latencies Without MDF on WiMAX/Ethernet

contrast to the RTSP latency network percentage change values shown above. This is explained by the fact that SIP traverses more entities for each signalling message. This results in the processing latency of each of the entities adding to the total delay. The lower proportional contribution of the transmission latency is seen in the small percentage change between the two networks.

Table 5.7 shows the percentage change³ of latencies from RTSP to SIP over WiMAX.

The percentage change from RTSP latency to SIP latency over WiMAX ranges from 110 percent to over 400 percent. This translates to SIP latency being more than two times larger than RTSP latency. As stated throughout this study the aim of the evaluation platform was to provide a comparative study of RTSP and SIP latencies. From the table and the results above, although SIP latency can be less than 0.5 seconds, it clear that SIP latency is still significantly larger than the RTSP latency.

Table 5.8 shows the latency differences as well as the latency percentage change from RTSP to SIP over Ethernet. The percentage change of the protocols over Ethernet is well over 1000. It is clear that as the transmission speed increases, the RTSP latencies are reduced where as the SIP latencies do not show any major improvements (as also highlighted in the SIP WiMAX-Ethernet percentage change values above). This implies that for SIP-based video interactivity, the

³PercentageChange = $(SIP_{latency} - RTSP_{latency}) / RTSP_{latency} * 100$

	WiMAX-Ethernet Delay Difference(s)	WiMAX-Ethernet Delay Percentage Change-%
Min Delay	0.06	16.18
Max Delay	0.24	12.18
Average Delay	0.05	11.49

Table 5.6: Network Delay Percentage Change for SIP-based Video Interactivity

	SIP-RTSP Latency Difference(s)	SIP-RTSP Latency Percentage Change-%
Min Delay	0.313	115.93
Max Delay	1.798	429
Average Delay	0.423	112.61

Table 5.7: Latency Percentage Change for RTSP vs. SIP based Video Interactivity over WiMAX

bottleneck lies primarily on the number of network processing elements traversed.

	SIP-RTSP Latency Difference(s)	SIP-RTSP Latency Percentage Change-%
Min Delay	0.285	Undef
Max Delay	1.532	2070.3
Average Delay	0.4048	18400

Table 5.8: Latency Percentage Change for RTSP vs. SIP based Video Interactivity over Ethernet

The proposed design did not include a translation proxy (as proposed in Khan [3]) that converts RTSP-based requests to SIP signalling message. The introduction of such a proxy is clearly untenable as it would introduce even more delays (i.e. by increasing the number of SIP processing entities).

5.2.2 Interactivity Delays by RTSP and SIP protocols - With MDF Processing

Figure 5.3 depicts an overview of the tests that were carried out to measure SIP latencies with video processing enabled (i.e. MDF included).

The next experiment assesses the delay introduced by video processing for video interactivity. Although SIP was used for these tests, it must however be recognised that, according to the evaluation platform design, the latency introduced by video processing is essentially independent of the protocol being used. Since the network dependent results have already been assessed above, in the following test only the Ethernet based tests will be considered.

The video processing primarily entails changing the video stream to reflect the requested action (e.g. PAUSE or FORWARD). The procedural call to the MDF that was disabled above is now enabled in this test.

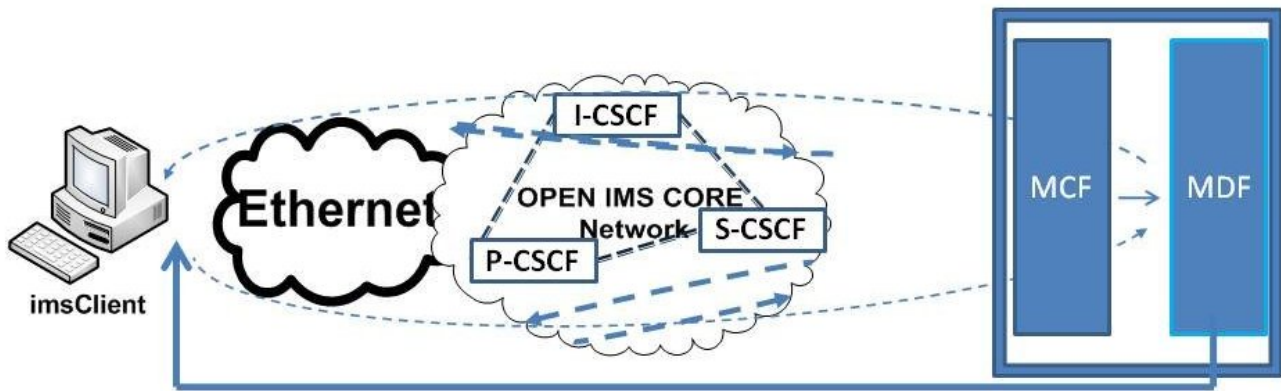


Figure 5.3: SIP With MDF Test Diagram

V.I. Function	Average Delay (s)	MAX Delay (s)	MIN Delay (s)	Variance
PLAY	0.405	0.407	0.403	0.000
PAUSE	0.393	0.430	0.243	0.002
FORWARD	0.370	0.406	0.202	0.005
REWIND	0.403	0.406	0.402	0.000

Table 5.9: SIP Latencies With the MDF on Ethernet

Where as over 1000 tests were performed for each of the interactivity latencies without the MDF processing above, only 10 tests were performed for each of the interactivity requests with MDF processing. Due to the limited video length, requests such as forward and rewind could only be processed a limited number of times for each video session. The results in Table 5.9 show that, on average, 'forward' latency is the lowest. The highest video interactivity latency is produced on the PLAY request. Given that the actual number of processing instructions for FORWARD at the MDF is higher (i.e. the current position is determined and the playback rate changed) than the number of processing instructions for PLAY (i.e. the video pipeline is simply set to PLAYING state), the averages in this table leads one to conclude that the video processing does not significantly contribute to the total latency of each request.

The highest maximum latency value shown in the table is for PAUSE, which is significantly higher than the average of 0.407 without video processing (shown in Table 5.5). This is still not sufficient to indicate the actual contribution of video processing. Since the actual processing primarily involved a simple instruction to the gstreamer video stream pipeline to set the streaming state to 'PAUSED', it is not entirely surprising that the video processing latency is negligible and not significant.

The anomaly apparent in the above table may also be explained by the fact that the SIP average latency in Table 5.5 above is affected by some unusually high latency values (e.g. 1.6s, shown in Table 5.5). The video content used for these tests is listed in Appendix C. Full presentation of the results can be found in the accompanying CD-ROM.

5.3 Evaluation of Video Interactivity and Video Server Load

This section tests the relationship between the video interactivity signalling rate and the video server processing load. The proposed Three-Layered Video Interactivity Framework (TLVIF) will be tested in the following section. The results below will incrementally show the contributions of the signalling, the video processing load as well as the impact of the TLVIF on the processing load. The following questions will summarily be answered in this section:

- How does the MCF processing load increase as the number of signalling requests' rate increase?
- Regarding the relationship between the MCF and the MDF, does the number of signalling requests processed impact significantly on the MDF processing load?

The processing loads were measured, using the *Dstat* program introduced in the previous chapter, at the rates of 0.5, 1, 2.5 and 5 video interactivity requests (or calls) per second (i.e. calls/s). For 0.5 calls per second, 1 client sent a request every 2 seconds; for the 1 call/s rate, a client sent a request every one second; for the rate of 2.5 calls/s, one video client was used to send requests every 0.4s. The 5 calls/s rate was estimated by having two video clients sending requests every 0.4s, thus effectively increasing the percentage of received requests at the video server by a factor of two (i.e. approximately 5 calls/s)⁴. For many of the processing load tests below, the *Dstat* program was allowed to run for at least 10 minutes⁵, and the load samples were captured every second.

5.3.1 Signalling Processing Overhead

Table 5.10 shows the averages of the processing load with nothing running; a server running, with no sessions established; with one client sending signalling requests; with two clients sending signalling requests at various rates. The most important column is the 'usr' (user) column as it indicates the processing time allocated to user processes. This indicates the active video server processing load. The 'sys' (system) load and the 'idl' (idle) loads are shown here for completeness. The load in the 'Average sys load' column shows the system load fluctuating from low 1.71 to 14.98 and down to 6.28 as the call rate increases. This is due to various system processes that are triggered in the background and do not seem to have any correlation to the running server which is running on 'usr' processing mode. A fourth column, the 'Average wai (waiting) load' is also shown. This column indicates the processing time spent in the system queues⁶. The results

⁴Most of these call rates were an approximation based on the assumption that each request took 0.4s to be processed, as per results in the preceding section.

⁵This is with exception to the video processing and TLVIF tests presented below, most of which could not be taken over a long period due to limited session lengths of various videos used as well as some software bugs detected during the tests.

⁶A full description of the how *Dstat* is provided in Section 4.7.3 in the previous chapter.

in the table are presented primarily to indicate that the *Dstat* program as well as the evaluation platform are functioning as expected.

