Optimizing IETF Multimedia Signaling Protocols and Architectures in 3GPP Networks. An evolutionary approach

---

PhD Thesis

PhD Student: David Viamonte Solé

Thesis Director: Anna Calveras Augé

Wireless Networks Group (WNG)

Telematics Engineering Department (EnTel)

Technical University of Catalonia (UPC)

January 2019

Abstract

Signaling in Next Generation IP-based networks heavily relies in the family of multimedia signaling protocols defined by IETF. Two of these signaling protocols are RTSP and SIP, which are text-based, client-server, request-response signaling protocols aimed at enabling multimedia sessions over IP networks. RTSP was conceived to set up streaming sessions from a Content / Streaming Server to a Streaming Client, while SIP was conceived to set up media (e.g.: voice, video, chat, file sharing, …) sessions among users. However, their scope has evolved and expanded over time to cover virtually any type of content and media session.

As mobile networks progressively evolved towards an IP-only (All-IP) concept, particularly in 4G and 5G networks, 3GPP had to select IP-based signaling protocols for core mobile services, as opposed to traditional SS7-based protocols used in the circuit-switched domain in use in 2G and 3G networks. In that context, rather than reinventing the wheel, 3GPP decided to leverage Internet protocols and the work carried on by the IETF. Hence, it was not surprise that when 3GPP defined the so-called Packet-switched Streaming Service (PSS) for real-time continuous media delivery, it selected RTSP as its signaling protocol and, more importantly, SIP was eventually selected as the core signaling protocol for all multimedia core services in the mobile (All-)IP domain. This 3GPP decision to use off-the-shelf IETF-standardized signaling protocols has been a key cornerstone for the future of All-IP fixed / mobile networks convergence and Next Generation Networks (NGN) in general.

In this context, the main goal of our work has been analyzing how such general purpose IP multimedia signaling protocols are deployed and behave over 3GPP mobile networks. Effectively, usage of IP protocols is key to enable cross-vendor interoperability. On the other hand, due to the specific nature of the mobile domain, there are scenarios where it might be possible to leverage some additional “context” to enhance the performance of such protocols in the particular case of mobile networks.

With this idea in mind, the bulk of this thesis work has consisted on analyzing and optimizing the performance of SIP and RTSP multimedia signaling protocols and defining optimized deployment architectures, with particular focus on the 3GPP PSS and the 3GPP Mission Critical Push-to-Talk (MCPTT) service. This work was preceded by a detailed analysis work of the performance of underlying IP, UDP and TCP protocol performance over 3GPP networks, which provided the best baseline for the future work around IP multimedia signaling protocols.

Our contributions include the proposal of new optimizations to enhance multimedia streaming session setup procedures, detailed analysis and optimizations of a SIP-based Presence service and, finally, the definition of new use cases and optimized deployment architectures for the 3GPP MCPTT service. All this work has been published in the form of one book, three papers published in JCR cited International Journals, 5

articles published in International Conferences, one paper published in a National Conference and one awarded patent.

This thesis work provides a detailed description of all contributions plus a comprehensive overview of their context, the guiding principles beneath all contributions, their applicability to different network deployment technologies (from 2.5G to 5G), a detailed overview of the related OMA and 3GPP architectures, services and design principles. Last but not least, the potential evolution of this research work into the 5G domain is also outlined as well.

It takes a leap of faith to get things done
Bruce Springsteen, Leap of Faith, 1992

Acknowledgements

This has been an intensive and extensive work. Looking back, I can only say that I have been lucky to be surrounded by colleagues, friends and family without whom this work would not have reached conclusion.

In 1999 my friendship with Internet multimedia protocols and SIP-based solutions and architectures started, spanning to academic, training, learning and professional activities alike, eventually providing a solid foundation for research and PhD activity to start a few years later. This has been a fun journey driven by curiosity and willingness to never stop exploring.

However, this is a journey I could have never done alone. First, I have to thank my parents, David and Maria Cinta, brother Jordi and sister Mònica for creating the environment that let me land in the Telecom and Computer Networking space and succeed in completing my BSc and MSc.

Professor Josep Paradells was instrumental in introducing me to the new area of Internet multimedia protocols, for which I will always be grateful. He has also been someone to whom I could always talk and share thoughts and impressions. Regardless of his impressive Industry and Academic experience he has always been approachable and friendly.

Truly, I would not be here if I would not have received the support of an extraordinary person. Professor Anna Calveras has been there, each and every day, month, semester and year. A substantial part of the energy, focus and drive to complete this work has come directly from her. From an academic perspective, Dr. Calveras has provided to me brilliant insights, advice, orientation and common sense to the writing of the papers, always helping to improve quality, provide better research context, enhance structure, simplify and make sure that such contributions were submitted on time and manner to the relevant calls. Such impressive academic support was exceeded by her personal attitude and touch. Her perseverance, push and demand to keep working have been a foundation for me to be able to complete this PhD. I will always be thankful to her for believing in this over all the time.

I have been lucky enough to be surrounded by yet another extraordinary woman. Belén, my friend for more than 25 years and my wife has always been a key support, always encouraging to move ahead and to complete the ongoing work. She is my travelling companion and someone to whom I owe so much. She is a brilliant, dedicated, passionate, intelligent, modest, loving person that brings sense and purpose to this task and to our family. I thank her for being there and look forward to living our adventure together.

Over time, this PhD work has seen the birth of Clara, Aroa and Genís. They spark joy, meaning, purpose and happiness. They are the most important thing in my life. As a parent, I am impressed by how special and unique they can be, by their sincerity and by their kindness and the goodness inside of them. My research and academic activities have stolen from them more time than I would have liked. Live is only lived once, and I am looking forward to support them, keep learning from them, keep playing together, sparking joy and growing as a family and as the loving, caring human beings they will become.

Peace and Love.

Table of Contents

List of Figures

List of Tables

Glossary

3GPP	Third Generation Partnership Project
5G	Fith Generation
AAC	(MPEG4) Advanced Audio Coding
AC	Admission Control
AIN	Advanced Intelligent Network
ALG	Application Level Gateway
AMR-NB	Adaptive Multi Rate – NarrowBand
AMR-WB	Adaptive Multi Rate – WideBand
AMR-WB+	Adaptive Multi Rate – WideBand Plus
ATM	Asynchronous Transfer Mode
BDP	Bandwidth Delay Product
BRI	Basic Rate Interface
CAN	Connectivity Access Network
CDMA	Code Division Multiple Access
CODEC	COder – DECoder
COPS	Common Open Policy Service
CS	Circuit Switched
CSCF	Call State Control Function
DANE	DASH-Aware Network Element
DASH	Dynamic Adaptive Streaming over HTTP
DL	Down Link
DTMF	Dual-Tone Multi Frequency
eAACplus	(MPEG4) Enhanced Advanced Audio Coding Plus
EDGE	Enhanced Data rates for the GSM/GPRS Evolution
EPC	Evolved Packet Core
EPS	Evolved Packet System
ETSI	European Telecommunication Standards Institute
EV–DO	EVolved Data-Oriented
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
GSMA	GSM Association
HD	High Definition
HSDPA	High Speed Downlink Packet Access

HSPA	High Speed Packet Access
HSS	Home Subscriber Server
HSUPA	High Speed Uplink Packet Access
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
ICE	Interactive Connectivity Establishment
ICT	Information and Communication Technologies
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IMS	IP Multimedia Subsystem
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
IoT	Internet of Things
IP	Internet Protocol
IP-CAN	IP Connectivity Access Network
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
ISIM	IMS Subscriber Identity Module
LTE	(3GPP) Long Term (RAN) Evolution
MAC	Medium Access Control
MEC	Mobile Edge Computing
MCPTT	Mission Critical Push To Talk
MMTEL	Multi-Media TELephony
MNO	Mobile Network Operator
MPEG4	Motion Pictures Expert Group (specification 4)
MS	Mobile Station
MSC	Mobile Switching Center
MSRP	Message Session Relay Protocol
NAT	Network Address Translation
BFV	Network Function Virtualization
OMA	Open Mobile Alliance
OWD	One Way Delay
PCRF	Policy and Charging Rules Function
P-CSCF	Proxy Call State Control Function
PCC	Policy Charging and Control

PCF	Policy Control Function
PCRF	Policy and Charging Rules Function
PDCP	Packet Data Convergence Protocol
PDF	Policy Decision Function
PDP	Packet Data Protocol
PEF	Policy Enforcement Function
PIDF	Presence Information Data Format
PMR	Professional Mobile Radio
PoC	Push-to-Talk over Cellular
POP	Post Office Protocol
POTS	Plain Old Telephone System
PRI	Primary Rate Interface
ProSe	Proximity Services
PS	Packet Switched
PSAP	Public Safety Answering Point
PSS	Packet-switched Streaming Service
PTT	Push-to-Talk
QoS	Quality of Service
RAB	Radio Access Bearer
RAN	Radio Access Network
RCS	Rich Communication Suite
RDF	Resource Description Framework
RFC	Request For Comments
RLC	Radio Link Control
RLMI	Resource List Meta-Information
RLS	Resource List Server
RPID	Rich Presence Information Data
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SAND	Server And Network Assisted DASH
SCE	Service Creation Environment
SDO	Standards Definition Organization
SDP	Session Description Protocol
SIM	Subscriber Identity Module
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions

SIP	Session Initiation Protocol
SMS	Short Messaging Service
SMTP	Simple Mail Transfer Protocol
SS7	Signaling System 7
SSRC	Synchronization Source Identifier
STUN	Session Traversal Utilities for NAT
TC	Traffic Class
TCP	Transmission Control Protocol
TETRA	TErrestrial Trunked RAdio
TFT	Traffic Flow Template
TIA	Telecoms Industry Association
TURN	Traversal Using Relays around NAT
TTS	Text To Speech
UDP	User Datagram Protocol
UL	Up Link
UMTS	Universal Mobile Telecommunications System
URLLC	(5G) Ultra-Reliable Low Latency Communication
USIM	UMTS Subscriber Identity Module
VCC	Voice Call Continuity
VoIMS	Voice-over-IMS (-over-IP)
VoIP	Voice-over-IP
VoLTE	Voice-over-LTE
VSP	(MPEG4) Visual Simple Profile
WAP	Wireless Application Protocol
WCDMA	Wideband Code Division Multiple Access
XCAP	XML Capability Access Protocol
XDM	XML Document Management
XML	eXtensible Mak-up Language

1 Presentation

Signaling in Next Generation IP-based networks heavily relies in the family of multimedia signaling protocols defined by IETF. Two of these signaling protocols are RTSP and SIP, which are text-based, client-server, request-response signaling protocols aimed at enabling multimedia sessions over IP networks. RTSP was conceived to set up streaming sessions from a Content / Streaming Server to a Streaming Client, while SIP was conceived to set up media (e.g.: voice, video, chat, file sharing, …) sessions among users. However, their scope has evolved and expanded over time to cover virtually any type of content and media session.

As mobile networks progressively evolved towards an IP-only (All-IP) concept, particularly in 4G and 5G networks, 3GPP had to select IP-based signaling protocols for core mobile services, as opposed to traditional SS7-based protocols used in the circuit-switched domain in use in 2G and 3G networks. In that context, rather than reinventing the wheel, 3GPP decided to leverage Internet protocols and the work carried on by the IETF. Hence, it was not surprise that when 3GPP defined the so-called Packet-switched Streaming Service (PSS) for real-time continuous media delivery, it selected RTSP as its signaling protocol and, more importantly, SIP was eventually selected as the core signaling protocol for all multimedia core services in the mobile (All-)IP domain.

With the selection of RTSP and SIP as signaling protocols for the PSS and the IP Multimedia Subsystem (IMS) by 3GPP, the first stone for the convergence of packet-switched and circuit-switched mobile networks was set. With the progressive standardization and maturity of the IMS architecture over 3G, 4G and 5G, the mobile networks of tomorrow will not have a specific infrastructure dedicated for voice calls anymore. Rather, a common IP multimedia core will run all traditional and services. For this purpose, apart from SIP and RTSP, 3GPP has been consistently leveraging standard Internet protocols defined at the IETF, such as DIAMETER, RTP, XCAP, …

As a result, today's and tomorrow's 5G networks will be fully IP-based, with the same protocols being used across fixed, wireless and cellular domains. This choice can represent dramatic savings for the industry as well as remarkable benefits for end users, since all their services can be available anytime, anywhere and from any device, with a common set of protocols and harmonized user experiences.

While the benefits of selecting standards-based IETF signaling protocols are obvious, it is as important to make sure that those protocols behave and perform well while delivering services over such wireless networks, and that the architectures defined by 3GPP scale well for all the use cases requested by consumers as well as professional users alike. Indeed, in the context of a 3GPP-based cellular network, IP packets are exchanged over a number of wireless and fixed links (the radio access network and the core network of the mobile operator) which break, transform, transmit, reassemble and deliver such IP packets from end to end.

In this context, our Thesis work has been focusing on three main areas, namely:

a) Understanding how IETF multimedia signaling protocols and architectures are deployed over 3GPP cellular network to deliver innovative services.
b) Optimizing the performance of such multimedia signaling protocols and architectures defined by 3GPP, without breaking the baseline principle of interoperability and backwards compatibility.
c) Understanding and outlining how the architectures and services that are enabled by such multimedia signaling protocols can be enhanced and enriched to deliver enhanced user experience and new use cases.

While SIP and RTSP can be used in a number of different context, we have focused the above goals in some of the main services that have been standardized for 3GPP networks, namely: the PSS streaming service mentioned above, the Presence service defined by the Open Mobile Alliance (OMA), which consists on sharing user status among contacts to enrich the call experience, and finally the Group Communication services that emulates traditional Private Mobile Radio networks (PMR) over 3GPP technology, namely the OMA Push-to-Talk over Cellular (PoC) service and the more recently standardized 3GPP Mission Critical Push-to-Talk (MCPTT).

In this work we will present initially an overview of how signaling concepts are required in communication networks, and how 3GPP decided to adopt IETF protocols to define the signaling network of 3G and future mobile communication systems. We will justify how this is a good design criterion: rather than reinventing the wheel, it is important that the IP-based mobile networks of today and tomorrow select standard Internet protocols for their operations and services (as opposed to defining a “mobile-specific” IP protocol suite). This is key for convergence, economies of scale and interoperability across networks and services.

Beyond this baseline, we will also explain and justify that it is possible to enhance, enrich and optimize standard IP multimedia signaling protocols and architectures, when deployed over mobile networks, if we enrich such protocols with additional context, or we define optimized architectures for mobile service delivery. We will justify that it is possible to achieve such goals without necessarily “breaking” such protocols and architectures, and without “breaking” the desirable interoperability and backwards compatibility aspects that are key for any service.