Running Processes	Average usr Load	Average sys Load	Average idl Load	Average wai Load
Nothing Running	1.57	1.71	96.81	0
Server Running	3.48	0.27	95.25	0
Signalling - 0.5 calls/s	39.8	14.97	44.16	0.22
Signalling - 1 call/s	41.03	14.98	0	39.1
Signalling - 2.5 calls/s	62.61	10.45	0	25.6
Signalling - 5 calls/s	78.52	6.28	10.5	4.45

Table 5.10: Processing Load Averages With and Without the Signalling Load

As can be seen in the table, the server processing load (i.e. MCF) increases gradually as the number of video interactivity requests (without video processing) is increased. The processing load increased marginally from 0.5 to 1 calls/s rate. For completeness, some of the processing time taken up by the waiting queues for 1 calls/s and for 2.5 calls/s is also shown. This waiting queues load does not seem to have a direct correlation with the increase in the user processing load as it decreased at the 5 calls/s rate. For these reasons, in the discussions below the 'wai' load will be ignored as it is mostly negligible.

Figure 5.4 graphically shows the growing CPU load as the number of clients (and the call rate) is increased.



Figure 5.4: Graph showing CPU Load for With and Without the Signalling Load

The results of the server processing load with and without the signalling in the figure show that the evaluation platform can be reliably used to further investigate the research questions posed in this study.

5.3.2 MDF Processing Overhead

The following tests show the server processing load with video processing enabled. The MDF processing tests were mostly captured for over 3 minutes. Table 5.11 shows the values of the server processing load at 0.5, 1, 2.5 and 5 calls (or video interactivity requests) per second.

Running Processes	Average usr Load	Average sys Load	Average idl Load
Video Processing - 0.5 calls/s	62.66	15.52	7.16
Video Processing - 1 call/s	62.85	12.75	16.2
Video Processing - 2.5 calls/s	71.97	7.23	19.95
Video Processing - 5 calls/s	x	x	x

Table 5.11: Processing Load Averages with Processing

There is a marginal difference between the processing loads for 0.5 and 1 calls/s rates. A significant increase is seen from 1 to 2.5 calls/s. The video processing load at 5 calls/s was not obtained as the video server crashed at this point, most likely due to video server over-loading and a software bug.

Figure 5.5 graphically shows the increasing CPU load of the video server with video processing load at different call rates. The signalling loads are also indicated in the figure.

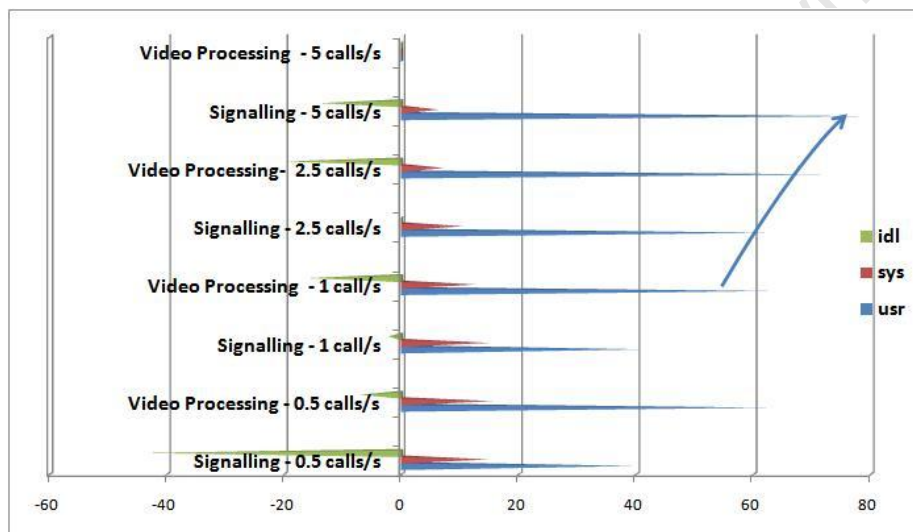


Figure 5.5: Graph showing CPU Load for Signalling Only vs. with Video Processing

The figure demonstrates that the difference of the signalling to the video processing loads is much larger at the lower call rates.

5.3.3 SIP-based Interactivity vs. Video Server Load Relationship

As discussed in Section 2.2.2, the relationship of many video interactivity designs is such that, as observed above, the increase in the number of processing requests leads to an increase in the server processing load. Many RTSP-based VoD designs follow this performance model. Because of direct interactivity communication between the client and the media server, most RTSP-based servers are prone to experiencing performance bottlenecks when many video clients simultaneously send requests such as forward, rewind and fast-forward.

Table 5.12 shows the processing load percentage change from video server processing with signalling only (i.e. MCF only) to server processing with both signalling and video processing (i.e. both MCF and MDF are enabled). Depending on the frequency of requests received at the video

Average Load with/out Video Processing	Processing Load Difference	Processing Load Percentage Change-%
At 0.5 calls per second	22.86	57.44
At 1 call per second	21.82	53.18
At 2.5 calls per second	9.36	14.95

Table 5.12: Processing Load Percentage Change With and Without Video Processing

Average Load with/out Video Processing	Processing Load Difference	Processing Load Percentage Change-%
From 0.5 to 1 call per second	0.19	0.3
From 1 to 2.5 calls per second	21.82	14.51

Table 5.13: Processing Load Percentage Change From Different Call Rates

server, it is clear that video processing introduces a significant load, varying from 14.95 percent to close to 60 percent to the total video server processing load.

Table 5.13 shows the percentage change of the video processing load from the results obtained in Table 5.11. A small percentage change from 0.5 to 1 call per second is realised. But from 1 to 2.5 calls per second, the percentage change is larger at 14.5 percent. So far, only one client is being used to track the processing load with video processing. The percentage change shown here indicates an increasing processing load as the call rate is increased.

The results above conclusively show that the video processing contributed significantly to the total video server processing load. The reduction of the video processing component of the total video server processing load, by the proposed TLVIF, is examined in the following section.

5.4 Performance Evaluation of Emulated TLVIF

This section answers the following question: *Regarding the effectiveness of the proposed TLVIF, does the reduced number of signalling requests forwarded from the MCF to the MDF significantly reduce the total video server (MCF + MDF) processing load?*

In order to evaluate the effectiveness of the proposed framework, the following scenarios will be considered:

- *Scenario 1.* The performance of the video server with video streaming enabled but no video interactivity processing allowed on the media server will be evaluated. Clients will be enabled to send interactivity signalling requests, but no video processing is enabled for the requests. This scenario therefore primarily consists of the load.
- *Scenario 2.* The performance of the video server with interactivity processing allowed on the video streaming. This is to assess the impact that video interactivity processing has on the server load. This scenario therefore comprises the total video server processing load as defined in Equation 1.2.
- *Scenario 3.* The performance of the video server on the Three Layered Video Interactivity Framework (TLVIF) is also investigated. This does not present a full evaluation of the

TLVIF. This scenario only assesses the possible impact of selective video interactivity processing by randomly enabling or disabling video processing for captured interactivity requests⁷. This scenario therefore comprises the total video server processing load when the TLVIF is implemented.

5.4.1 TLVIF QoE

Before assessing the performance of the TLVIF vis-a-vis the video processing load and the signalling load, the Quality of Experience (QoE) of various requests that were rejected and allowed to be processed at different call rates is presented below. QoE here represents a percentage of requests that are actually processed.

The tables below will show the QoE for requests received at the video server at rates of 0.5 calls/s, 2.5 calls/s and 5 calls/s. The results for 1 calls/s are not presented as they could not reliably be obtained.

Table 5.14 shows the percentage of processed requests for each interactivity function at 0.5 calls/s. This table shows that PLAY was allowed almost 50 percent of the time. Other requests were only allowed at a percentage of less than 45 percent.

	PLAY	PAUSE	FORWARD	REWIND
Total Number of Calls	30	31	30	30
No of processed requests	14	12	13	9
No of non-processed requests	16	19	17	21
QoE Percentange -%	47	39	43	30

Table 5.14: QoE Performance of TLVIF at 0.5 calls per second

Table 5.15 shows the percentage of processed requests for each interactivity function at 2.5 calls/s. The table shows that PAUSE and REWIND were allowed at a percentage less than 30 percent and only PLAY and FORWARD were above 30 but below 40 percent.

	PLAY	PAUSE	FORWARD	REWIND
Total Number of Calls	21	21	21	21
No of processed requests	8	3	7	6
No of non-processed requests	13	18	14	15
QoE Percentange -%	38	14	33	28

Table 5.15: QoE Performance of TLVIF at 2.5 calls per second

Table 5.16 shows the percentage of processed requests for each interactivity function at 5 calls/s. The table shows that most video interactivity functions were processed about a third of the time with the exception of PAUSE which was processed almost half of the time.

⁷The aim of the Three Layered Video Interactivity Framework (TLVIF) is to enable/disable certain functions (especially those that are demanding such as FORWARD and REWIND) depending on the user subscription (enabled through the use of the Service Management Layer). This investigation only holds for the case when some subscribers may choose not to subscribe to all video interactivity requests. In the case where all the subscribers have full interactivity, this scenario will become equivalent to scenario 2.

	PLAY	PAUSE	FORWARD	REWIND
Total Number of Calls	60	61	60	60
No of processed requests	18	28	22	18
No of non-processed requests	42	33	38	42
QoE Percentange -% ⁸	30	46	37	30

Table 5.16: QoE Performance of TLVIF at 5 calls per second

The overall QoE percentages above were captured during the tests of the TLVIF. The following subsection will now present the results of the performance of the TLVIF (as defined by the QoE percentages above).

5.4.2 TLVIF Performance

For the measurement of the processing loads for TLVIF, tests were allowed to run for as long as the program (and the video being played) allowed. To increase the reliability of the results, several tests were taken at each call rate. Out of several tests taken at each call rate for TLVIF load measurements, the most consistent output which provided the longer processing load output (before the server crashed) was used in the results. The reason for the server crash has been attributed to a software bug. The tests for 0.5 calls/s and 5 calls/s ran for 2 minutes. The tests for 1 calls/s and 2.5 calls/s rates only ran for under a minute. The 1 calls/s rate results were not used as the server crashed within a second several times the test was attempted. The full presentation of these results can be found in the accompanying CD-ROM (see Appendix E for the folders that contain various test results). Table 5.17 shows the averages of the TLVIF processing loads at 0.5, 2.5 and 5 calls/s.

Running Processes	Average usr Load	Average sys Load	Average idl Load
TLVIF - 0.5 calls/s	56.49	7.4	15.4
TLVIF - 1 call/s	-	-	-
TLVIF - 2.5 calls/s	66.5	33	0
TLVIF - 5 calls/s	91.25	8.06	0

Table 5.17: Processing Load Averages with Video Processing

The objective of the proposed TLVIF was to reduce the total video server processing load. Compared to the video processing load results in the previous section, the results above seem to indicate that the proposed TLVIF achieved this objective. Table 5.18 shows the processing load percentage change from clients with video processing enabled to clients with the TLVIF. Only the 0.5 and 2.5 calls/s results are shown as the server crashed for video processing at 5 calls/s and for TLVIF at 1 call/s. It can be observed from the table that the TLVIF clearly provides a saving to the total server processing load. This has been achieved by rationing requests sent to the MDF for processing (based on the simulation of various clients' service profiles).

The saving indicated on the table is almost 10 percent for 0.5 calls/s. The percentage of requests processed at this rate (according to Table 5.14 above) is above 30 percent. The saving at 2.5 calls/s rate is similarly close to 10 percent and the percentage of requests processed (according

to Table 5.15 above) is also about 30 percent (except for PAUSE at 14 percent). These results show that the contribution of the video processing load to the video server load is significant and can be reduced by other functions, that are less processor consuming, that map the user requests to a certain criteria (in this case a random number simulating user profiles) for either allowing or disallowing the requests.

Average Load with/out TLVIF	Processing Load Difference	Processing Load Percentage Change-%
At 0.5 calls per second	-6.7	-9.85
At 2.5 calls per second	-5.47	-7.6

Table 5.18: Processing Load Percentage Change With and Without TLVIF cf. Video Processing

Another observation concerns the contribution of the TLVIF processing load to the signalling only load. Figure 5.6 shows the measured CPU load of the video server comparing the TLVIF load with the signalling load.

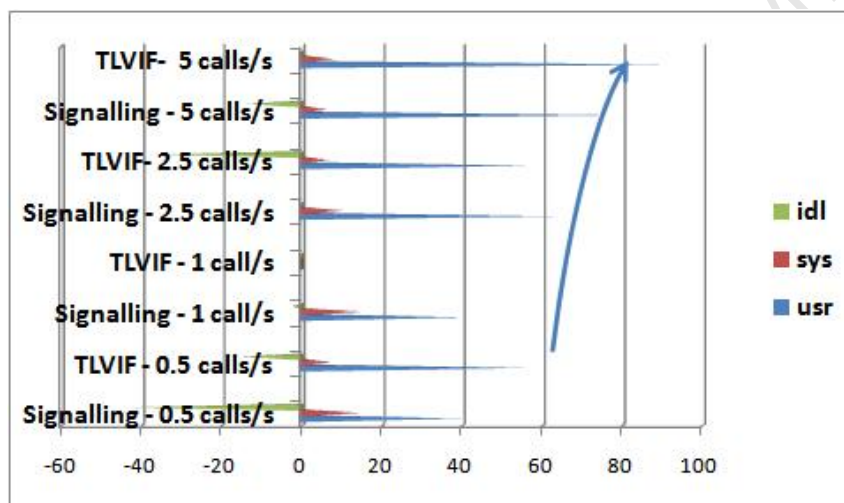


Figure 5.6: The Server Processing Load With and Without TLVIF Graph cf. Signalling Only

Table 5.19 shows the processing load percentage change from clients with signalling only to clients with TLVIF. It is clear from the figure above and the table that, on average, the TLVIF consumes more load than signalling only. This is an obvious observation, as we expect more processing to be incurred by the TLVIF as well as by the MDF processing for some requests that are allowed to be executed. At 0.5 calls/s, a larger increase of the processing load with TLVIF relative to signalling only is observed. This can be attributed to the larger percentage of requests allowed at this particular call rate. In the 2.5 calls/s case, there is the smallest increase in the processing load with TLVIF. This can similarly be attributed to a smaller percentage of requests allowed at this particular call rate.

The results above show that with the TLVIF, the video server processing load can be reduced. The TLVIF provides full interactivity on the video streams, while allowing clients to subscribe to various service profiles. Users who only subscribe to certain requests, the QoE will be to the level of user expectations. This is an improvement on the Internet-based mechanisms that aim to reduce the load directed at the video server (by downloading video locally to the clients), as full

Average Signalling Load with/out TLVIF	Processing Load Difference	Processing Load Percentage Change-%
At 0.5 calls per second	16.69	41.93
At 2.5 calls per second	3.89	6.21
At 5 calls per second	9.36	14.95

Table 5.19: Processing Load Percentage Change With and Without TLVIF cf. Signalling Only

live interactivity - which allows features such as video adaptation - is now possible with the TLVIF whilst also reducing the video server processing load.

5.5 Chapter Summary

This chapter has presented the results for video interactivity delays using RTSP and SIP. The tests aimed to provide a quantifiable percentage difference of SIP latency compared to RTSP latency. The results show that, over WiMAX, the SIP latency is more than twice larger than RTSP latency. The results also showed that over networks with higher transmission speeds, the percentage change of the latencies from RTSP to SIP is significantly larger at 18400 percent. Although less than 1 second, the SIP video interactivity latency is therefore largely untenable compared to RTSP. The results have also shown that, depending on the video server design, the contribution of the video processing to the overall latency is negligible.

The results presented have shown that even though video processing adds to the processing load, the percentage of the contribution to the processing load due to video processing will largely be affected by the video server design. The proposed TLVIF aimed to reduce the processing load by limiting requests sent to the MDF. The test results showed that with the different QoE levels, the TLVIF reduced the video server processing load by up to 10 percent.

Chapter 6

Conclusions and Future Work

6.1 Summary

This study aimed to assess the video interactivity latency for SIP compared to RTSP. A discussion on the current literature on interactivity in current IPTV systems (i.e. over the Internet) shows that various NGN platforms, particularly the IMS, are being adopted for IPTV. The IMS uses SIP for controlling sessions, yet RTSP is the most widely deployed protocol for interactivity with video streams. The SIP-based video interactivity architecture by Sivasothy et al [2] was discussed and it was found to be requiring a major redesign of the SIP protocol. Other designs, such as the ones proposed by Khan et al [3] also have deficiencies as they introduce extra processing elements on the network leading to increased latency.

This study also investigated the relationship between video interactivity signalling and the video server processing load. There is currently insufficient published research that provides a comprehensive analysis on this discussion. Nahum et al [36] have only provided preliminary results on the impact of transport protocols (i.e. UDP, TCP) and authentication requests on control protocols such as SIP.

Various internet-based techniques were discussed by looking at how current bandwidth scalability mechanisms indirectly reduce the actual interactivity processing load on the video server by downloading videos to users. These approaches were however found to be inadequate as they reduce the user's live interaction with streaming content.

This study investigated the relationship between video interactivity and the video server processing load on the IMS testbed using the Fraunhofer Fokus Open IMS Core, the UCT IMS CLIENT, a modified UCT IPTv server, and tools such as the Dstat.

A discussion on how video interactivity protocols, such as SIP and RTSP, can be leveraged to provide a framework for the development of future interactivity requests has been provided. Research shows that the most important service requirement in IPTV services is interactivity. This led to the proposed Three Layered Video Interactivity Framework (TLVIF), which aims to reduce SIP video interactivity latencies as well as to leverage features of the SIP protocol to allow for rapid development of video interactivity requests.

6.2 Conclusions

The proposed SIP-based platform has managed to achieve a SIP latency more than double the RTSP latency. In a higher speed network, the relative SIP latency is significantly larger than RTSP latency (i.e. up 18400 percent more than RTSP latency on average). The increased SIP latency is due to a greater number of processing entities SIP messages traverse. The proposed design did not include a translation proxy (as proposed in Khan et al [3]) that converts RTSP-based requests to SIP signalling message. It is clear that the introduction of such a proxy is untenable as it would introduce even more delays (i.e. by increasing the number of SIP processing entities). This study offers one of the first contributions to the evaluation of SIP-based video interactivity designs that will further reduce the latency. The study's main contribution is the quantification of SIP latency relative to RTSP latency in the IMS context.