In our discussion we will focus such work in the multimedia services mentioned above (PSS, OMA Presence, OMA PoC, 3GPP MCPTT). We will end up the discussion with conclusions and future research work in this area, as 5G network technology unveils.

Enjoy the reading.

2 Introduction. Background and Motivation

2.1 Introduction

The last years of the XXth Century and the first two decades of the XXIst Century have seen a number of technology waves growing exponentially and impacting the life of human beings and Society as a whole. Effectively, if our parents and grandparents experienced the arrival of the radio, television broadcasting and fixed telephony, over the last two decades we have seen arrival and growth of a) the Internet, b) Mobile telephony, c) The Smartphone phenomenon and, lately, d) the boom of high quality, UHD/HD anytime, anywhere, any-device content streaming. With a bit of perspective, the frequency of these waves and the steepness of their growth is accelerating. End users, as well as Operators, Content Providers, Technologists and Technology Companies need to become used to such acceleration… or die.

In parallel with service and technology trends, the type, amount, richness of services that are available to end users, either provided by traditional Operators or new Internet-based platforms, have also exploded. Today users are familiar with mapping and location technologies, group chatting, file and image sharing, instant video services, traffic monitoring, webcam applications, social networking and a myriad of services. This situation also means that the core of Operator's and Service Provider's networks has also had to evolve dramatically since the end of the nineties, to be able to cope with such service acceleration.

In this context, an interesting change of paradigm that has influenced all aspects of communications is the shift from traditional telephony-oriented protocols and architectures developed during the second half of the XXth Century to multimedia signaling protocols developed by IETF and adopted by Standards Defining Organizations such as 3GPP, ETSI or TIA.

With the progressive adoption of Next Generation protocols, new services became possible as we will describe through this text. Among these services, it is worth mentioning the following ones:

- Packet-switched Streaming (PSS), as defined by 3GPP, to deliver multimedia streaming content to mobile devices.
- SIP/SIMPLE based Presence services, to share user's availability status with her peers.
- Group Communication services, as defined by OMA PoC and 3GPP MCPTT, to enable walkie-talkie-like services over 3GPP 3G / 4G / 5G networks.

The bulk of our thesis work will consist on investigating how signaling protocols and architectures can be enhanced to enrich the three services described above. Before that, we will briefly provide some additional context by describing the evolution that enabled the appearance of these new services.

2.2 Signaling Protocols in Telephony Networks

The Plain Old Telephony Service (POTS) was in place for over one century, well into the end of XXth Century. Still today, a significant fraction of the global telephony network relies on analog copper lines. POTS generally consisted on analog voice signal exchanged over copper loops. Historically, the POTS analog telephony service was offered by incumbent Operators universally. By being defined as a universal service, incumbents were urged to make sure that most citizens would have access to it at affordable –typically regulated– price, (mostly) regardless of location. In such environment, with the telephony service being delivered by public companies (or private companies effectively working in a monopolistic environment) competition was almost non-existing and the rate of innovations that hit end users was heavily controlled by such monopolistic stakeholders.

POTS remained untouched for almost a Century since its first deployments in the last quarter of the XIX Century until the first deployments of the Integrated Digital Services Network (ISDN) from 1988 onwards. ISDN relied on a similar architecture, but delivered digitized voice and communication links within the telephony network (i.e.: both in the access links as well as the links connecting local and transport exchanges).

From an architectural perspective, whether POTS or ISDN, the fixed telephone service can be seen as a network of interconnected entities. Effectively, users in a telephone network are generally served by telephony switches. A switch (sometimes referred as an “exchange”, in the context of traditional telephony) is a network entity capable of performing “switching” or “routing” decisions required to connect a call. In general, in order to serve a large number of users, several switches will be required, due to scalability and redundancy purposes. In such cases a phone call between a Caller and a Callee may traverse two or more switches. Switches are interconnected through redundant transmission links that allow setting up a call between any two users within a given telephone network. In turn, interconnection of Operator’s networks as well as international routes enable international calling, so that a call can be established between virtually any two telephony devices in the world.

There is a clear functional difference among the end user media exchanged end-to-end –namely the voice spoken by one user and heard by the remote user– and the set of operations required to interact with the telephone system. Such operations involve setting up a call, tearing down a call, redirecting a call, forwarding it, connecting it into a voice mailbox, … As an example, when a user (A-Party, Caller) starts a call, it places some user-to-network signaling to instruct the network to initiate a call to another telephony user.

When a call needs to traverse several switches, they need to exchange network-to-network signaling to “route” a call until the switch that directly connects (terminates) to the Callee. Once the Callee has been

located an end-to-end path for media (voice) is also established so that communication can flow between the users.

User-to-network signaling refers to the set of messages and information exchanged between an end user device and a network entity in order to perform service layer operations such as starting calls, ending calls, putting voice on hold, redirecting, forwarding or similar operations. Network-to-network signaling refers the set of messages and information exchanged among network entities (e.g.: switches) in order to perform actions upon a call (e.g.: route, terminate, redirect, put media on-hold, …).

In the early analog telephone systems, call setup and signaling operations used in-band signaling in which information was exchanged by playing special dual-tone multi frequency (DTMF) tones into the telephone line. This mechanism allowed advanced users and hackers to place special tones into the media channel and trick the telephony system network. Eventually, in-band signaling was replaced by an out-of-band signaling between switches, the so-called Inter-Office Common Channel Signaling System 7 (SS7) [1]. With SS7 a specific signaling channel between telephony switches would be used. User-to-network signaling would still generally rely on DTMF signaling, due to the large installed base of analog end user equipment and copper lines, but it would not be possible anymore for end users to interact with the separate inter-switches signaling network managed through SS7. Furthermore, the adoption of digital telephony with ISDN would progressively bring the signaling vs. media splitting to the user-to-network connection as well, with the adoption of ITU-T Q.931 standard [2] to signal BRI and PRI ISDN connections (even though the market share of analog-based POTS end user lines remains very relevant at the time of writing).

The following picture summarizes the structure of the telephone network and the media vs. signaling (control) plane splitting in a very simple way.

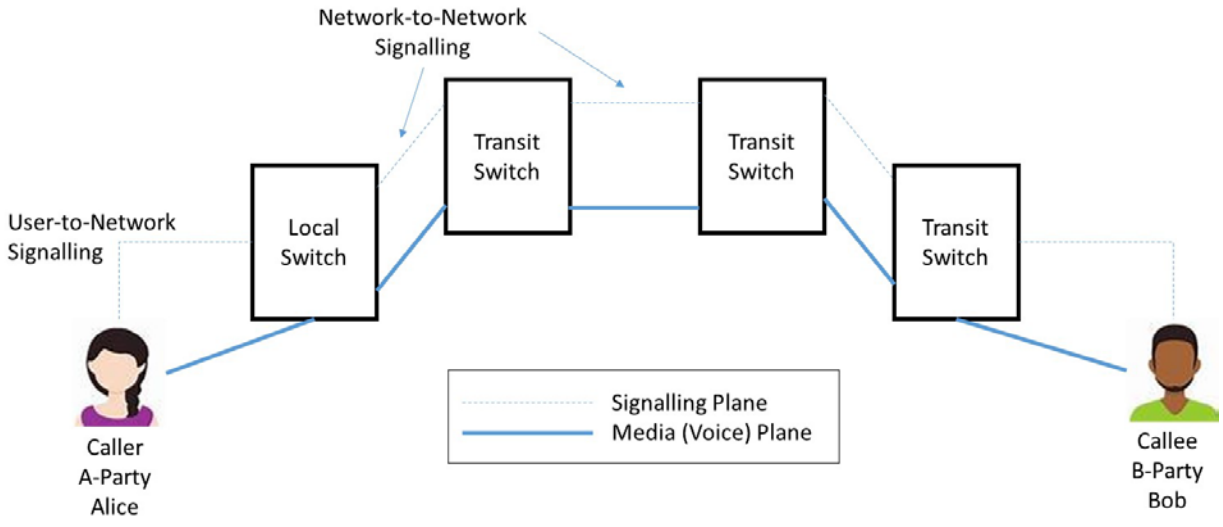

Figure 1. High level architecture of the telephone system. Signaling and Media paths.

While SS7 was the first truly interoperable signaling protocol developed for telephony, the lack of a truly competitive environment in the 80's and 90's did not lead to widespread interoperability among diverse SS7 implementations. The main use case for SS7 interoperability was routing of international calls, for which operators dedicated specific equipment (International Exchanges). On the other hand, internal network signaling would typically be based on regional or local SS7 variants. This situation progressively improved as new telephony operators appeared in the fixed and mobile domain, and SS7 is today the signaling protocol most widely used when signaling circuit switched communications.

The traditional telephony network is generally based on a circuit switched communication paradigm. This means that when a call is established the network reserves a set of resources (circuits) end-to-end until the call is ended. Since resources are limited, the network can get congested. As an example, each E1 ISDN link is slotted to support up to 32 simultaneous calls. Several links can be used in parallel to support more communications. However, it is quite typical that one or more slots are reserved to exchange signaling information between switches, thus the real number of supported calls per link is generally smaller.

As a consequence, SS7 signaling is exchanged over one of the circuits reserved for signaling. Typically, there is at least one signaling channel in every link that interconnects two switches. The combination of all such signaling channels represents a signaling network that overlays the physical structure of the telephony network and the rest of voice circuits.

The SS7 signaling network is considered a packet-switched network. This means that each switch may place SS7 messages on the signaling link, but a packet-switched communication paradigm is run within such switch. In general, each signaling channel may run at 64kbps, which is the native data rate of each channel in an ISDN context (an E1 channel, representing 32 circuits, aggregates 32x64kbps=2Mbps).

Importantly, SS7 (as well as other signaling protocols, such as Q.931) uses a binary structure. This is an important design decision. Since SS7 runs over narrowband channels at 64kbps and it has to handle a large number of calls, protocol efficiency is fundamental. Furthermore, given the limited set of capabilities of the POTS / ISDN network, it is possible to define a binary protocol that supports these features. In such context, expanding the protocol with new capabilities is typically very hard, but this is generally not a problem in a framework where innovation and rapid delivery of new services is not required.

In this context it is important to note that the first digital mobile communication systems such as GSM quickly adopted the inter-office SS7 signaling system to enable communication among mobile telephony switches (also called Mobile Switching Centers, MSCs). This would enable quick and relatively straightforward interoperability between the POTS network (based on SS7 signaling at its core) and the Second Generation mobile communication systems such as the Global System for Mobile communications (GSM).

After their deployment, POTS, ISDN, SS7 and, later on, the GSM system would coexist and interoperate in a relatively smooth way for at least a decade, while first generation mobile communication systems exclusively offered a voice communication service, much like the one delivered by the fixed POTS/ISDN network.

2.3 Next Generation Signaling Protocols for IP Multimedia Networks

In the mid-nineties, with the arrival of the liberalization of the ICT sector, competition and new players arose. In turn, the Internet was slowly but steadily spreading outside of the academic domain and entering both corporate and residential usage.

Availability of Internet connectivity (even if it was only a few tens of kbps), arrival of the first generation mobile telephony and the appearance of non-incumbent new actors in different markets accelerated the need and the quest for a new generation of open standards and innovation in the Internet.

This trend can be observed with the steady increase in the approval of new Internet standards in the form of Request for Comments (RFCs) by the Internet Engineering Task Force (IETF) [3].

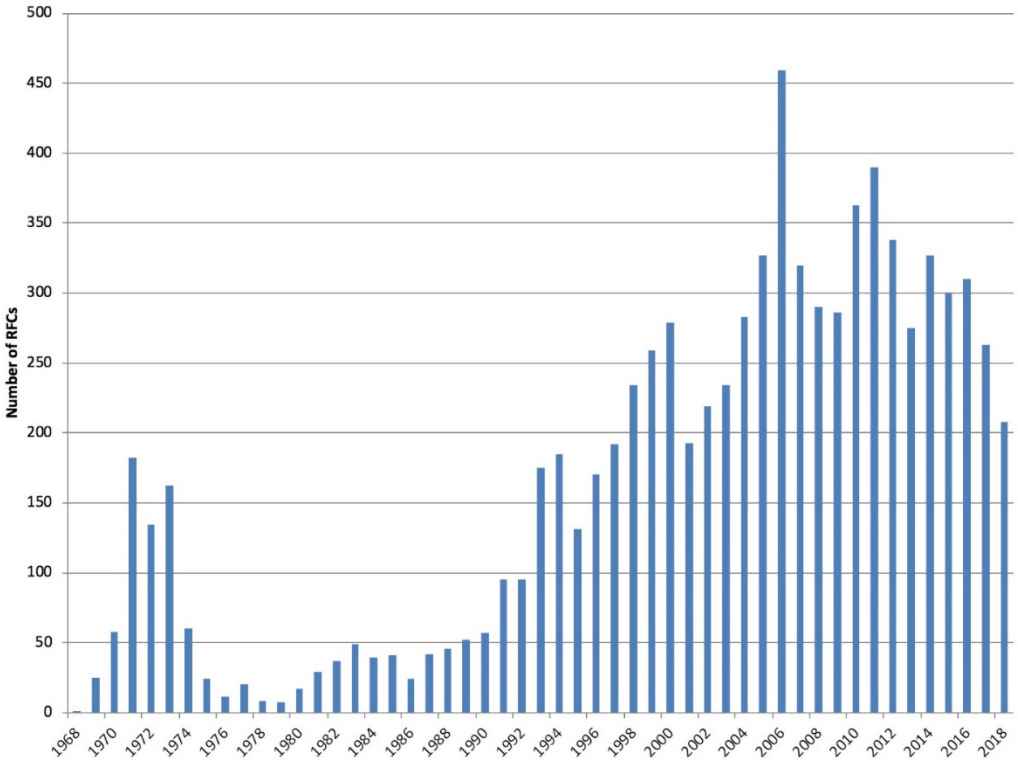

Figure 2. Number of RFCs published per year [4].

With the acceleration of innovation, service delivery and the arrival of new actors, the need for openness and interoperability became crucial [4] in enabling the explosive market growth that would characterize the turn of the Century.

At the mid-90s the first wave of Internet protocols had already been well established for a while, including underlying networking and transport-layer Internet Protocol (IP) [5], Transmission Control Protocol (TCP) [6] and the User Datagram Protocol (UDP) [7]. While the above references date from the early 80s, work in the core Internet protocols had started more than a decade before.

Additionally, the first wave of application-layer Internet protocols and applications leveraging the underlying TCP/IP stack were enabling the rapid growth and success of the Internet. Effectively, protocols such as IRC, SMTP, IMAP, POP, HTTP and the Hyper-Text Markup Language (HTML) developed at CERN enabled users to start the first Internet primitive chat sessions, send messages to individuals or groups in the form of electronic mail, or publish text and simple media information as pages in the world-wide web.