The results on the investigation of the relationship between video interactivity and video server processing load showed that the video server processing load increases substantially when the video (or MDF) processing is added. This means that the reduction of video interactivity processing from each user leads to a significant reduction in the video server processing load. The significance of this reduction in the video processing load from the same amount of users is that more users could potentially be supported on the same server system.

In the proposed TLVIF, client profiles that allow subscribers to choose which and how many video interactivity requests they would like to subscribe to were introduced. These profiles aimed to maintain the level of user service expectations while also reducing the video server processing load. The results of the performance evaluations of the TLVIF showed a reduction in the video server processing load. Although this approach may be reducing the number of requests allowed, the user is guaranteed a certain level of experience that conforms to the subscribed profile.

The conclusions of the research by Shin [43] stipulate that future video interactivity experience will be enhanced largely by the user customised requirements than by the amount of interactivity allowed. This means that even though the amount of video interactivity requests processed has been reduced, the proposed TLVIF will not necessarily reduce the overall user service experience.

6.3 Recommendations and Future Work

The proposed TLVIF was not fully implemented in this study. Further development on the implementation and thorough evaluation of TLVIF can be done. This development will involve the complete implementation of the Service Control Function (SCF), the Media Control Function (MCF) and the Media Distribution Function (MDF) implemented on separate workstations. The implementation would also include an enhanced profiling solution implemented on the Home Subscriber Server (HSS) as well as at the SCF that includes the new profile items introduced by the proposed TLVIF. A further evaluation of the proposed TLVIF on a completed test-bed platform may provide more insight on the effectiveness and performance of the TLVIF. These developments can be used to further test the functional effectiveness of the mapping framework as well to quantify the contribution of the mapping to the total video server processing load.

Further research could also be done to investigate whether the proposed TLVIF leads to better video server scalability. Although this research has shown how the video server processing load can be reduced with the same amount of users, more tests could be performed on the TLVIF to ascertain whether it leads to support for more users.

The role of SIP in video interactivity has scope for future research. The SIP latencies measured in this study were very large compared to RTSP latencies. These can be reduced by more efficient SIP designs that would take advantage of the flexibility of the SIP protocol. An investigation of other methods that would involve either reducing the number processing entities traversed for video interactivity or the processing time at each of the entities could be done.

The evaluation of SIP video interactivity in this study was only compared with RTSP. Other designs that are proposed in the literature could also be evaluated on the same test-bed platform and compared with the proposed solution. The research of the various SIP-based interactivity latencies will provide a more comprehensive analysis of how the different implementations can affect the incurred latency and ultimately, the user service experience.

University of Cape Town

Bibliography

- [1] TISPAN, *Draft ETSI RTS 182 027 V3.3.0 IPTV Architecture; IPTV functions supported by the IMS subsystem*. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), 2009-08.
- [2] S. Sivasothy, G. Lee, and N. Crespi, "A Unified Session Control Protocol for IPTV Services,"
- [3] S. Khan, R. Gaglianella, and M. Luna, "Experiences with blending HTTP, RTSP, and IMS [IP Multimedia Systems (IMS) Infrastructure and Services]," vol. 45, pp. 122–128, 2007.
- [4] TISPAN, *TS 183 063 V2.1.0 IMS-based IPTV stage 3 specification*. Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), 2008-06.
- [5] Fokus, "Open ims core <http://www.openimscore.org/>," 2009. [Online; Accessed 9th-January-2009].
- [6] W. I. Stats, "World internet users and population stats <http://www.internetworldstats.com/stats.htm>," 2009. [Online; Accessed 30th September 2009].
- [7] L. Press, "The internet and interactive television," *Commun. ACM*, vol. 36, no. 12, pp. 19–23, 1993.
- [8] ABC, "Abc website <http://www.abc.com/>," 2009. [Online; Accessed 07 December 2009].
- [9] NBC, "Nbc website <http://www.nbc.com/video/>," 2009. [Online; Accessed 07 December 2009].
- [10] CBS, "Cbs website <http://www.cbs.com/>," 2009. [Online; Accessed 07 December 2009].
- [11] HBO, "Hbo on demand website <http://www.hboondemand.com/>," 2009. [Online; Accessed 07 December 2009].
- [12] D. on Demand, "Dstv Broadband <http://www.dstvbroadband.co.za/>," 2009. [Online; Accessed 07 December 2009].
- [13] Betanews, "Mobile iptv experiment in korea <http://www.betanews.com/article/Mobile-IPTV-Experiment-in-Korea-Could-Compete-with-DVBH-MediaFLO/1171491160>," 2009. [Online; Accessed 05th October 2009].
- [14] G. O'DRISCOLL, *Next Generation IPTV Services and Technologies*. 2008.
- [15] E. Prosperetti, G. Tripaldi, and V. Comandini, "IPTV missed expectations. Can regulation do the trick?,"
- [16] YouTube, "Youtube-broadcast <http://www.youtube.com/>," 2010. [Online; Accessed 24th-March-2010].

- [17] Joost, "Joost video website <http://www.joost.com/>," 2011. [Online; Accessed 27 January 2011].
- [18] V. Ozianyi and N. Ventura, "Efficient and Scalable Charging systems for IP Multimedia Networks,"
- [19] W. Sincoskie, "Video on demand: Is it feasible?," in *Global Telecommunications Conference, 1990, and Exhibition. 'Communications: Connecting the Future', GLOBECOM'90., IEEE*, pp. 201–205, IEEE, 2002.
- [20] EHOW, "How does satellite tv picture quality compare to cable http://www.ehow.com/how-does_4673250_tv-picture-quality-compare-cable.html," 2011. [Online; Accessed 21 January 2011].
- [21] I. Más, V. Berggren, R. Jana, J. Murray, and C. Rice, "IMS-TV: An IMS-based architecture for interactive, personalised IPTV," *Communications Magazine, IEE*, vol. 46, no. 11, pp. 156–163, 2008.
- [22] Google, "Google videos <http://video.google.com/>," 2010. [Online; Accessed 24th-March-2010].
- [23] A. Agarwal, "Digital inspiration website <http://www.labnol.org/internet/video/the-success-of-web-video-depends-on-quality-and-content/1086/>," 2009. [Online; Accessed 27 January 2011].
- [24] V. Li, W. Liao, X. Qiu, and E. Wong, "Performance model of interactive video-on-demand systems," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1099–1109, 1996.
- [25] R. et al, *SIP: Session Initiation Protocol RFC 3261*. 2002-06.
- [26] IETF, *Media Playback Control Protocol Requirements - IETF Draft <http://tools.ietf.org/>*. 2008.
- [27] J. F. K. D. T. Jayanta K. Dey-Sircar, James D. Salehi, "Providing VCR Capabilities in Large-Scale Video Servers," 07 1994.
- [28] S. Carter and D. Long, "Improving video-on-demand server efficiency through stream tapping," in *Proc. International Conference on Computer Communications and Networks*, pp. 200–207, 1997.
- [29] W. Liao and V. O. K. Li, "The split and merge (sam) protocol for interactive video-on-demand systems," in *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, (Washington, DC, USA), p. 1349, IEEE Computer Society, 1997.
- [30] R. Spiers, R. Marston, R. Good, and N. Ventura, "The uct ims iptv initiative," *Next Generation Mobile Applications, Services and Technologies, International Conference on*, vol. 0, pp. 503–508, 2009.
- [31] D. Waiting, R. Good, R. Spiers, and N. Ventura, "Open source development tools for IMS research," in *Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*, p. 41, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

- [32] T. M. C. Riede, A. Al-Hezmi, "Session and media signaling for iptv via ims," *Mobilware'08 February 12-15*, 2008.
- [33] K. Singh and H. Schulzrinne, "Unified messaging using SIP and RTSP," in *IP Telecom Services Workshop*, pp. 31–37, Citeseer, 2000.
- [34] IETF, *Media Control Protocol Requirements - IETF Draft* <http://tools.ietf.org/>. 2006.
- [35] IETF, *SIP extensions for media control - IETF Draft* <http://tools.ietf.org/html/draft-siva-sip-media-01/>. 2009-07.
- [36] E. Nahum, J. Tracey, and C. Wright, "Evaluating SIP proxy server performance," in *17th International Workshop on Networking and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Citeseer.
- [37] K. Ram, I. Fedeli, A. Cox, and S. Rixner, "Explaining the Impact of Network Transport Protocols on SIP Proxy Performance," in *Performance Analysis of Systems and software, 2008. ISPASS 2008. IEEE International Symposium on*, pp. 75–84, IEEE, 2008.
- [38] J. Lee, "On a unified architecture for video-on-demand services," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 38–47, 2002.
- [39] L. Guo, S. Chen, Z. Xiao, and X. Zhang, "Analysis of multimedia workloads with implications for internet streaming," in *WWW '05: Proceedings of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 519–528, ACM, 2005.
- [40] T. Little and D. Venkatesh, "Prospects for interactive video-on-demand," vol. 1, p. 14, Citeseer, 1994.
- [41] H. Bouwman, C. Carlsson, F. Molina-Castillo, and P. Walden, "Barriers and drivers in the adoption of current and future mobile services in Finland," *Telematics and Informatics*, vol. 24, no. 2, pp. 145–160, 2007.
- [42] J. Altgeld and J. D. Zeeman, "The iptv-vod challenge - upcoming business models," in *BM Paper for the International Engineering Consortium*, 2008.
- [43] D. H. Shin, "Potential user factors driving adoption of iptv, what are customers expecting from iptv?," *Elsevier-Technological Forecasting and Social Change* 74, pp. 1446–1464, 2007.
- [44] D. A. Adams, R. R. Nelson, and P. A. Todd, "Perceived usefulness, ease of use, and usage of information technology: a replication," *MIS Q.*, vol. 16, pp. 227–247, June 1992.
- [45] M. L. I. Ruczinski, C. Kooperberg, "Logic regression," *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pp. 475–511, 2003.
- [46] K. Kilkki, "Quality of Experience in Communications Ecosystem," *Journal of Universal Computer Science*, vol. 14, no. 5, pp. 615–624, 2008.
- [47] F. Pereira, "A triple user characterization model for video adaptation and quality of experience evaluation," in *Proc. Seventh IEEE Multimedia Sig. Process. Workshop*, pp. 1–4, 2005.