All these ingredients (liberalization, new stakeholders, market growth, spread of the Internet reach to businesses and end users, …) coupled with a progressive improvement in the quality, availability and access speed to Internet connections worldwide, posed a key requirement into the development of new solutions, architectures and protocols.

In that context, one of the key differences between IP based networks and the POTS / ISDN service was about the switching technology at the core of the network. Effectively, IP-based networks rely on the packet switching concept. Packet switching applies to all IP traffic, including signaling as well as media. In such context, there is no fixed “circuit” constraint to the communications exchanged within the infrastructure. While –in certain contexts– packet switching may be less efficient than circuit switching, for a given voice communication, packet switching in general offers far greater flexibility, efficiency and redundancy than traditional circuit switching.

In such context, and with the ever increasing capacity of Internet connections, a new generation of signaling protocols was needed. Some of the characteristics that should be met by such new generation include:

a) Backwards compatibility. The new generation should support at least the same set of telephony services traditionally supported by SS7 / Q.931 (call setup, call finalization, call forwarding, voice mailbox access, …).
b) Support for new services. Next Generation signaling protocols should support the type of services enabled by the Internet. In addition to traditional calls, these include status and context sharing

(Presence), Rich Messaging (Text, Images, Files, Multimedia), Video calling, Video content delivery, …

c) Flexibility and Extensibility. Rather than protocol efficiency (which is fundamental in a narrowband environment, such as SS7 networks), in an IP Multimedia context, a key requirement is flexibility and extensibility. In order to support forward compatibility, it is more important for next generation signaling protocols to be extensible and flexible rather than to have an extremely efficient binary implementation.

d) Access agnostic and network forward compatibility. Next Generation IP Multimedia signaling protocols should be able to cope not only with the network technologies available at the time when they were being standardized, but also support future technology evolution such as fixed network evolution (e.g.: xDSL, FTTH, …) or mobile network evolution (e.g.: 3G, 4G, 5G, …).

With such driving forces, the IETF was the right environment to enable the creation of the suite of Next Generation Multimedia Protocols for IP networks. A set of leading academic and industry experts, particularly Henning Schulzrinne and Jonathan Rosenberg, pioneered the creation of the protocols that would be adopted and extended over the next thirty years.

Four key deliverables by such talented group of authors include:

- The Session Initiation Protocol (SIP) [8]. SIP became the signaling protocol for multimedia sessions, from VoIP to video calls, Messaging, Presence, File Sharing, Location Sharing, … SIP steadily but surely became a successor of traditional signaling protocols in circuit switched networks.
- The Session Description Protocol (SDP) [9]. SDP decoupled the actual format of the communication signaled through SIP (e.g.: audio, video, CODECs used, …). While SDP has proven to be a too cumbersome and error prone protocol, the design decision to split media signaling from session signaling proved key in enabling the fast adoption of new services and improved audio and video CODECs over time. Should such signaling had been built into the core of the session signaling protocol (as is the case in SS7) such evolution might have been much more complicated.
- The Real-time Transport Protocol (RTP) [10]. RTP consisted on the key transport layer to enable delivery of real-time media (e.g.: audio spoken by participants in a call or a conference). Effectively, audio communication in an IP network is transported based on a packet-switched paradigm, which may lead to packet loss or packet reordering. RTP delivered the necessary functionality to enable packet loss detection and packet reordering, and enable reliable audio communications and playback over packet-switched IP.

- The Real Time Streaming Protocol (RTSP) [11] [12]. RTSP is also a signaling protocol specifically designed to enable content delivery from a content server to a streaming client. It was mainly conceived to enable an audio / video streaming service, as a precursor of today's IPTV technologies. Streaming sessions are generally signaled through RTSP, while actual media delivery is based on RTP.

Note that other protocols have been developed by IETF in relation to delivery of multimedia services. However, for the sake of this thesis we will focus on the main ones mentioned above. We will outline some of the characteristics of these protocols:

The signaling protocols layer mentioned above (SIP, SDP, RTSP) comprises three text-based protocols. In particular, SIP and RTSP leveraged on the success of other existing text-based protocols at the time, which had already proven their success, such as SMTP (mail) or HTTP (web). Effectively, while binary protocols are more efficient from a link usage perspective, the IETF decided –in spite of the fact that first early drafts of RTSP– that future signaling protocols would be text-based, they would follow a request-response paradigm and would replicate the structure of similar protocols such as HTTP, with a request line (the protocol line that defines the main operation that is intended with a given protocol primitive) and a set of headers that provide additional information. This structure also allowed for easy extensibility of the protocol, since addition of new headers is always possible (nodes that do not understand a given header may ignore it, while specific headers such as `Requires`: or `Supported`: can be used to ensure that the appropriate capability set between clients and servers is available to deliver a given service).

The following figure shows the basic structure of a SIP message (SDP content excluded) and displays the "request-line" / "headers" structure.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

Figure 3. Example SIP INVITE message [8].

At the time, as the IETF community was progressing with the definition of the next generation of multimedia protocols, the 3GPP was also progressing in defining what the future of mobile networks and

mobile services would be. We will provide a quick overview of the convergence of these two efforts in the next section.

2.4 Adoption of Internet Multimedia Protocols in the 3GPP framework

As early as 3GPP Release-6, 3GPP identified that the future evolution of 3G networks should converge toward a single network technology based on IP packet switching. This was a big shift when compared to 1G, 2G and the first 3G releases, where effectively two separate "infrastructures" were defined, one (circuit-switched) to deliver voice services, and one packet-switched to deliver data services and Internet connectivity. At some point it was obvious that future 3GPP networks should converge toward a single technology capable of delivering real-time voice communications over a packet-switched domain.

As soon as such decision was made, the need to define a packet-switched architecture to support voice communications first and any type of multimedia services later on raised. In 3GPP Release-6 the so-called IP Multimedia Subsystem (IMS) was defined, and SIP was selected as the core signaling protocol in IMS.

The IMS architecture is displayed below.

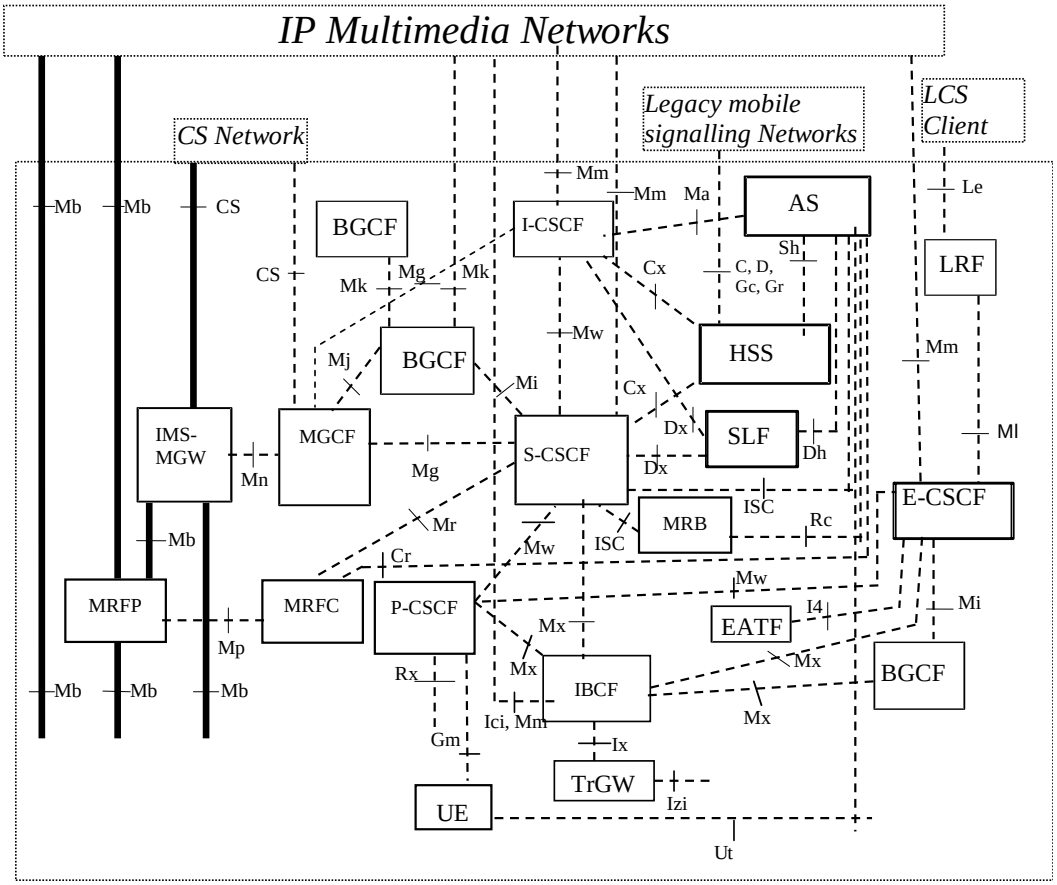

Figure 4. 3GPP IMS architecture [13].

We will not go into details defining the specifics of the IMS architecture. We will briefly present some key concepts.

- In an IMS architecture, the UE (the handset that connects to the 3GPP network through the air interface) runs an IMS capable client. The client connects to the IMS network through the standardized Gm interface.
- The IMS generally consists of a set of SIP Proxies in charge of specific tasks. While the IMS defines several functional nodes, whether they are implemented separately or converged into a physical platform is left for implementation decision. These are the key proxy elements defined in an IMS architecture.
    - The HSS is the Home Subscriber Server, which hosts subscriber information.
    - P-CSCF. The (Proxy)-Call-State Control Function terminates Client-Network signaling. It keeps a secure connection to the UE and may handle some additional aspects such as compression of signaling messages (SigComp) or NAT Traversal (e.g.: sometimes the P-CSCF is implemented through a SIP B2BUA SBC function).
    - S-CSCF. The (Serving)-CSCF is the one who actually serves a given user in her Home network. The S-CSCF performs authentication and authorization. This proxy is the one that provides access to the SIP services that are specifically available to a given user (e.g.: depending on her user profile, subscription status, purchased services, …). The S-CSCF retrieves subscriber information from the HSS using a DIAMETER-based connection.

There are other nodes defined in the IMS architecture, even though we will not devote a detailed description. Some of the main ones include:

- The I-CSCF is the Interrogating-CSCF, which is used when interconnection of different networks is required. The I-CSCF delivers a SIP routing / redirection capability.
- The E-CSCF handles Emergency calls. This node is in charge of handling the special procedures required to serve 911/112 calls, which include locating the calling user and routing the call to the appropriate PSAP.
- The MGC is in charge of handling voice interconnection with the PSTN/ISDN.
- The MRFC/MRFP is in charge of handling in-call tones, announcements and DTMF digits processing.
- Application Servers are SIP applications that are deployed on top of the IMS to deliver SIP-based services (e.g.: Presence, Messaging, Push-to-Talk, VoLTE, …). It is quite frequent that applications behave as B2BUAs, which means that –from the SIP perspective– that the application server “terminates” SIP signaling received by one user and “initiates” SIP signaling toward other users “on behalf” of the initiating user. By

contrast, SIP Proxies (e.g.: CSCF nodes) simply forward SIP messages generated by a UE, but cannot apply application specific policies beyond the routing logic (e.g.: as a maximum, a SIP proxy can drop a message if a user is not authorized to access a service, but the Proxy cannot initiate a new SIP dialog or transaction as a consequence of a received SIP message).

The following diagram shows how a VoLTE call between two UE's is routed. Note that the diagram is heavily simplified, assuming both users are served by the same IMS network. Only routing of the initial INVITE request is displayed. Routing of subsequent responses and messages may follow several different options depending on IMS routing configuration.

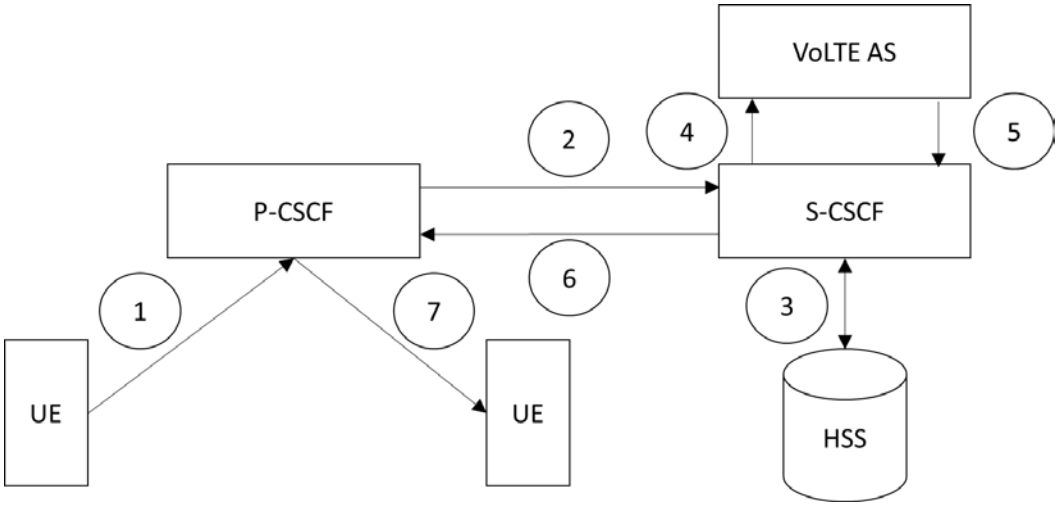

Figure 5. Example simple SIP routing in an IMS network.

We shall assume that the SIP INVITE message depicted in Figure 3 is sent through an IMS network. While real IMS messages are significantly more complex, this will serve the purposes of our example. In this case the INVITE message is sent by a calling user (e.g.: Alice) to reach another user (Bob) and establish a multimedia session. In the IMS domain, multimedia sessions are managed by a SIP applications called the Voice-over-LTE Application Server (VoLTE AS). In such simplified case, the routing of the INVITE message can be conceived as follows:

1. The initial INVITE message is generated by the UE and sent to the P-CSCF (note that the UE has been previously provisioned with the address of the P-CSCF as its "outbound proxy").
2. The P-CSCF decompresses and deciphers the SIP INVITE message and forwards it to the S-CSCF.
3. The S-CSCF, which is in charge of serving Alice (and, for the sake of our discussion, also Bob) sends a DIAMETER Query to the HSS to determine whether Alice is subscribed to the VoLTE service.

4. Since Alice (and Bob) is subscribed to the VoLTE service, the SIP INVITE message is routed to the VoLTE AS.
5. The VoLTE AS receives the INVITE message, it determines that Alice is trying to call Bob and forwards back the request to the S-CSCF.
6. The S-CSCF checks Bob's subscription status (to the S-CSCF, even though this is not displayed in the picture, for the sake of simplicity). The S-CSCF determines that the INVITE message should be routed to the P-CSCF node that is currently serving Bob.
7. The INVITE message is received by the P-CSCF, which encrypts and compresses the message to send it through the air interface to Bob's UE. Bob's UE will start ringing to alert the end user, who will eventually attend the incoming call.

Note that we have made a number of simplifications in the above flow, including the fact that the VoLTE AS would generally behave as a B2BUE and should have generated a new INVITE request, as opposed to forwarding the incoming one (note that we would have intended to simply forward requests we could have done it directly from the S-CSCF without having to involve an Application Server). However, this simplified presentation serves the purpose of our discussion, which is providing a high level short description of the IMS concept.

We will provide a brief summary of some of the main features of the IMS architecture that are relevant for the sake of our research and subsequent chapters.

- The IMS represents the core of the Next Generation signaling network that will serve 4G, 5G and beyond core and multimedia services.
- The IMS represents an evolution from traditional signaling concepts developed for fixed telephony, such as Q.931 or SS7.
- SIP has been the selected protocol for the signaling plane in IMS. SIP is used both for User-to-Network signaling (UE – P-CSCF), Network-to-Network signaling (Inter-CSCF communications) as well as IMS-to-Application signaling (e.g.: S-CSCF-AS ISC interface).
- SIP and IMS follow a packet-based paradigms and are deployed over IP.
- While SIP and IMS define a set of core Operator services (e.g.: VoLTE/MMTEL), SIP and IMS can be extended to support new multimedia services.

- IMS and SIP have become the default signaling platform in 4G networks, where both voice and data are carried over an IP domain and no specific network infrastructure is deployed to deliver any circuit-switched-based service[1].

After this quick overview of the IMS concept, architecture and signaling protocol selection, we will go into more details about the specific packet-switched mobile services that have been subject of our attention through the thesis work.

2.5 Definition of new 3GPP multimedia services: Media Streaming and Group Communications

In the previous section we have provided an explanation of the evolution of 3GPP multimedia services toward IP-based packet-switched technology, which in turn required the usage of a new signaling protocol. This led to the definition of the IMS architecture and the selection of SIP.

In parallel with this approach, 3GPP also leveraged another interesting text-based, flexible, client-server, extensible signaling protocol to define how a specific service would be deployed over 3GPP networks. Effectively, with video gaining importance in the Internet, it was clear that video would play a key role on future 3GPP networks as they were slowly evolving from voice-only to narrowband data, to enhanced mobile broadband. In such context, at the same time IMS and SIP were selected for multimedia user-to-user communications, 3GPP selected RTSP as the signaling protocol for future server-to-client multimedia streaming content delivery.

While this selection will evolve through 3GPP releases (e.g.: with IMS gaining importance in the content delivery domain and DASH progressively replacing RTSP), RTSP has been the core signaling protocol for content delivery services since its incorporation into 3GPP in Release-4. The RTSP-based Packet-switched Streaming (PSS) service is an example of one service that can be plugged over a 3GPP network to deliver value to end users. This is one of the few exceptions (together with MCPTT) where 3GPP has standardized the application layer that would sit on top of its infrastructure. The main reason of doing so at the mid-

[1] Note that while 4G networks are based on a packet-switched-only paradigm, regular voice calls between 4G users are generally run over 3G, unless the operator has deployed full VoLTE capabilities in the network. This does not mean that a circuit-switched domain is used in 4G. Rather, if VoLTE is not available in the network, the UE will downgrade connectivity to a 3G connection (which does have a circuit-switched domain) and run the call over a 3G circuit-switched connection. When the voice call is over the device will automatically upgrade network connectivity to (packet-switched) 4G if coverage is available. This is the so-called 4G Circuit-Switched Fall Back (CSFB) defined in 3GPP for 4G networks that do not support VoLTE yet.

2000s was ensuring availability of standardized multimedia services on top of the first 3G infrastructures that operators would deploy in the subsequent years.

As 3GPP releases have evolved, so has 3GPP PSS. In particular, it has incorporated a number of features to better interact with the underlying 3GPP network and ensure that network resources are used in the best possible way, while guaranteeing best possible Quality-of-Experience (QoE) of the delivered multimedia session. In this framework, there are a lot of opportunities for researchers to enhance or further evolve the PSS concept over time, as we will see through the rest of the document.

The other two main services that will be subject to research through this thesis work are IMS-based, SIP-based services, namely:

- The OMA Presence service based on the SIP/SIMPLE framework. This service was conceived for users to share status, location, context… with their connections and create a context for multimedia communications. Very much like social networking, Presence was conceived as the "dial-tone" for Next Generation multimedia communications, as it would provide relevant context information prior to setting up a session with a user or a group of users.
- The OMA Push-to-Talk over Cellular (PoC) and the 3GPP Mission Critical Push-to-Talk (MCPTT) service. This is a SIP/IMS-based service that emulates traditional walkie-talkie communications over a 3GPP network. As an example, if we think about the call between Alice and Bob, we can conceive the basic MCPTT use case as follows:
    1. Alice wishes to talk to a group of peers in a half-duplex fashion and places an INVITE request towards a Group Identity (e.g.: Alice's Friends Channel) as opposed to Bob's identity in the example above.
    2. The SIP INVITE message is routed to the Group Communications Application Server, which will trigger a new INVITE message to all group members, who can accept and join the session.
    3. When a user presses a PTT button (which can be a physical button or emulated in the UE User Interface) she broadcasts audio to all connected group members. Actually audio is sent upstream to the Group Communications AS, which replicates it downstream to all group participants.

The details of the MCPTT service are described in detail in section 5. For the sake of our purpose it is important to understand that MCPTT is an IMS/SIP-based service that is oriented toward supporting instant, real-time communications among groups of users, emulating the traditional walkie-talkie concept.

Importantly, note that, should the IETF have selected a binary, non-extensible protocol, as a simple replacement of SS7, the definition of new services and architectures such as 3GPP PSS, OMA PoC or

3GPP MCPTT, would have been much more difficult, or would have faced more severe interoperability problems (which would have impacted industry adoption). In this context, the selection of text-based, flexible, extensible protocols such as RTSP and SIP to serve as the foundation of these services proved to be a key one, that would ensure forward compatibility with future releases, features and services.

2.6 Overview and thesis work motivation

In the previous section we have outlined the main 3GPP services based on IETF multimedia signaling protocols that are subject of our study, namely 3GPP Packet-switched Streaming, OMA SIMPLE Presence and OMA PoC / 3GPP MCPTT.

While OMA PoC, Presence, Instant Messaging and XDMS were mainly defined as (person-to-person) communication services, it is interesting to note that, from a protocol perspective, they share a number of commonalities with the services and architectures defined for delivery of multimedia content as defined in the 3GPP PSS service.

Firstly, both 3GPP PSS as well as (e.g.) OMA PoC and Presence comprise two separate communication planes, namely: the signaling plane (sometimes also referred as the “control plane”) and the media plane (sometimes also referred as the “user plane”). The control / signaling plane handles the service logic, while the media / user plane consists on actual media delivery (e.g.: audio, video, pictures, text, …) as determined by the control plane transactions.

While PSS uses IETF-defined Real-time Streaming Protocol (RTSP) [11], PoC and Presence use IETF-defined Session Initiation Protocol (SIP) [8]. Both RTSP and SIP are text-based signaling protocols and follow a similar request / response approach (inheriting from some other successful IETF protocol designs, such as HTTP or SMTP). Secondly, for audio and video delivery, both SIP-based and RTSP-based services use the Real-time Transport Protocol (RTP) as defined by IETF [10].

So, as mentioned, both SIP and RTSP are the “signaling” protocols used in (among others) PSS, Presence, PoC and MCPTT services. As the signaling concept in traditional telephony networks, signaling protocols in the context of IP multimedia services, carry the “intelligence” of the service and enable rich capabilities and interaction between users and the service and among users themselves. Since signaling protocols define service delivery, their performance generally determines key performance indicators such as session setup time or service interactivity. Furthermore, signaling protocols, such as SIP and RTSP and many others, are client-server request-response based, which means that there must be one or more “handshakes” between a client and a server prior to service delivery. At the end of such handshake(s), the Client(s) and the

Server(s) have a common view of what has actually been requested and requested service delivery (e.g.: streaming media delivery, a VoIP call, an MCPTT group call) can take place.

Due to their client-server and transaction-based nature, careful design and optimization of signaling protocols is of importance. Effectively, even if the performance of early 2.5G and 3G networks has dramatically evolved until today's 4G and 5G networks (both in terms of throughput as well as round-trip-time (RTT), with gains of roughly two orders of magnitude in both areas over the last ten years [14]), cellular systems have different levels of granularity to deliver different types of traffic, since network resources, bandwidth and wireless spectrum are a scarce resource that has to be handled properly. In this context, ensuring the proper interaction between signaling protocols and the wireless network is of importance.

As an example, today it is possible to stream content to an LTE device at rates of 100MBps with <40ms round-trip-delay. The Bandwidth-Delay Product (BDP) has hence dramatically increased when compared with first 2.5G or 3G systems. An order of magnitude of increase in total throughput and decrease in RTT is expected with 5G. Regardless of such evolution in network technology and performance as specified by 3GPP and deployed by Mobile Network Operators (MNOs), the fact that multimedia signaling protocols (e.g.: RTSP, SIP) are based on a client/server, request/response paradigm lead to situations where unnecessary delays are incurred. Today's cellular networks, whether 2G, 3G or 4G, may suffer well-known impairments that affect data transmission, including interference, noise, cell breathing, congestion and distance attenuation.

While substantial literature exists covering the optimization of TCP and transport-layer protocols to maximize throughput over a link with certain transmission characteristics, the situation is different when it comes to signaling protocols: effectively, when using protocols such as SIP or RTSP or any other signaling protocol, the request / response nature is built in a way that media transmission or session setup cannot happen until both communication endpoints (e.g.: a client and a server) have exchanged the necessary information to complete session negotiation. This information may range from type of media (audio, video, text, data, …), media CODECs to be used, CODEC rates and CODEC parameters, transport-layer protocol (RTP/UDP), transport-layer ports and so on.

In contrast, in a bulk media delivery context, the first interactions of a resource consuming task may not play a relevant role, since we are looking for sustained bandwidth and capacity over a relatively long period of time. On the other hand, in a signaling context, the initial protocol interactions determine key performance indicators such as session setup time or channel switching time, as mentioned above.

In a nutshell, the fact that signaling protocols operate in a request / response fashion that requires several RTTs to successfully complete a transaction (e.g.: "start call", "receive media stream", "talk to the group",

…) means that today's SIP-based and RTSP-based services cannot take full benefit from the bulk transmission rates enabled by IP mobile networks. In such "chatty" situations, the delay incurred to complete each transaction may play a significant role, more so when the demands and expectations from today's users may exceed the ones from the previous decade, in terms of Quality-of-Service (QoS), Quality-of-Experience (QoE) and responsiveness of today's content and communication services.

In this framework, the optimization of signaling plane protocols and architectures in order to minimize the amount of information and the number of messages required to complete session setup is as important with any underlying network technology. Effectively, when content streaming was delivered in early 2.5G and 3G networks, session setup delay could lead to users dropping the session or losing interest, as they were not willing to wait 15-30 seconds prior to being able to visualize content. Today, when users are used to being able to receive media content simultaneously through several devices (e.g.: IPTV, Tablet, Smartphone, Laptop, …) users may be dynamically switching among several contents in the same instant.

Through the Thesis work the author has focused on the optimization of such multimedia and communication services, as defined by 3GPP and OMA, over 3GPP-defined networks. The underlying element across all Thesis work has been the optimization of the signaling plane of multimedia and communication services over wireless networks.

In the particular case of Mission Critical communications, which has been the subject of study in the final part of the Thesis work, an MCPTT user is not willing to wait more than 300ms to receive a (signal) confirmation that he can talk when pressing a PTT button [15], or have his voice transmitted to a remote end.

In a nutshell, while network performance and capabilities have improved dramatically from 2G to 4G/5G, so have users' expectations as well.

Interestingly, there is a lot we can do to optimize and improve user experience. First and foremost, by design, signaling protocols need to be efficient and complete the tasks they are required to accomplish in the most effective way. On top of such baseline (e.g.: best practices) there are situations in which the service and the network may jointly have some specific "context" that, if used properly, can be used to further enhance protocol / architecture design and, hence, optimize service delivery and user experience. As an example, if a content provider encodes thousands of video clips with the same encoding settings template, it may be useful to store the common encoding information in the mobile client, so that when a user wishes to switch among different content from the same provider, redundant information does not traverse the air interface unnecessarily.

In such framework our goal is to leverage extended context information to optimize signaling plane protocols when setting up and modifying multimedia and communication sessions based on SIP and RTSP over 3GPP networks.

In all situations described above, irrespective of the throughput or delay of the underlying cellular technology in use (2.5G, 3G, 4G, 5G), the less information we exchange over the cellular network and the lesser number of RTTs we can use to complete a transaction, the quicker experience we will provide to either a consumer wishing to switch a multimedia video channel transmission, or to a business user willing to setup a multimedia conference or to a First Responder in the need of talking to a team while in imminent peril. Our goal is precisely to optimize those exchanges of information by reducing the number of transactions or the amount of information carried by each individual transaction. Such reduction will be compensated, in turn, by (e.g.) caching frequently used information or by deriving implicit information from the surrounding context.

Such enhancements can be achieved either through protocol optimization, as we will see in some examples or, in other cases, by enhancing and extending service architecture. An example of this later case can be found in chapter 5, where MCPTT service agility and reaction time can be improved by delivering automated MCPTT client ”bots” as close as possible to the end user.

Importantly, during all the Thesis work we have consistently taken into account two underlying guiding principles to the optimization of multimedia signaling protocols applied to 3GPP services. Such guiding principles can be stated as “Do more with less” and “Keep it simple”. As the reader will see, our approach is generally to “achieve more without modifying the core protocol”, with [16] [17] being two good examples of this, or “maximize gains while applying minimal protocol changes without breaking backwards-/forward-compatibility”, with [18] [19] being good example of the later.

The rest of the document provides an overview of all the thesis work carried out. The main structure of the document is as follows:

- Chapter 3 provides an overview of preliminary work completed by the author in evaluating transport-layer protocols, that served as a good foundation to later on evaluate and enhance application-layer signaling protocols’ performance and architectures.
- Chapter 4 focuses on the proposed enhancements related to the 3GPP Packet-switched Streaming Service.
- Chapter 5 comprises a description of IMS- / SIP-based services, and proposed enhancements of protocols / architectures associated to SIMPLE Presence, OMA PoC and 3GPP MCPTT.
- Chapter 6 contains a recap and conclusions, and outlines potential future lines of research.
- The References chapter contains all bibliographic sources for the Thesis work.