- [48] ITU-T, *G.1080: Series G: TRANSMISSION AND MEDIA, DIGITAL SYSTEMS AND NETWORKS Multimedia quality of service and performance - Generic and user-related aspects*. 2008-12.
- [49] G.-M. M. Cristina Hava Muntean, "Framework for Interactive Personalised IPTV for Entertainment," in *IPTV Workshop, International World Wide Web Conference, May 23, 2006*.
- [50] C. P. M. Handley, V. Jacobson, *SDP: Session Description Protocol RFC 4566*. 2006-07.
- [51] C. M. S.-M. E. M. S. M. F. Y. Tim Bray, Jean Paoli, *XML: Extensible Markup Language*. 2008-11.
- [52] A. Vetro, "Mpeg 21 digital item adaptation: Enabling universal multimedia access," 2004.
- [53] J. Simoes, C. Riede, and T. Magedanz, "New interactive experiences with IPTV services using an IMS infrastructure," in *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, pp. 217–221, Institute of Electrical and Electronics Engineers Inc., The, 2009.
- [54] A. Cuevas, J. Moreno, P. Vidales, and H. Einsiedler, "The ims service platform: a solution for next-generation network operators to be more than bit pipes," *Communications Magazine, IEEE*, vol. 44, no. 8, pp. 75–81, 2006.
- [55] A. Al-Hezmi, C. Riede, O. Friedrich, S. Arbanowski, and T. Magedanz, "Cross-fertilization of ims and iptv services over ngn," in *Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference*, pp. 153–160, IEEE, 2008.
- [56] O. Friedrich, A. Al-Hezmi, S. Arbanowski, and T. Magedanz, "Next Generation iptv services for an extended ims architecture," in *Autonomous Decentralized Systems, 2007. ISADS'07. Eighth International Symposium on*, pp. 429–436, IEEE, 2007.
- [57] O. Friedrich, A. Al-Hezmi, S. Arbanowski, and T. Magedanz, "Evolution of next generation networks towards an integrated platform for ims-based iptv services," in *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, pp. 10–10, IEEE, 2007.
- [58] B. Chatras and M. Sa'id, "Delivering quadruple play with iptv over ims," *Journal of the Institute of Telecommunications Professionals*, pp. 9–14, 2007.
- [59] H.-Y. K. Yong-Ju Lee, Ok-Gee Min, "Performance evaluation technique of the rtsp based streaming server," *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS05)*, 2005.
- [60] O. www.opensips.org, "Opensip," 2011. [Online; Accessed 16th-March-2011].
- [61] MySQL, "Mysql www.mysql.com," 2011. [Online; Accessed 16th-March-2011].
- [62] V. L. Project, "Video lan <http://www.videolan.org/>," 2009. [Online; Accessed 01st-February-2010].
- [63] R. G. David Waiting and R. Marston, "How to run the uct advanced iptv server http://uctimsclient.berlios.de/uctiptv_advanced_howto.html/," 2010. [Online; Accessed 12th-March-2010].

- [64] Swik, "Exosip website <http://swik.net/eXosip>," 2010. [Online; Accessed 05th-February-2010].
- [65] D. Wieers, "Dstat: Versatile resource statistics tool <http://dag.wieers.com/home-made/dstat/>," 2010. [Online; Accessed 12th-March-2010].
- [66] O. <http://openmaniak.com/iperf.php>, "Iperf," 2010. [Online; Accessed 08th-March-2010].
- [67] Openmaiak, "Iperf <http://openmaniak.com/iperf.php>," 2010. [Online; Accessed 08th-March-2010].
- [68] N. TV, "Now tv <http://www.now-tv.com>," 2011. [Online; accessed 20-April-2011].
- [69] K. K. Masahito Kawamori and J. Kishigami, "The iptv-vod challenge - upcoming business models," in *Global Standardization Activity - Recent Trends in IPTV*, 2008.
- [70] K. Carney, M. Fandre, R. Nambiath, and N. Shringarpure, "IPTV: Business Model Analysis from Porters Five Forces Perspective," *Retrieved March*, vol. 23, p. 2008, 2006.
- [71] I. F. G. I. W. C-0608, *FG IPTV-C-0610, 4th FG IPTV meeting: Bled*. 2007-05-11.
- [72] O. I. F. Members, *Functional Architecture V 1.1*. Open IPTV Forum, 2008-01.
- [73] S. et al, *RTSP: Real Time Streaming Protocol RFC 2326*. 1998-04.
- [74] Gstreamer, "Gstreamer open source multimedia framework <http://www.gstreamer.net/>," 2010. [Online; Accessed 05th-February-2010].
- [75] TED, "Technology, entertainment and design <http://www.ted.com/>," 2010. [Online; Accessed 12th-March-2010].

Appendix A

IPTV Business Models and Current IPTV Standards

Due to the ever diminishing revenue from fixed-line services, many Telecommunications operators (Telcos) are looking for alternative means of making profits. Triple-play services offer a very lucrative business opportunity to most operators. Since most of the Telcos already own the fixed line infrastructure, they could use this infrastructure to deploy triple-play services.

However a business model that a Telco operator employs is critical to the success of the triple-play service deployment. In the following sections, we will now look at how different Telcos deployed IPTV services.

A.1 Telco Example

Now TV was the first IPTV service in the world. Now TV is being offered by PCCW. PCCW is a large Telco company in China. Now TV has experienced rapid acceptance and growth since its inception in 2003. With 850 thousand IPTV subscribers in August 2007 [68], Now TV has the largest number of IPTV subscribers in the world.

There are a number of reasons behind the Now TV success. Now TV has been able to strike a number of key exclusivity contracts for premium content with ATV, HBO, Cinemax, Star Movies, MGM and ESPN Star Sports and has also secured a three-year contract for Premier League matches. A-la-carte pricing models provide flexibility in channel selections.

The only set-back of Now TV is that it has a walled-garden IPTV model. This means that if any other Telco in the world wants to deploy an IPTV service like one deployed by Now TV, they have to buy or pay royalty costs to PCCW. This is a major setback as it stifles innovation, competition and growth at different levels of the IPTV chain (i.e. content distributor, IPTV media delivery, IPTV transport, IPTV equipment levels).

A.2 Why some IPTV Trials Fail

Internet Protocol Television (IPTV) has been deployed in a number of countries and has been provisioned with mixed success. This was influenced by a number of factors. The main factors

were lack of¹:

- good content in the media
- interoperability between different IPTV components
- integrated services (to increase Average Revenue Per User-ARPU)

A.2.1 Flexibility

IPTV systems that were developed were proprietary and 'once-off' systems. This made it difficult for IPTV systems to be upgraded. Due to the high Capital Expenditure (CAPEX) cost of proprietary systems, it was difficult for operators to introduce a new service or upgrade incumbent systems.

IPTV architecture also made very little concession for a change (in case of an ostensible failure/dissatisfaction) in the operator's Business Model. For example, many Telco operators perceived IPTV as an overhaul of incumbent satellite/cable TV systems. As a result they constructed architectures that gave them full control and monopoly over different levels of the IPTV architecture. The content, media delivery, media transport and set top boxes were all provisioned by one IPTV operator. These operators commonly structured and designed an architecture that fits a particular business model of the operator. The problem with this approach was that if the business model failed, the complete developed system would be rendered useless.

A.2.2 Good Content in the Media

The content provided by many deployed IPTV systems has not been any better than that of classical TV systems. This is due to a couple of reasons. Firstly, IPTV operators have been inclined to deploy their own (copyrighted) content in their respective systems. Most of this content was not appealing to IPTV users.

Secondly, generating good content for users was especially a challenge as operators were not establishing mutually beneficial business agreements with other prominent TV or video content distributors. The reluctance on the side of operators to establish these agreements could have been spawned by bad business models (discussed below).

Thirdly, lack of established Digital Rights Management (DRM) and Media Security systems made it difficult for operators to guarantee security and copyright enforcements. This further discouraged business agreements between operators and prominent TV/Video content distributors and thus adversely affected the type and value of content that operators offered to IPTV users.

A.2.3 Interoperability between Different Architectures

One of the main setbacks of early IPTV systems was that since they were proprietary, interoperability was a big hurdle. For example, an IPTV user would need to replace all of the IPTV equipment if they wanted to change a TV provider.