- Annex A contains some example RTSP signaling flows according to the enhancements proposed by the author. This material complements chapter 4.
- Annex B contains a section from 3GPP TS 26.234, to ease comparison of a new RTSP procedure proposed at 3GPP / IETF, when compared with the Early Setup concept described by the author in chapter 5.
- Annex C contains a summary of all the work published in the scope of this Thesis.

3 TCP/IP traffic evaluation over wireless networks

3.1 Introduction

Prior to starting the core of our thesis work around signaling protocols and architectures over 3GPP networks, it was of interest to characterize the performance of underlying transport and network layer protocols over such 3GPP networks, namely TCP, UDP and plain IP. This is of interest in order to understand the performance of raw packet flow over the network, and will prove an invaluable baseline into the subsequent work where our focus moved some layers above in the TCP/IP protocol stack.

The goal of evaluating the performance of TCP/IP protocols over wireless networks to better understand potential challenges and impairments that may help define some of the key strategies to optimize signaling protocols over wireless networks.

Hence, in this chapter we will provide a brief overview of the tasks performed by the author in the scope of this Thesis work to characterize the performance of underlying TCP/IP protocols over wireless cellular networks.

3.2 TCP/IP traffic evaluation over wireless networks

3.2.1 Introduction

This section will initially focus on the performance of TCP/IP protocols over 3G networks, with some conclusions derived for 4G and 5G in the subsequent chapters.

After the launch of 2G networks in the early 2000's, WCDMA / UMTS became the global standard for 3G enabling the delivery of truly multimedia services over mobile networks. While the arrival of mobile broadband would require the delivery of 3.5G HSPA and, even better, 4G and LTE-Advanced, the launch of the first 3G networks allowed operators to offer IP-based services, including some degree of IP multimedia such as video streaming or file sharing over 3G.

The transmission of information over wireless networks incurs a number of potential challenges which are intensely linked to the nature of the wireless link, including fading, interference, multipath transmission, signal loss. Such conditions are also dynamic due to the movement of users following pedestrian or vehicle movement patterns.

On the other hand, the early design of IP networks and protocols focused on impairments such as packet loss in the presence of network congestion. Since the loss patterns and optimal recovery strategies under packet loss due to bad link quality vs. network congestion, might differ, our main interest was analyzing in

detail the performance of TCP/IP protocols over early 3G networks as well as wireless networks based on WiFi technology.

3.2.2 Data transfer performance evaluation over 3GPP networks

With the above background, we decided to evaluate the performance of IP communications over 3GPP networks. We managed to have access to an early pre-commercial 3G network composed of 2 base stations, which allowed us to perform real-life testing over a real infrastructure deployment. Our initial interest focused on evaluation of basic performance over such infrastructure in a static scenario to characterize achievable bandwidth.

Initially we carried out a detailed study of a pre-commercial WCDMA infrastructure, including evaluation and characterization of the performance of the system under various network, user behavior and transport layer protocol configurations. In such environment, the effects of pedestrian and vehicle-based mobility under two base stations without soft / softer handover were evaluated under realistic conditions.

The picture below [20] outlines the test scenario, which consisted on two UMTS base stations. Several pedestrian and vehicle routes were evaluated, and A and B were determined as the areas where handover occurred when moving A → B and B → A respectively.

Figure 6. Test scenario evaluating WCDMA network.

During the work described in [20] [21] a pre-commercial 3G UMTS network consisting of two nodes was tested including pedestrian and vehicle-based scenarios. Some important conclusions were derived out of the testing.

A first part of the study focused on the evaluation of the pure IP transmission evaluation over the 3G network setup described above.

Uplink and Downlink IP level throughput. Static scenario		
MTU (bytes)	**Uplink throughput (kbps)**	**Downlink throughput (kbps)**
128	45.14	230.50
256	51.40	318.47
512	55.64	356.76
1024	56.61	355.35
1500	55.83	370.95

Figure 7. Throughput evaluation in a pre-commercial WCDMA network.

As it was expected, it was also possible to compare RTT delay performance under different MTUs. Since WCDMA uses a link protocol that "breaks" upper layer packets into smaller link layer frames, it is no surprise that larger IP packets lead to longer RTT delays (since such packets would require additional lower layer frames to be delivered). Furthermore, since the link layer protocol has an internal retransmission capability, larger frames may experience delay due to link layer retransmissions as well. More details on the impact of this behavior are expanded in chapter section 4.6.6.

With WCDMA having a significant end-to-end delay it was no surprise that larger MTUs would lead to larger RTTs. An important remark was, however, that when increasing MTU by 1000% (1,500 bytes vs. 128 bytes) roughly a 100% RTT increase (430ms vs. 205ms.) was observed. Hence, in relative terms it can be concluded that it is much more efficient to minimize the number of RTTs and maximize the amount of information that is embedded into each packet, than to cause frequent request-response interactions based on small packet exchange.

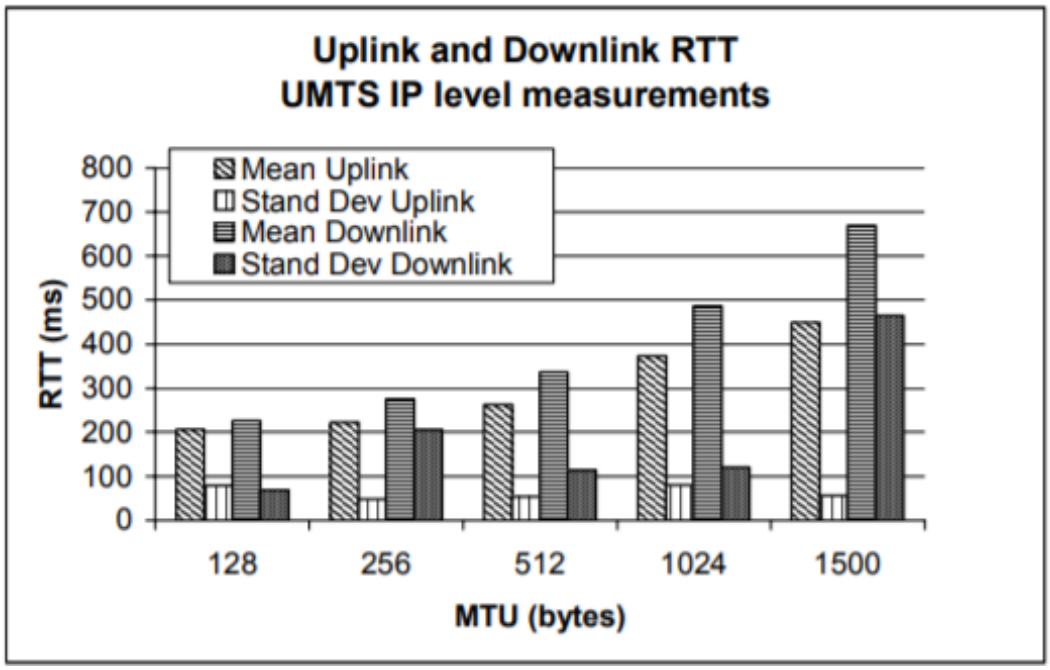

Figure 8. RTT measurements of uplink and downlink ping trials over UMTS.

In addition to the effects of MTU size in RTT and throughput, we also evaluated the effect of hard and soft handover upon UE mobility.

When soft handover was not enabled in the 3G network and only hard handover was supported, some packets were lost and, additionally, packet inter-arrival delay increased until the connection was fully recovered after completing the handover. These effects are depicted below, showcasing delay increase ("DELAY") as well as loss of packet bursts ("GAP").

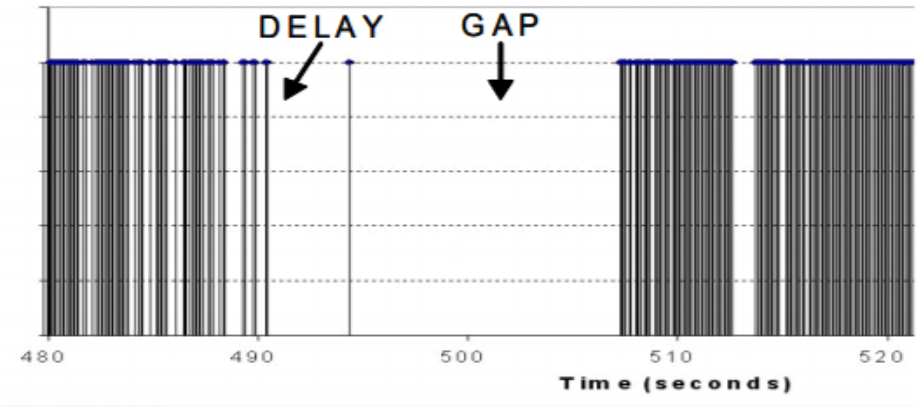

Figure 9. Effects of hard handover in packet arrival over UMTS.

On the other hand, when soft handover was enabled smooth packet inter-arrival delay was experienced when sending a continuous stream of packets over the link, as depicted in the figure below.

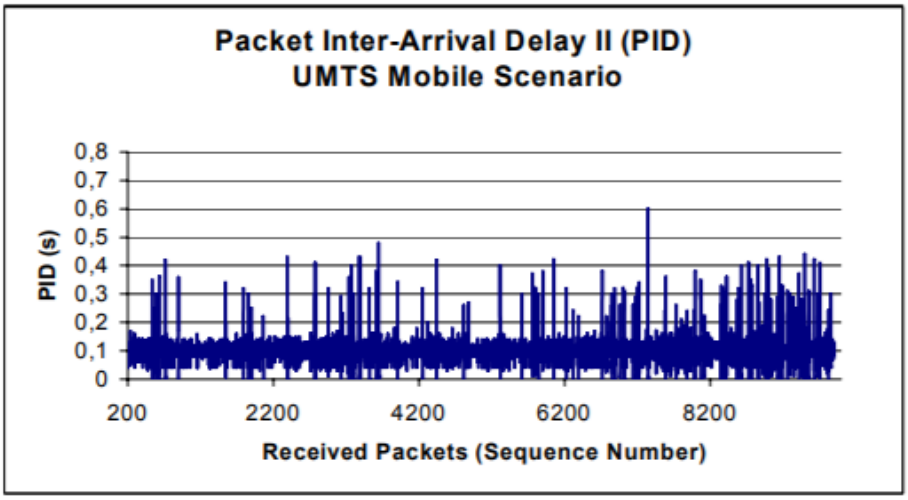

Figure 10. Packet inter-arrival delay under soft handover over UMTS.

In conclusion, in addition to characterizing raw bandwidth levels that can be achieved over such pre-commercial network, we also confirmed that network impairments such as mobility, handover and packet loss can impact packet rate and delay performance of IP transmissions.

3.2.3 TCP performance aspects

After the evaluation of pure data transfer performance over the mobile link (which were based on UDP with no retransmissions, to simply investigate the achievable throughput of the wireless interface) we also investigated the performance implications of interactions with slightly additional logic. Hence, we started to investigate the throughput implications when running a protocol with additional capabilities, such as TCP. In the case of TCP, mechanisms such as the connection setup (three-way handshake), retransmission capabilities and associated variants (e.g.: slow start, fast retransmissions, timers, …) may behave differently over a mobile connection subject to the impairments mentioned above.

This type of analysis provided a good background for the future evaluation of multimedia signaling protocols, where the protocol logic itself may determine how the protocol interacts with the network capabilities and, hence, what is the performance level that can be achieved when running such protocols over 3GPP networks.

Our study focused on two key parameters of the TCP protocol stack, namely:

- The (TCP) Maximum Segment Size (MSS). This is the size of payload information sent at the TCP layer. In order to avoid fragmentation, the MSS (plus TCP headers) should not exceed IP MTU, otherwise throughput may be degraded. On the other hand, since TCP implements retransmissions at the “segment” level, lower MSS values may allow faster recovery from packet losses. Note also that in a 3GPP environment user data may be encapsulated over TCP segments, which are in turn

encapsulated over IP packets, which are in turn segmented and encapsulated into link layer frames. The interaction of segmentation and encapsulation at several layers may lead to unexpected performance situations.

- The Maximum Transmission Reception Window (RWIN). This parameter defines the maximum amount of information that can be sent without acknowledgement in a TCP communication. For a link with a given BDP characteristic, in order to fully utilize the link capacity RWIN should be equal or bigger than the BDP. On the other hand, too big RWIN values may lead to too many segments being lost depending on the retransmission strategy implemented at the TCP level.

Taking these parameters into account, we measured the performance level of three different MSS configurations and three different RWIN configurations. Uplink throughput measurements are depicted in Figure 11 while downlink static scenario is presented in Figure 12.

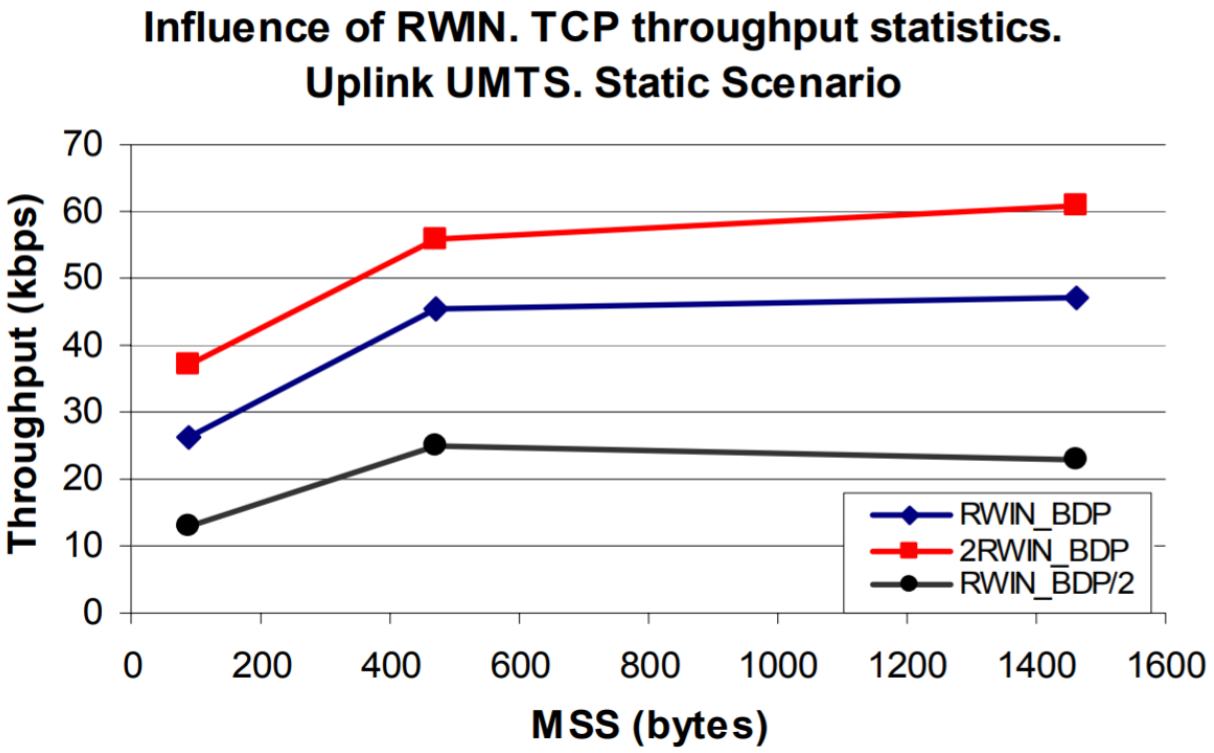

Figure 11. MSS and RWIN influence on TCP throughput over UMTS. Uplink static scenario.

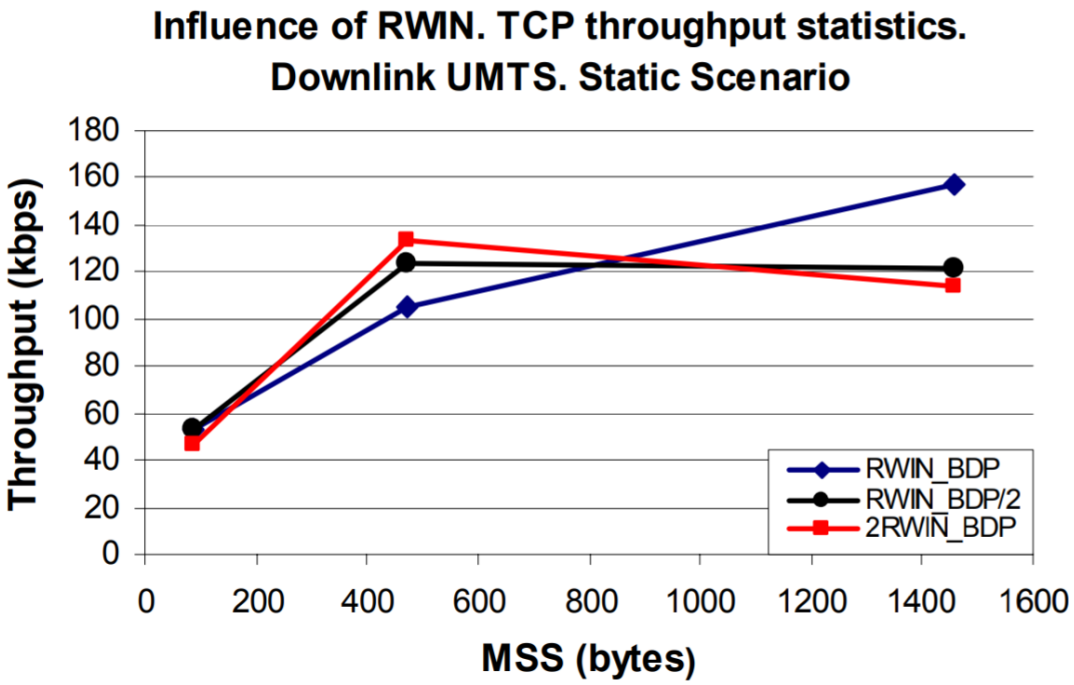

Figure 12. MSS and RWIN influence on TCP throughput over UMTS. Downlink static scenario.

On the above figures we can observe that highest MSS values yield best throughput performance in general. However, the gain from 512B to 1,500B is marginal, suggesting that with MSS values close to MTU values some TCP segments may be fragmented at the IP layer, hence leading to suboptimal performance. A conclusion could be derived to use less aggressive MSS values, in the range of 1,000B probably.

When it comes to RWIN, in the uplink case RWIN>=2BDP yielded to best performance, while in the downlink case it was RWIN=BDP. While RWIN>=2BDP seemed to be initially the optimal value, results did not seem to confirm this impression. Note also that results variability was also quite significant.

One of the potential explanations of the results is that in the uplink case the RNC is not power limited and it is permanently providing power control messages to the UE. Hence, we can expect that quality of reception at the RNC is better than quality of reception at the UE in downlink case. Hence, in the uplink case using a larger RWIN leads to better performance, since probably the number of link layer retransmissions is low. On the other hand, in the downlink case if the link quality is worse more frames are lost, hence a less aggressive RWIN configuration is optimal.

In general, as a result of the study it was clear that underlying TCP mechanisms in charge of managing ACKnowledgements, retransmissions and window management do have impact into overall system utilization and performance. This type of analysis provided a great background to the future work to be carried out when evaluating application layer protocols performance over 3GPP networks.

3.2.4 Application level aspects. Web-browsing performance characterization

As an initial step, the work developed in [22] expanded the area of work into evaluating the performance of a specific Internet service (web browsing) over 2.5G and 3G.

Web browsing requires a number of interactions including several DNS lookups, TCP three-way handshake procedures, several request-response cycles to retrieve different web resources… When run over a 2.5G / 3G network with delay and performance constraints, web browsing performance under-utilizes the capacity of the wireless link.

In the following picture we present a comparison of different theoretical models of browsing techniques. In particular, we compared different HTTP versions with different capabilities in terms of pipelining or handling parallel requests. In the rightmost column we showcased the theoretical performance that can be achieved when implementing an ideal web browsing protocol that is capable of sending a unique request and all content can be delivered as part of one single stream.

Note that in all the below cases there is an initial interaction between the web client and the server (e.g.: TCP handshake, request of one or more URL's…) and, subsequently, a phase where raw delivery of the requested resources takes place. This type of interaction became of interest to us as it is quite similar to the signaling exchanges that are used in multimedia signaling protocols such as SIP or RTP. This work helped us to establish a theoretical background to further develop the analysis of signaling multimedia protocols performance over mobile networks.

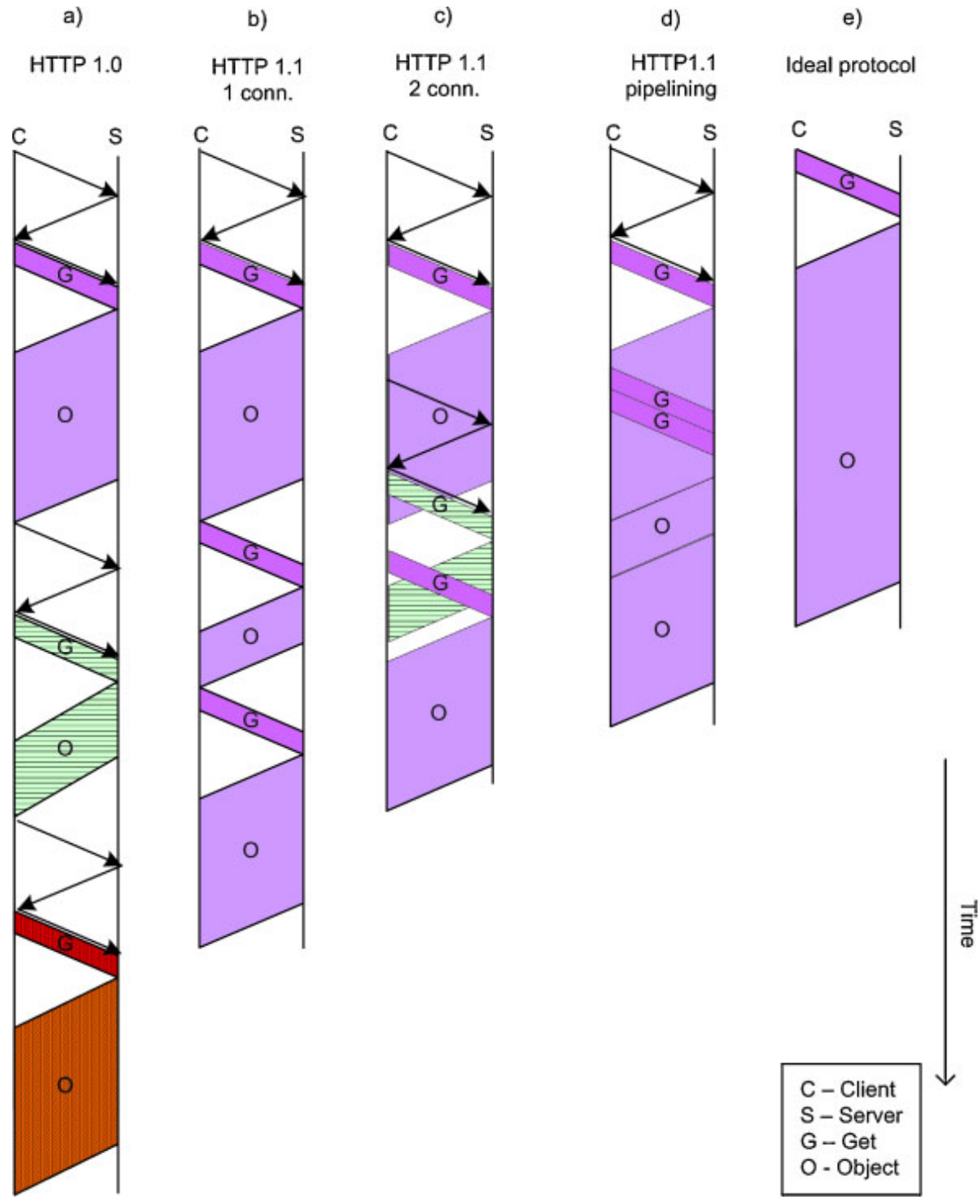

Figure 13. Comparison of end-to-end HTTP options plus an ideal web protocol.

The first case in Figure 11 shows the performance of HTTP1.0. The wireless link is significantly under-utilized due to the need to re-establish a full TCP three-way handshake for every requested HTTP resource. Options b and c offer better performance by upgrading to HTTP1.1 with connection reuse (b) and usage of parallel TCP sockets for HTTP requests, respectively.

Option c offers an interesting concept, HTTP1.1 pipelining, where the client has the option of issuing several HTTP GET requests in parallel, thus avoiding the impact of having to wait for previous requests to complete (whether TCP handshakes or previous HTTP GET transactions). Finally, column e outlines the performance of an ideal web protocol that could theoretically retrieve all information in one single request. Such last option is of interest for the sake of [22], where web performance enhancement proxies are discussed. However, such option is not feasible if the web client does not have prior knowledge about all the resources it intends to retrieve (effectively, a performance enhancing proxy may perform a number of operations "on behalf" of a web browser, while a web browser needs first to retrieve a resource and resolve the different links contained in it, which may trigger additional requests as the contents of the resource become known to the web browser).

For the sake of our discussions in the following chapters it is worth highlighting the similarities between retrieving a complex web page (which requires several requests, which may be handled sequentially, through parallel TCP connections, pipelined, …) and the signaling transactions based on protocols such as SIP or RTSP when handling multimedia session setup and control. This will be an important remark from chapters 4 and 5.

In summary, all the work described in the above paragraphs and sections provided us with a solid understanding of the impact of the radio link and its implications in IP data delivery as well as TCP protocol interactions. This work crystalized in the form of several co-authored contributions as papers [20] [21] [22] [23].

3.3 Additional considerations on performance of TCP/IP over other wireless links

In addition to the evaluation of TCP/IP performance over 2.5G and 3G as described above, it is of interest expanding the scope of work into other wireless links, such as WiFi, 4G/LTE or 5G environments. The author has made some small contribution to the work by García Villegas et. al in [23], when evaluating the performance of WiFi in the presence of multi-rate transmissions by different stations.

In a 4G scenario, Huang et al [24] develop a work with some similarities to [20] [21], but applied to the evaluation of a pre-commercial 4G infrastructure. Apart from the strong relevance of power measurements described there, Huang performs a set of empirical measurements to characterize 4G UL OWD / DL OWD, with resulting RTT in the range of 70 to 86ms [24] for packet sizes between 200 bytes and 1,400bytes.

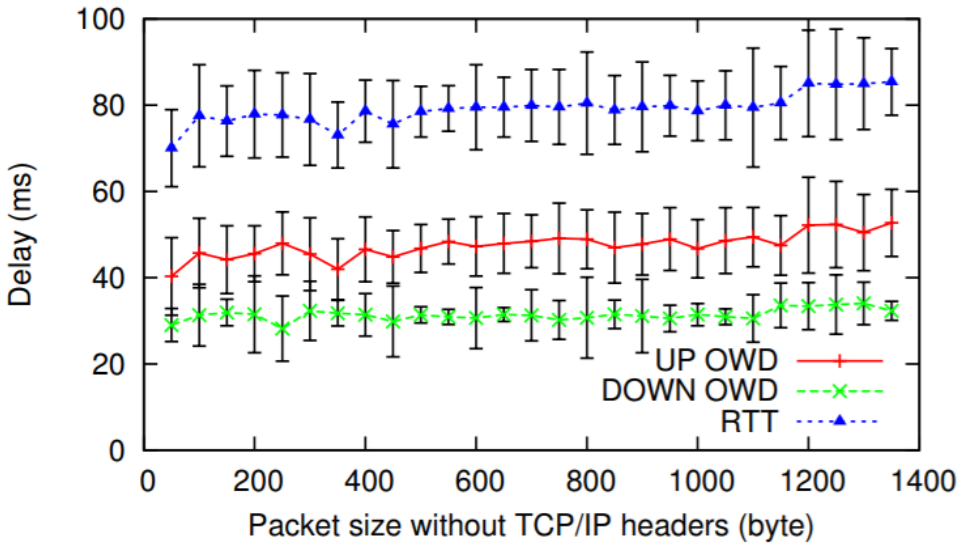

Figure 14. 4G UL OWD / DL OWD and resulting RTT for different packet sizes over 4G LTE [24].

Another relevant result includes the throughput and RTT comparison among different wireless technologies (e.g.: WiFi, WiMax, HSDPA, EVDO, 4G LTE).