¹Most of the insights for this discussion were observed from several White Papers IBM [42], Cyber Solutions Laboratories [69]

A.2.4 Integrated Services

Video content delivery over IP networks could be further enhanced by enabling other services (VoIP, IM, Video conferencing) to be integrated with the TV service. Research has shown that IPTV user experience could increase when the IPTV service is offered in conjunction with other services [43].

A.2.5 The Need for an IPTV standard

All of the above mentioned failures of early IPTV highlight the need for an international IPTV standard. A global IPTV standard would ensure interoperability between different systems. Flexibility in the standardised architecture would allow for different business models. And a global standard will also allow for robust laws and regulations to be established that will protect different players in the IPTV market. This will further encourage strong business relationships between IPTV operators and video/film content distributors. These relationships will ensure that IPTV users get the best IPTV service (e.g. good content) and thus improve the IPTV user experience.

There are however some challenges with regards to developing a robust international IPTV standard. There are many recommended IPTV standards. Most of them are either still work-in-progress architectures or completed with various inadequacies.

A.3 Requirements of IPTV Standards

The requirements of IPTV standards (or solution) are set by The Alliance for Telecommunications Industry Solutions Interoperability Forum (ATIS IIF) and have been modified by Telecoms and Internet converged Services & Protocols for Advanced Networks (TISPAN) in the TS 186 006 document. The following requirements are especially chosen for their pertinent relevance to this discussion:

- A good IPTV standard should be provisioned with high quality video
- Network Security and Piracy issues should be thoroughly addressed by an IPTV standard

Carney et al have used a Porter's Five Star model to show that content security is indeed one of the major factors in IPTV business models [70].

An IPTV standard should also have the following characteristics:

- An SDO with good international relations and status
- A well-defined standardisation scope
- A well-defined NGN platform
- A High Level IPTV Architecture
- TV/Video/Content Platform
- DRM system

- Security Protocols
- Integrated Services platform
- Advertisements and Interactive TV capability

IPTV standards discussed in this paper are NGN confined. The reason for this is that NGN architectures address most of the issues historically and currently encountered in modern IP networks. NGN architectures thus serve as a good platform for IPTV deployment in the foreseeable future.

A.3.1 ETSI TISPAN

The body international relations and status

European Telecommunications Standards Institute (ETSI) is an independent, non-for-profit, standardisation organisation of the telecommunications industry (equipment makers and network operators) in Europe, with worldwide projection. ETSI has been successful in standardising the GSM cell phone system amongst many other systems. TISPAN is an ETSI core competence centre for fixed networks and for migration from switched circuit networks to packet-based networks with an architecture that can serve in both.

The standardisation body scope

TISPAN is responsible for all aspects of standardisation for present and future converged networks including the NGN (Next Generation Network) and including, service aspects, architectural aspects, protocol aspects, QoS studies, security related studies, mobility aspects within fixed networks, using existing and emerging technologies. This work is in line with, and driven by, the commercial objectives of the ETSI membership (from TISPAN website).

A.3.2 ITU-T IPTV Focus Group

The body international relations and status

Due to its longevity as an international organisation and its status as a specialised agency of the United Nations, standards promulgated by the ITU carry a higher degree of formal international recognition than those of most other organisations that publish technical specifications of a similar form. It is a standardisation body that enjoys government sponsorships, multi-telco support and is renowned for establishing widely recognised standards, government enforced regulations and publishing reports on various technologies.

The standardisation body scope

The scope is very broad. Various telecommunications-related technologies with specialised standardisation groups (e.g. IPTV FG SG13) are discussed. Each group may have several working groups (WGs) under it.

The NGN platform

WG5 - End Systems and Interoperability WG4 Network Control

The high level IPTV architecture

WG1 Architecture (ATIS Architecture)

The Video/TV Content Platform

WG6 - Middleware and Content Platforms

The Architecture Media Delivery Platform

WG4 Network Control WG2 QoS

Integrated Services Platform

WG6 Application Middleware

The Architectures DRM system

WG6 - Middleware and Content Platforms WG3 Security, Content Protection

The Architecture security protocols

WG3 Security, Content Protection

Interactive TV and Advertisements Capability

WG6 Application Middleware

A.3.3 Open IPTV Standard

The body international relations and status

Currently (as of May 26, 2008) OIF has about 29 members from IT and Telecom industry.

The standardisation body scope

Not yet verifiable.

A.3.4 ATIS IIF

The Alliance for Telecommunications Industry Solutions IPTV Interoperability Forum (ATIS IIF), has proposed an IPTV standard that mainly tackles inter-operability issues.

The body international relations and status

ATIS IIF has developed eight standards in under two years of its existence.

ATIS has representatives from the requisite industry sectors and has established liaisons with many leading organisations working in the IPTV realm. These include companies like British Telecoms, Huawei Technologies, Cisco Systems AT&T. ATIS IIF also has liaisons with standards bodies inclusive of ITU-T FG IPTV, TISPAN, DSL Forum, Digital Living Network Alliance (DLNA).

The standardisation body scope

According to ATIS, the scope of IPTV Interoperability Forum is coordinating IPTV related activities. IIF works closely with other SDOs to develop Implementation Agreements (IAs) and technical reports and other types of ATIS standards where appropriate. The scope of documents published to date by IFF is High Level IPTV and DRM requirements.

The NGN platform

The NGN platform for ATIS systems is non-defined. The IIFs High Level Architecture allows for both core IMS and non-IMS approaches for IPTV in the NGN framework.

Integrated Services Platform

Although a full IFF IPTV High Level Architecture Standard paper could not be accessed. IFF does not seem to define a comprehensive integrated services platform. This is because IIF does not have a well-defined NGN platform of choice.

The Architecture security protocols

ATIS main function is to coordinate (not define) IPTV related activities like define security protocols, Interactive TV and Advertisement Capabilities.

A.3.5 3GPP MTV MBMS over 3G

MBMS is limited to mobile networks. MBMS was not developed for IPTV Integrated services. Allows for smooth transition from 3G network access provisioning to service provisioning

A.3.6 The TV Anytime Forum

Metadata for EPG and DRM

A.3.7 OMA

The body international relations and status

The Open Mobile Alliance (OMA) is well known in the mobile application development industry.

The standardisation body scope

NGN end-system features on mobile devices.

A.3.8 Video on Demand and the IP Multimedia Subsystem

VoD Standardisation

Video on Demand solutions need to be standardised to ensure successful adoption of the service. Installation cycle times can also be significantly reduced when a common design standard is chosen.

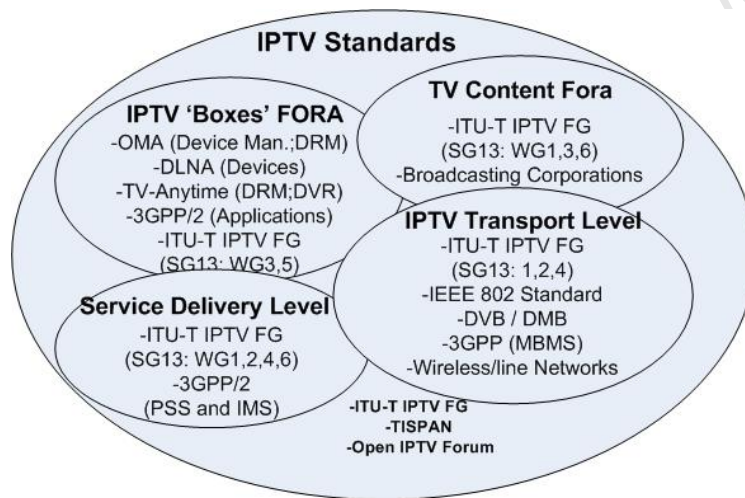


Figure A.1: Video on Demand Standards

Standards that enforce robust Quality of Service (QoS), charging and service security measures will be critical to the success of the VoD service. Standards are necessary to ensure interoperability between various implementations and to reduce installation cycle times and related costs. Taxonomy of different standardisation bodies at different levels of the IPTV-VoD market chain (i.e. user equipment, transport, service and content production) is shown in Figure A.1.

Standards enable different players on the VoD market chain to mutually participate in the proliferation of new services and aggregation of content that is attractive to VoD subscribers [15]. There are three main standardisation bodies for IPTV: Open IPTV Forum, TISPAN and the ITU-T IPTV Focus Group (FG). Open IPTV Forum is a consortium of companies (about 26 members to date) that has formulated and published an open IPTV standard ². TISPAN is a European Telecommunications Standards Institute (ETSI) core competence centre responsible for all aspects of standardisation for present and future converged networks including the NGN ³. ITU-T

²<http://www.openiptvforum.org/>

³<http://www.etsi.org/tispan/>

IPTV FG coordinates and promotes the development of global IPTV standards taking into account the existing work of the ITU study groups as well as Standards Developing Organisations, Fora and Consortia ⁴.

The IMS VoD Standard

The IP Multimedia Subsystem is a service delivery platform for NGN Multimedia services such as video telephony, presence, VoIP and VoD. The IMS was developed by Third Generation Partnership Project groups (3GPP and 3GPP2). The IMS allows service providers to use a unified all-IP network to deploy new services with the least cost and risk. Quality of Service (QoS), fair charging schemes and integrated services have been touted to be niche service enabling technologies provided by the IMS.