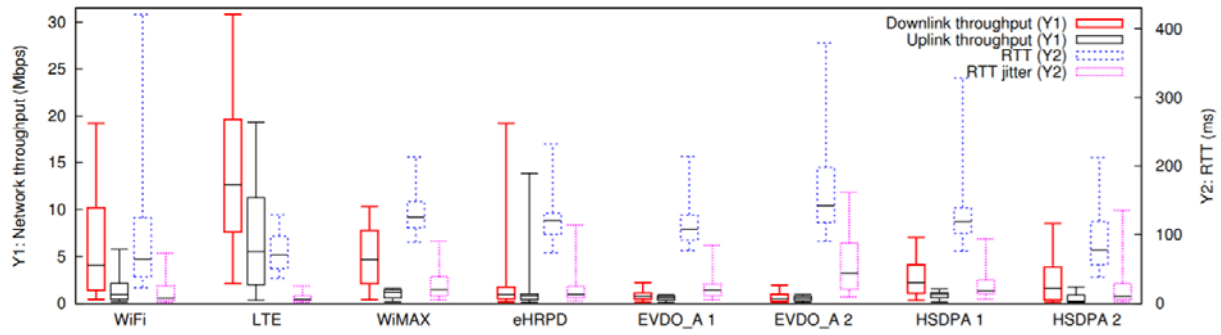

Figure 15. Throughput / RTT comparison among several network types based on 4GTest result analysis [24].

As depicted above, 4G LTE brings throughput results in the range of 5.64Mbps – 12.74Mbps, with RTT in the range 70ms-86ms as stated before. Measured RTT's for the family of 3G technologies lies in the range of 78ms – 200ms depending on the 3G / 3.5G technology in use. One important remark is that, as network RTT is reduced as new wireless technologies are deployed, other aspects in the end-to-end delay budget may become more relevant, such as internal device processing time [24].

3.4 Conclusions

Some of the conclusions outlined so far in this section will become relevant in subsequent chapters, as the contributions of the thesis are described in greater detail.

In general, most Internet-based services, and Internet Multimedia Services in particular, require some degree of client-server interaction where several transactions need to be completed prior to completing an intended interaction (e.g.: completing a multimedia session setup, starting an online video transmission, …). As stated, there are similarities between this type of interactions and the different web transactions required to retrieve a complex web content. Based on the work described in this chapter we can derive a preliminary outline of good practices for such interactive client-server interactions, when run over wireless networks:

1. If you need to send a packet, you'd better fit it with as much information as possible. Effectively, we have seen that while there is a small increase in RTT when sending large packets over 3G or 4G, it is much more effective to send one single large packet close to MTU value, than sending the same amount of information into several smaller packets.
2. Segmentation, encapsulation and retransmission strategies at link layer and transport layer may interact in different ways, which may require careful protocol parametrization to maximize performance.
3. Interactive client-server protocols are based on a request-response paradigm. The more operations you can run in parallel by queuing / pipelining requests, the better performance will be obtained in terms of optimizing interactivity of the given protocol.

With increasing throughput with each new mobile generation deployed (2.5G, 3G, 4G, 5G), it is RTT minimization that has a greater impact on service interactivity. As long as RTT becomes a relevant contribution into the end-to-end delay budget (especially when compared with transmission delay), reducing the number of transactions to achieve a given result may have a much greater impact into improved interactivity than other protocol optimizations such as signaling or payload compression. Actually, as outlined by [24], with ever improving throughput and round-trip delay, end user device processing power is becoming an increasing bottleneck for protocol and service designers, hence protocol simplicity is also an indirect but important factor in ensuring best possible experience of present and future Internet multimedia services.

Our contribution in this area, together with other co-authors, was reflected in publications [20] [21] [22] [23], which provided us with a solid background to tackle future work described in the following chapters, in which we moved our analysis and research toward the upper layers of the OSI and TCP/IP stack.

It is with these ideas in mind that our thesis work has focused on optimizing multimedia signaling protocols used for setting up streaming and/or multimedia communication sessions based on interactive protocols such as RTSP or SIP. In the following chapters we will describe how such design criteria have represented

an important factor in the thesis work that will be described in greater detail in the next chapters, focusing on enhancements of the 3GPP Packet-switched Streaming Service (PSS) (chapter 4) and the Push-to-Talk (PTT) service and other aspects of SIP-based services (chapter 5).

Before getting into that, it is worth noting that with the irruption of 5G and ultra-reliable low latency communications (URLLC) the industry is not only focusing on delivering "4G plus one". Actually a complete new set of use cases, scenarios and possibilities are opened up by enabling an ultra-low latency mode with round-trip times below 5ms. In this context, the delay contribution of adding a few transactions to a given scenario may not be as dramatic as with previous 3GPP generations. This will open up a new wide range of possibilities of instant interaction among virtually any type of device, that will be the baseline of the future of the Internet of Things (IoT), connected vehicle, vehicle-to-anything communications. The scope of our thesis work will mostly focus on 3G/4G environments, where RTT still plays a relevant contribution to the end-to-end delay budget.

4 Optimizing signaling protocols in the 3GPP Packet-switched Streaming service

4.1 Introduction

After the completion of our work related to the evaluation of underlying TCP/IP protocols over 3GPP networks, we focused our activity in the core of the thesis contribution, namely the evaluation and optimization of IP multimedia signaling protocols and architectures over 3GPP networks. These are protocols that run on top of the underlying TCP/IP stack, hence the tools and methodology developed in the previous chapter will prove useful moving forward.

Among the two main signaling protocols which are object of our study (SIP and RTSP) we will devote chapter 4 to the RTSP protocol and the 3GPP Packet-switched Streaming Service (PSS). The chapter will be split into the following parts: in sections 4.2, 4.3 and 4.4 we will provide an overview of 3GPP PSS standardization, focusing on initial standardization work and more recent standardization work respectively. In section 4.5 we will provide an overview of our contribution aimed at using session information level to enhance 3GPP network admission control procedures. In section 4.6 we describe our two other contributions aimed at optimizing streaming session setup based on the RTSP protocol when used over 3GPP networks with Quality-of-Service support. Section 4.7 contains the chapter conclusions and additional context and relates of our proposed contributions with more recent research and standardization at IETF and 3GPP.

4.2 Origins and Overview of the Packet-switched Streaming Service

At the time of its creation, the initial focus of 3GPP was in the development of new radio and core network elements to support the delivery of a new mobile communications system providing greater spectral efficiency and higher data rates. Such effort delivered results in the form of the WCDMA-based Radio Access Network (RAN) for the UMTS system, together with the harmonization of the evolution path of other 2G / 2.5G technologies (e.g.: GPRS, EDGE, cdma2000, EV-DO, …) towards the 3G umbrella. These deliverables in turn evolved towards so-called 3.5G and 4G radio interfaces recently defined by 3GPP in releases 6, 7, 8 and beyond (e.g.: HSPA, HSUPA, LTE, LTE-Advanced, …).

One of the important elements of 3GPP Release-4 – Release-6 work was the inclusion of a new packet-based service enabler that would leverage the newly available high speed packet-based radio interface (i.e.: WCDMA). Such service was the *Packet-switched Streaming Service* (PSS).

Strictly speaking, at the time 3GPP initiated PSS standardization, the first services that would leverage the packet domain in cellular services had already been defined: WAP and MMS had been standardized by the WAP Forum and 3GPP Release-99 respectively. However, PSS can be considered the first IP multimedia service defined by 3GPP with the ultimate goal of delivering an innovative set of attractive multimedia services to end users that would leverage new radio and core technologies standardized for 3G and beyond.

At this point, it is worth highlighting the reasons why PSS standardization represents an important milestone:

- As opposed to the definition of previous services such as WAP or MMS, the Packet-switched Streaming Service was developed with full reusability of Internet technologies in mind. Hence, PSS was the first one to reuse the IETF multimedia protocols umbrella (with some adaptations), rather than defining a new protocol suite.

- PSS could be thought as a particularization of the more general Internet based multimedia streaming service. Hence, 3GPP put special attention to ease the interoperability between the 3GPP PSS service and other streaming systems available in other IP domains.

- In contrast to previously available services such as WAP or MMS (or even SMS), PSS was the first service conceived to exploit greater capacity of 3G networks, eventually incorporating the benefits of mobile broadband with 4G (e.g.: PSS over 3G would deliver real-time streaming of video encoded at “hundreds” of kbps –up to 2Mbps, while PSS over 4G would enable HD 4K video streaming at several Mbps).

In this framework, it is interesting to observe some of the goals of 3GPP when starting the definition work of PSS [25]. The service was conceived as filling the gap between non-interactive applications such as MMS or content downloading, and conversational services. The core PSS spec [26] would include the possibility to stream both live and on-demand content, remarked the possibility of triggering streaming sessions from the reception of multimedia messages over MMS and highlighted the fact that –after the definition of a standardized service for multimedia delivery– operators and content providers would not need to purchase diverse proprietary platforms. Further, the fact that media streaming requires less processing power than full-duplex multimedia (due to the fact that only the DE-coding operation is required, without any need to EN-code media in the case of PSS) was seen as an interesting feature, enabling users to enjoy multimedia content without the need to increase device complexity.

The first releases of the 3GPP PSS standard would be included in Release-4, and would consist of two main documents:

- A basic "stage 1" description of the intended service contained in 3GPP TS 26.233 [26]. "Stage 1" documents generally include an introduction of the architecture, as well as a general description of high level technical requirements and generic technical guidance. In particular, [26] presents a general framework, defining the usage scenarios, overall high level end-to-end service concept, and lists terminal related functional components. It also lists any identified service interworking requirements and indicates that IETF-defined *Real Time Streaming Protocol* (RTSP) and *Real-time Transport Protocol* (RTP) would be used to support the 3G streaming service. While in Release-4, a service overview and stage-1 document was consolidated into [26], in later releases a splitting between requirements definition [27] and general description [28].
- A more specific document (3GPP TS 26.234) defining a more detailed profile and guidance on usage of RTP, RTSP and CODEC formats [29] [30].

The basic architecture proposed by 3GPP for the initial PSS service in Release-4 is described in the figure below.

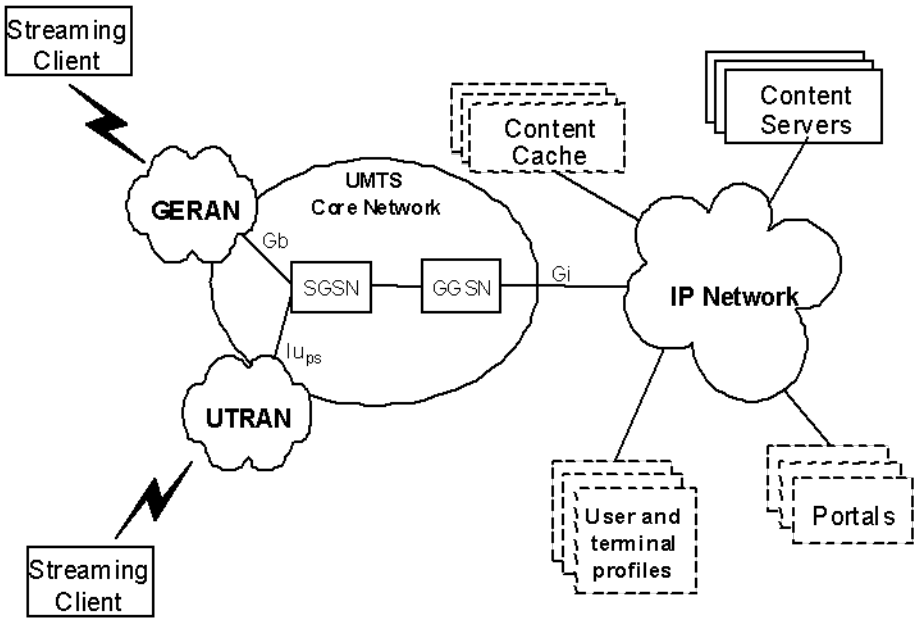

Figure 16. 3GPP PSS high level architecture in Release-4 (source: [26]).

The following picture provides a good overview of how RTSP works in a 3G context. In particular, MMS-triggered streaming is considered [26]. In order to support MMS-triggered PSS, the MMS service specified the need that the MMS application should be able to properly interpret a multimedia message carrying SDP content, and trigger the media player application with the corresponding content URL contained in the SDP description file [26] [31].

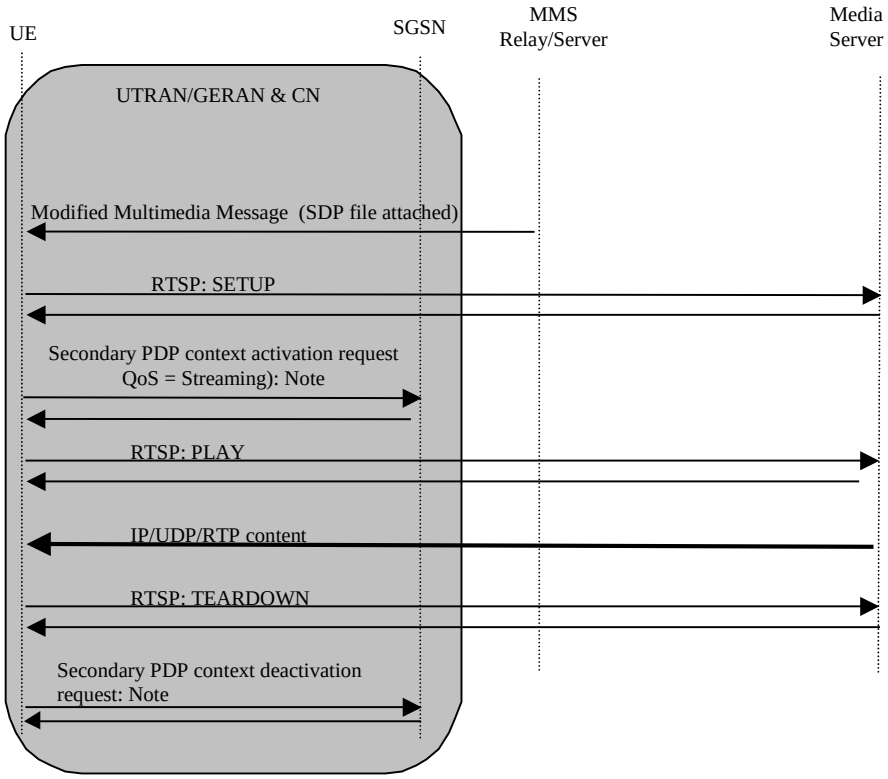

Figure 17. Multimedia streaming triggered by reception of an MMS with SDP content [26].