Service enabling technologies offered by the IMS offer a number of advantages when a VoD service is deployed over the IMS. QoS can ensure high quality video provisioning; a charging system can be used to price VoD clients fairly; and a unified all-IP platform will be central in developing interactive and conversational services that could be integrated with the VoD service. ITU-T [71], TISPAN [4] and Open IPTV [72] Forum use the IMS as a NGN platform for deploying video services. Given that all these main IPTV standardisation bodies support the IMS platform, the NGN VoD service will thus likely be deployed over the IMS platform for interoperability and efficient VoD service deployment.

IMS IPTV Functions

The SCF functions are defined as follows:

- Service authorisation during session initiation and session modification, which includes checking IPTV users profiles in order to allow or deny access to the service.
- Credit limit and credit control

With regards to managing interactivity, the functions of the MCF are:

- Handling media flow control of MDF.
- May manage the media processing of MDF.
- Monitoring the status of MDF.
- Managing interaction with the UE (e.g. trick mode commands).

The MDF is primarily responsible for:

- Handling media flows delivery (for delivering multimedia services to user).
- Status reporting to MCF (e.g. reporting on established IPTV media streams).
- Store of media and may also store some service information stored with media for IPTV services.

⁴<http://www.itu.int/ITU-T/>

Appendix B

Protocols' Overview and Implementation

In live streaming video interactivity architectures, signalling protocols are required to control the video streams during a streaming session. The advantage of video interactivity signalling protocols is that live media streams can be controlled at the video server without waiting for the streams to be delivered across the network and to be buffered (and controlled) locally at the client device. Video interactivity protocols are generally designed to be separate from the media transport layer. They do not directly interact with protocols such as the Real Time Protocol (RTP) that stream video content.

Figure B.1 provides an overview of the signalling and media planes of a video system. As shown in the picture, the signalling plane is kept separate from the media plane. As a result, although inter-related, the client-server modules for video streaming and for signalling (for video stream control) are kept separate. For example, a signalling plane message such as *PAUSE* will be delivered out-of-band from the client to the server, in order to cause the effect of halting the video stream. Since the signalling is separate from the media plane, different signalling protocols can be used to achieve interactivity with video content.

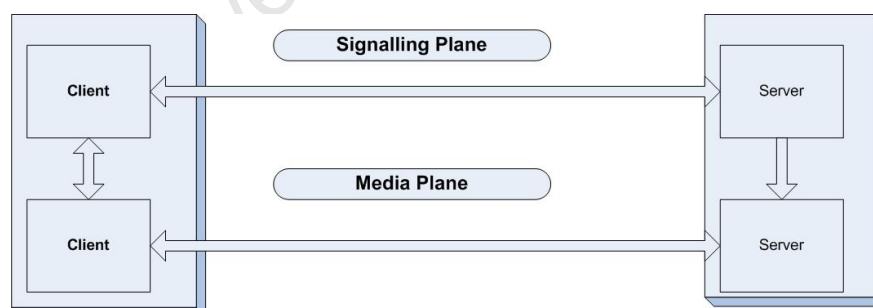


Figure B.1: Signaling and Media Processing

The two main media protocols used in the signalling plane for controlling media sessions, RTSP and SIP, are discussed below.

B.1 RTSP

The Real Time Streaming Protocol (RTSP) was developed by Real Networks. The protocol was further enhanced and adopted by the Internet Engineering Task Force (IETF) and published as RFC 2326 in 1998 [73]. RFC 2326 defines RTSP as an application-level protocol for video control on data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video.

The RTSP protocol is a connectionless protocol that keeps track of the state of a session through the use of a session identifier. Clients are allowed to issue VCR-like commands such as play and pause to enable media playback functionality with the media server.

The RTSP protocol supports the following main operations:

- Retrieval of media from a media server
- Invitation of a media server to the conference
- Addition of media to an existing presentation

According to the RTSP RFC, RTSP supports the following modes:

- PLAY request will cause one or all media streams to be played
- PAUSE request temporarily halts one or all media streams that can later be resumed with a PLAY request
- RECORD is used to send a stream to the server for storage
- SET PARAMETER sends requests to set the value of a parameter for a presentation or stream specified by the Uniform Resource Identifier (URI)
- GET PARAMETER requests different values of parameters

The simple design of the protocol lends it to being a good candidate for implementing video interactivity with low latency. There are currently no existing exhaustive evaluations of the latencies incurred by the RTSP protocol as most designs depend on the computing platform used and the number of components deployed in the implementation. Most RTSP implementations are proprietary. An example of an RTSP 'play request' message is shown below:

```
PLAY rtsp://10.128.1.46:9554/carver RTSP/1.0
CSeq: 1
Session: 1
```

The first line indicates the request URI and the protocol version number. The second line indicates the sequence number of an RTSP message within a session, and the last line indicates the session number.

B.2 SIP

SIP is a signalling protocol for controlling multimedia sessions for services such as Instant Messaging and Voice over IP (VoIP). SIP is a control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls [25].

Like the RTSP protocol, SIP is not a vertically integrated communications protocol. This means it is not related to any media transport protocol like the Real Time Protocol (RTP). SIP is based on an HTTP-like request-response model. A SIP request-response transaction happens within a dialog. A SIP dialog is initiated by an INVITE message. A dialog is a peer-to-peer SIP relationship between two user agents that persists for some time.

The following are the methods used by SIP:

- INVITE - indicates a client is being invited to participate in a call session
- ACK - confirms that a client has received the final response to an INVITE request
- BYE - terminates a call and can be sent by either the caller or the callee
- CANCEL - cancels any pending request
- OPTIONS - queries the capabilities of servers
- REGISTER - registers the address listed in the 'To' header field with a SIP server
- PRACK - provisional acknowledgement
- SUBSCRIBE - subscribes for an Event of Notification from the Notifier
- NOTIFY - notify the subscriber of a new Event
- PUBLISH - publishes an event to the Server
- INFO - sends mid-session information that does not modify the session state
- REFER - asks recipient to issue SIP request
- MESSAGE - transports instant messages using SIP
- UPDATE - modifies the state of a session without changing the state of the dialog

An example of a SIP 'invite' message is shown below:

```
INVITE sip:carver@1xx.1xx.1xx.2xx:2244 SIP/2.0
Via: SIP/2.0/UDP 1xx.1xx.1xx.2xx:5061;rport;branch=z9hG4bK1503321643
Route: <sip:mo@pcscf.vod-ims.test:4060;lr>
Route: <sip:mo@scscf.vod-ims.test:6060;lr>
From: "Bob" <sip:bob@vod-ims.test>;tag=913175202
To: <sip:carver@1xx.1xx.1xx.2xx:2244>;tag=1661416236
Call-ID: 631594936
CSeq: 22 INVITE
Contact: <sip:bob@1xx.1xx.1xx.2xx:5061>
Content-Type: application/sdp
```

```
Max-Forwards: 70
User-Agent: UCT IMS Client
P-Preferred-Identity: "Bob" <sip:bob@vod-ims.test>
Privacy: none
P-Access-Network-Info: 3GPP-GERAN
Require: sec-agree
Proxy-Require: sec-agree
Supported: 100rel
Content-Length: 565
```

```
v=0
o=- 0 0 IN IP4 1xx.1xx.1xx.2xx
s=IMS Call
c=IN IP4 1xx.1xx.1xx.2xx
t=0 0
m=audio 36266 RTP/AVP 0 8 101
b=AS:64
a=curr:qos local none
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos mandatory remote sendrecv
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11
m=video 29100 RTP/AVP 96
b=AS:128
a=curr:qos local none
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos mandatory remote sendrecv
a=rtpmap:96 H263-1998
a=fmtp:96 profile-level-id=0
```

Where 1xx.1xx.1xx.2xx represents any IP address. The message above is shown merely to demonstrate the layout and length of a typical SIP message. The relative bulkiness of a SIP message (compared to the RTSP message) is evident in the message shown above. The explanation of the complete design of the SIP message structure can be found in the SIP RFC 3261[25] and falls outside the scope of this discussion. The SIP protocol was primarily designed to work through proxy servers that execute filters and sometimes caches for various content (e.g. web/HTML, audio, video) servers and hence the design and implementation of SIP messages is relatively more complex and allows less direct interaction with content compared to the RTSP protocol.

B.3 Video Interactivity Signalling Implementation

B.3.1 RTSP Trick-play

The RTSP signalling was implemented using C based socket.h library. Although the current UCT IMS Client already had a RTSP trick-play implementation, the interface was extended as explained in Section 4.4.1 to support gstreamer media control. The RTSP trick-play request, as per RTSP RFC [73] is as follows:

For play:

```
1 PLAY rtsp://137.158.125.230:9554/carver RTSP/1.0
2 CSeq: 0
3 Session: 2
```

For pause:

```
1 PAUSE rtsp://137.158.125.230:9554/carver RTSP/1.0
2 CSeq: 0
3 Session: 2
```

B.3.2 SIP-based Trick-play

The eXoSip [64] module was used to implement SIP-based trick play.

The following SIP message, adapted from the signalling implementation of the TISPAN compliant UCT IMS Client, was used to pause a video session:

```
1 INVITE sip:carver@137.158.125.230:2244 SIP/2.0
2 Via: SIP/2.0/UDP 137.158.125.230:5061;rport;branch=z9hG4bK931274078
3 Route: <sip:mo@pcscf.vod-ims.test:4060;lr>
4 Route: <sip:mo@scscf.vod-ims.test:6060;lr>
5 From: "Bob" <sip:bob@vod-ims.test>;tag=355622925
6 To: <sip:carver@137.158.125.230:2244>;tag=393847847
7 Call-ID: 574232647
8 CSeq: 25 INVITE
9 Contact: <sip:bob@137.158.125.230:5061>
10 Content-Type: application/sdp
11 Max-Forwards: 70
12 User-Agent: UCT IMS Client
13 P-Preferred-Identity: "Bob" <sip:bob@vod-ims.test>
14 Privacy: none
15 P-Access-Network-Info: 3GPP-GERAN
16 Require: sec-agree
17 Proxy-Require: sec-agree
18 Supported: 100rel
19 Content-Length: 565
20
21 v=0
```

```

22 o=- 0 0 IN IP4 137.158.125.230
23 s=IMS Call
24 c=IN IP4 137.158.125.230
25 t=0 0
26 m=audio 21839 RTP/AVP 0 8 101
27 b=AS:64
28 a=curr:qos local none
29 a=curr:qos remote none
30 a=des:qos mandatory local sendrecv
31 a=des:qos mandatory remote sendrecv
32 a=rtpmap:0 PCMU/8000
33 a=rtpmap:8 PCMA/8000
34 a=rtpmap:101 telephone-event/8000
35 a=siptrick:pause

```

The last sip attribute line *siptrick:pause* is used for SIP-based interactivity requests. Similarly, SIP-based interactivity requests for rewind, forward, stop and play where implemented by using *siptrick:rewind*; *siptrick:forward*; *siptrick:stop*; *siptrick:play* attribute lines respectively.

B.4 Media Processing Layer Implementation

The gstreamer library [74] was used to implement media processing. The UCT Advanced IPTv Server initially used the VLC LAN project [62] for loading media server streams. The server was modified to use gstreamer modules. The C-based gstreamer library allows for low level manipulation of video streams. Video streams are formed from the gstreamer elements concatenated into a *pipeline*. Gstreamer elements include video codec type, codec quality, picture quality, and container format. The modularised design of the gstreamer library allows for easy manipulation of the video streams during media processing. Various client video interactivity processes can efficiently handled through the gstreamer framework.

B.4.1 Implementing Pause and Play

To implement PAUSE and PLAY, the gstreamer `gst_element_set_state` method as shown below:

```

GstStateChangeReturn  gst_element_set_state (GstElement *element,
                                           GstState state);

```

GstState can be set to PAUSED for pausing and PLAYING for resuming play.

B.4.2 Implementing Forward and Rewind

In this implementation the video playback *speed* is increased, and this increases the framerate. REWIND, FORWARD were implemented through the use of the gstreamer SEEK method, with the parameters as shown below:

```

1 void forward_client(GstElement *stream)
2 {
3     GstFormat fmt = GST_FORMAT_TIME;
4     gint64 pos, len;
5     if (gst_element_query_position (stream, &fmt, &pos)
6         && gst_element_query_duration (stream, &fmt, &len)){
7         gst_element_seek(stream,4.0, GST_FORMAT_TIME,GST_SEEK_FLAG_FLUSH,
8             GST_SEEK_TYPE_SET, pos, GST_SEEK_TYPE_NONE,
9             GST_CLOCK_TIME_NONE);
10            //g_print ("Time: %" GST_TIME_FORMAT " / %" GST_TIME_FORMAT "\r",
11 //GST_TIME_ARGS (pos), GST_TIME_ARGS (len));
12 }
13 else
14     printf("query failed \n");
15     gst_element_set_state(stream, GST_STATE_PLAYING);
16 }

```

The (`gst_element_query_position (stream, &fmt, &pos)` and the `&& gst_element_query_duration (stream, &fmt, &len)`) (Lines 5 and 6) query the current position of the video stream. The `gst_element_seek(stream,4.0, GST_FORMAT_TIME,GST_SEEK_FLAG_FLUSH,, GST_SEEK_TYPE_SET, pos, GST_SEEK_TYPE_NONE, and GST_CLOCK_TIME_NONE);`) (Lines 7, 8 and 9) set the framerate to a multiple of 4 to increase the playback rate. The user could then decide to pause the video where he wishes to continue watching the video stream.

```

1 void rewind_client(GstElement *stream){
2
3     GstFormat fmt = GST_FORMAT_TIME;
4     gint64 pos, len;
5
6     if (gst_element_query_position (stream, &fmt, &pos)
7         && gst_element_query_duration (stream, &fmt, &len)){
8         gst_element_seek(stream,-4.0, GST_FORMAT_TIME,GST_SEEK_FLAG_FLUSH,
9             GST_SEEK_TYPE_SET, pos, GST_SEEK_TYPE_NONE,
10            GST_CLOCK_TIME_NONE);
11     g_print ("Time: %" GST_TIME_FORMAT " / %" GST_TIME_FORMAT "\r",
12         GST_TIME_ARGS (pos), GST_TIME_ARGS (len));
13
14 }
15 else
16     printf("query failed \n");
17     gst_element_set_state(stream, GST_STATE_PLAYING);
18 }

```

The rewind code is very similar to the forward code above, the only difference is that the playback rate is set in the negative direction instead.

Appendix C

Bandwidth Estimations

The values shown in Table C.1 and Table C.2 were taken to measure the bandwidth of the WiMAX and Ethernet bandwidth.

Number of Instance	WiMAX Link Speed (Mbps)	95th Percentile Values (Mbps)
1	8.18	8.18
2	8.54	0
3	7.83	7.83
4	7.89	7.89
5	7.91	7.91
6	7.93	7.93
7	7.90	7.90
8	7.90	7.90
9	7.74	7.74
10	7.77	7.77
Average (Mbps)	7.96	7.89
Min (Mbps)	7.74	7.74
Max (Mbps)	8.54	8.18

Table C.1: WiMax Link Speeds

C.1 MDF Video Server Setup

The following videos were used during the testing: Sunitha, Malika and Pilo that were obtained from the Technology Entertainment and Design (TED) website.

Number of Instance	Ethernet Link Speed (Mbps)	95th Percentile Values (Mbps)
1	95.30	95.30
2	94.80	94.80
3	94.90	94.90
4	94.90	94.90
5	94.90	94.90
6	95.80	0
7	94.00	94.00
8	94.60	94.60
9	94.80	94.80
10	94.80	94.80
Average (Mbps)	94.89	94.77
Min (Mbps)	94.00	94.00
Max (Mbps)	95.80	95.30

Table C.2: Ethernet Link Speeds

Appendix D

Ethics of Content, Hardware and Software Used

Most of the video content material used in this research was legally and legitimately obtained from free sources. The videos were freely obtained from the Technology Entertainment and Design (TED) Talks website [75]. TED allows free downloads of the video content, according to the following guidelines:

1. Attribution: you reference explicitly TED as the original source of the materials, and TED's logos and visuals as well as those of the TEDTalks sponsors remain untouched and unedited.
2. NonCommercial: You can't use TED Talks (or any parts of them) for commercial purposes
3. NonDerivative: You cannot alter the videos in any way (edit, remix, cut, etc)

The Hardware used is part of the purchased UCT Communications Research Group (CRG) equipment.

Open Source software, which allows for modification and re-distribution of source code, was used to implement the evaluation platform and testing scenarios.

Appendix E

Accompanying CD-ROM

The following information can be found on the CD-ROM that has been included with this document.

Software: All the source code that was used to develop the evaluation framework can be found in the 'Software' directory.

Server Code: Software/Code For Video Server With SIP-RTSP Video Interactivity Support

Client Code: Software/Code For RTSP - SIP Based Video Interactivity Clients

Evaluation Results: Raw data collected during performance evaluations in the Evaluation Results directory.

Client (Ethernet) RTSP: Evaluation Results/Client Latencies/Ethernet_Tests/RTSP Delays

Client (Ethernet) SIP: Evaluation Results/Client Latencies/Ethernet_Tests/SIP Delays

Client (WiMAX) RTSP: Evaluation Results/Wimax Tests/RTSP Delays

Client (WiMAX) SIP: Evaluation Results/Client Latencies/Wimax Tests/SIP Delays

Nothing Running Load: Evaluation Results/Interactivity Results/Scalability-Interactivity-Testing/Processing Load/Server Processing Load/Nothing

Server Running Load: Evaluation Results/Interactivity Results/Scalability-Interactivity-Testing/Processing Load/Server Processing Load/Server Running

Signalling Load at different rates: Evaluation Results/Interactivity Results/Scalability-Interactivity-Testing/Processing Load/Server Processing Load/Signalling Load

Video Processing Load at different rates: Evaluation Results/Interactivity Results/Scalability-Interactivity-Testing/Processing Load/Server Processing Load/Video Processing Load

TLVIF Load at different rates: Evaluation Results/Interactivity Results/Scalability-Interactivity-Testing/Processing Load/Server Processing Load/TLVIF Load

Selected Publications: Collection of written papers resulting from this work.

Research Literature: Electronic copies of research papers and other literature used in this research can be found in the 'Research Literature' directory.

Thesis: This document, in PDF format, can be found in this directory.