In the above picture we can see several important aspects of the PSS service that will be leveraged later on:

1. In this case, session setup is triggered by the reception of a multimedia message. This message generally contains an SDP description, which lists all available media in the session, source IP addresses and UDP ports, media URLs to trigger the stream, as well as CODEC and additional information required to start the stream. In a typical RTSP environment, this process replaces the usual RTSP DESCRIBE transaction, which is used to receive session description information.
2. Once media description has been received, the client may issue one or more RTSP SETUP commands towards the PSS Server. One SETUP command is issued per every media that the user intends to receive (e.g.: audio, video, subtitles, …).
3. In parallel, in a 3GPP network with QoS support, the Client / UE may request the setup of a Secondary PDP-Context bearer. This is a mechanism specific to 3G / 4G networks, that lets the UE request the setup of a bearer service with specific Quality-of-Service (QoS) requirements. As an example, in a PSS context, media delivery typically requires a guaranteed bitrate bearer with packet loss ratio in the range of 10^{-6}, because the transport or application layer generally do not implement retransmission of lost frames. On the other hand, for non-real-time video, delay constraints are generally less strict, with delays in the range of 300ms being acceptable in general.

4. One the session has been "prepared" through the SETUP message and a secondary PDP-Context has been allocated, the client will issue a PLAY message for one or more streams, and downlink media delivery of audio and/or video over RTP will happen.
5. Once media reception has completed, the client may tear down the session and the secondary PDP context in order to free network resources.

In the research domain, at the timeframe of conception of 3GPP PSS, two key papers were issued by two of the key sponsors of the PSS work item at 3GPP: Nokia and Ericsson. In September 2001, IEEE Computer published "Streaming Technology in 3G Mobile Communication Systems" [32], where the PSS service was presented in scope of the evolution of radio access technologies and the industry trend of developing horizontal applications decoupled from the transport layer. Further, the authors highlighted the importance of determining device capabilities and allowing negotiation of session characteristics during session setup. Both suggestions would form an important improvement of the new features of the PSS service to be standardized in Release-6.

Later on, "Deployment of IP Multimedia Streaming Services in Third-Generation Mobile Networks" was published in IEEE Wireless Communications [33]. In this contribution, several authors from Nokia highlighted the importance of leveraging the packet-switched domain for media streaming services, as the most efficient way to utilize radio resources. Another important contribution of this paper consisted in highlighting the possibility of connecting the application layer domain (i.e.: the PSS server) with the network layer domain to improve performance of the service over 3G networks.

The idea behind this proposal was as follows: 3GPP had defined a policy node in the packet-switched domain. Such node was called the Policy Control Function (PCF) and implemented Policy and Service Level control over the media flows and PDP-Contexts requested by applications in the packet-switched domain. The PCF was a logical function residing in the P-CSCF, that behaves as the outbound SIP proxy for IMS clients in the cellular domain. The PCF residing in the P-CSCF would then connect towards the GGSN through a signaling interface (the Go interface). The role of the Go interface was to ensure consistency between the radio resources requested, reserved and used by applications for user plane traffic (e.g.: RTP media) and the bandwidth requirements signaled by application layer protocols such as SIP).

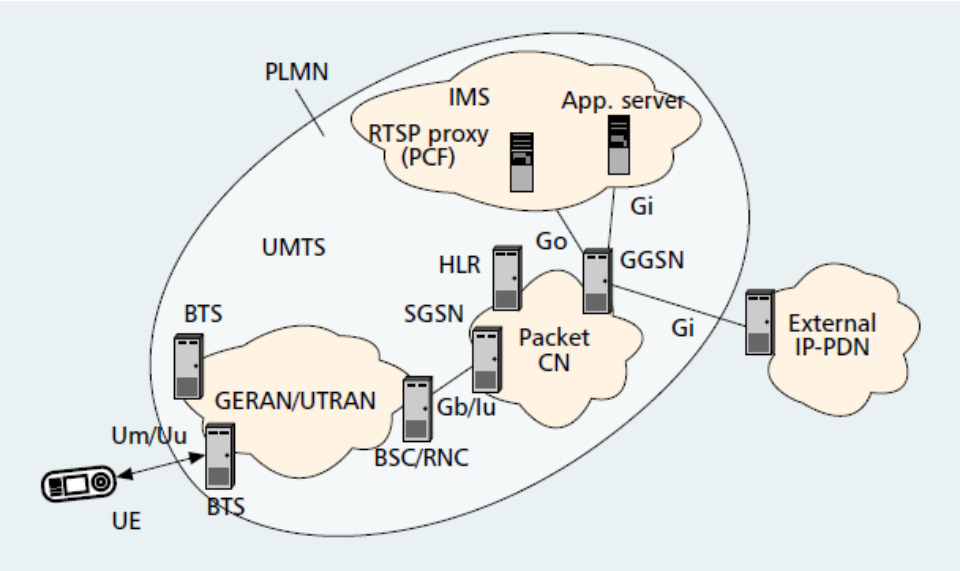

Figure 18. PSS streaming connected to the PCF through Go interface (source: [33]).

The benefits of connecting the RTSP/PSS proxy to the PCF would be two-fold: first, being able to provide policing functions to streaming services (for which it might be of particular interest, given their typical bandwidth requirements), and second –as a consequence of the first– the possibility to leveraging IMS infrastructure (the PCF) without having to migrate RTSP functionality towards the SIP protocol (this has been a recurring topic in standardization and research within the Telecom Industry in the last years, as we will see).

The architecture depicted above would not end up evolving further, particularly due to the lack of interest and standardization work on the Go/Gq interface and the lack of interest in the COPS protocol they used [34]. However, 3G PDF and Go/Gq interface would later evolve toward the 4G PCRF (Policy and Charging Rules Function) and the corresponding Gx and Rx interfaces (based on DIAMETER [35]), with substantial similarities with the above architecture.

4.3 Initial enhancements of the 3GPP PSS service

After the baseline definition of the service, in Release-5 an important upgrade of the PSS was made. A fundamental feature was the definition of a new file format to store media content: the 3GP file format. In Release-4 a profiling of the MP4 format was specified, in order to support storage of video content (H.263 and MPEG4) and AMR-NB audio content (this later case was not supported by the native mp4 format, so it had to be defined in [31]). With the definition of the new and extensible 3GP format, 3GPP gained greater flexibility to incorporate new features to the streaming service and new CODECs and profiles in future releases, without dependencies on other SDOs defining the MP4 format, in which 3GP was based.

PSS Release-5 also included the addition of new media types, such as timed-text (i.e.: text whose actual display within the multimedia stream is time-synchronized, so that it appears in a precise intended moment and position, to support display of sub-titles).

When it comes to the actual delivery of media streaming content over the 3G network, two new concepts appear: Capability Exchange and advanced buffering mechanisms. The Capability Exchange concept was borrowed from the WAP service. It consists on an XML-formatted file called a *Resource Data Framework* (RDF) [36] that is stored in a content server and contains detailed information about the capabilities of a given user agent (e.g.: a client residing in a mobile device). When a client issues a streaming request, it also sends a "User-Agent" header and a "x-wap-profile" header that provide information about the type of client being used. The PSS server retrieves the PSS-specific "capabilities" of such client (since the mechanism is shared among services such as PSS and WAP, the file may contain information about different services. Such information is structured within the XML document). This mechanism lets the PSS server filter out and provide a pre-adapted version of the media offering when initiating a streaming session. The procedure is depicted in the figure below.

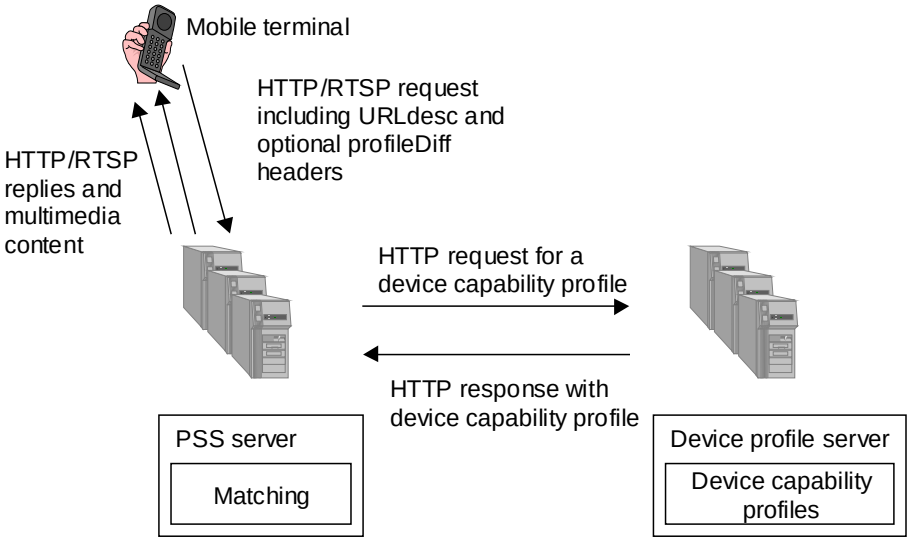

Figure 19. The PSS capability Exchange mechanism (source: [36]).

The other new enhancement introduced in Release-5 was the definition of a buffering algorithm. This buffering algorithm, combined with client capabilities exchanged prior to session setup, and periodic RTCP feedback delivered by the PSS client should help the PSS server perform a coarse adaptation of the actual rate at which RTP packets would be delivered towards the network. The buffering mechanism introduced in [36] would be the actual foundation of the more complex buffering and feedback mechanisms to be defined in Release-6.

After Release-5, the most relevant Release-6 enhancements included:

- Addition of new CODECs, profiles and formats. Specifically, with the progressive spread of WCDMA 3G networks, it was noticed that video profiles available in Release-4 and Release-5 would not take full advantage of the Radio Access Bearers (RAB) that would become available in the packet-switched domain. Hence, new (H.263 profile 3 and MPEG4 VSP level 0b) were added, allowing larger screens (e.g.: QCIF) and, specifically, larger bandwidth (e.g.: up to 128 kbps). Another interesting addition was the incorporation of next generation audio and video CODECs such as H.264 and AMR-WB+ and eAACplus, respectively.
- Addition of a "progressive download" profile based on HTTP / TCP, that would enable the rapid playback of received multimedia content (stored in a 3GP file) without the need of actually having completed the download. This new feature would not leverage the adaptation and QoS features of the PSS service based on RTP/UDP, but would allow a pseudo-streaming experience to be displayed to the user from an integrated web browser. The main impact of this feature is during media encoding and storage: actual streams ("tracks") do not need to be "hinted" (adapted) for real-time delivery, but optimized for a download experience, and the 3GP file needs to be stored in a way that –once its delivery starts– the recipient can start rendering it immediately (i.e.: deliver streams in parallel, rather than sequentially).
- Specification of the end-to-end architecture that would enable the provision of QoS-enabled streaming services and policy control. This work would ease the mapping of RTSP/SDP signaling information into the QoS requirements that PSS clients would need to demand to the 3G network when requesting the setup of Streaming QoS-enabled bearers. Furthermore, 3GPP started the definition of an architecture that would allow the connection of the application layer functions (e.g.: a PSS server), the Policy Decision Function (PDF) and the Policy Enforcement Function (PEF, e.g.: the GGSN) in a 3G network. This architecture would allow operators the implementation of proper admission control, resource policing and charging functions, with full visibility of all media streams (e.g.: RTSP, RTP, RTCP) associated to a multimedia streaming session, or to a multimedia service in general. This architectural work was first included in the 3GPP QoS activities in Release-6 ([37],). This new architecture defined by 3GPP, which would standardize the interfaces proposed in [38] was also described by the authors in [16].

  It is also relevant to observe that the QoS mechanisms and architecture introduced in these references posed a new challenge into the PSS end user experience. Effectively, when the service is run over 3G networks with enabled QoS, the session setup time may be affected by the need to establish a secondary data connection (a PDP-Context, in 3G terms) for actual media delivery using a Streaming QoS Traffic Class. This concept can be deducted from inspecting Figure 12

Addition of new advanced RTP and RTCP profiles, to enable detailed client reception feedback. This new extension would enable a PSS client to report detailed information about the actual quality of the reception of each media stream. New information would overcome the limitations of the more reduced RTCP feedback defined in basic RTP/RTCP [10]. The new RTP/RTCP extended reports and retransmission formats would enable PSS clients report detailed information about the status of their reception buffer (as to avoid buffer overflow or underflow) and detailed reception maps, containing a complete and accurate information about what packets have been lost during a transmission. With PSS Release-6, hence, the streaming service would enjoy the possibility of application layer retransmissions, which in the past had been considered a contradiction for real-time services. This new range of capabilities allowed for much smoother media delivery and playback, due to better adaptations to the varying conditions of the wireless channels. These capabilities, hence, justified the consideration of the Release-6 streaming service as a "transparent" or "adaptive" application, thus enhancing the behavior of previous PSS releases [39].

The main elements that would allow for a more adaptive PSS service were: detailed client feedback, RTP retransmissions and media stream switching based on reported network conditions. This latter feature meant that a same content (e.g.: a given video feed) could be encoded at different bitrates, thus effectively generating several media streams. The service could dynamically switch among those streams when perceived network conditions (e.g.: due to detailed RTCP feedback) vary significantly. The actual stream switching, hence, required the definition of additional extensions to the 3GP file format, which were consolidated (together with the added new CODECs) into [40].

- Another important aspect incorporated to the PSS service was the possibility for the client to report "application level" quality of experience reports. This feature meant that the application could deliver reports to the service, informing about the impact of potential network issues (e.g.: packet losses, reordering, …) into the actual quality of the rendering experienced by the end user. This way, the application can report information such as: "video stream has been frozen during 5 seconds, out of the last reported minute". This information can then be used by the PSS operator to implement policy and charging decisions (e.g.: to apply a discount or not to charge the user, unless a minimum quality has been experienced).

An important reference presenting the advances of 3GPP Release-6 PSS can be found in [39]. Further, 3GPP has made intense effort in providing RTP usage guidelines to implementers, in areas such as packet sizes, encoding and transmission strategies… This work has been progressively compiled into each release version of [41].

1. After completion of Release-6 standardization, PSS offered a feature rich and adaptive service that would offer basic interoperability to simple PSS clients, or advanced mechanisms to implementations compliant to the latest standard release. The most relevant updates to the PSS service in standardization and research activities has been in three main directions, namely: a) addition of new video and audio CODECs, b) enabling fast session startup and channel switching times, and c) merging PSS and MBMS services. Some of the benefits incorporated by 3GPP into Release-7 specs are similar to the optimizations proposed by the author in [18].

The goal of incorporating new optional audio and video CODECs as to keep the PSS specification up-to-date with new device capabilities, such as new available CODECs, processing power, enhanced radio interfaces (e.g.: HSPA) or improved screen sizes, resolutions or color depth.

Secondly, as the importance of accessing digital media from any device, using alternate technologies has become more and more relevant, the PSS service has evolved as a Mobile-TV platform, thus providing mobile access to any type of digitized multimedia content, including availability of live TV channels over the cellular domain. With this business perspective in mind, the importance of fast session setup and channel switching has increased: users demand the same level of interactivity to streaming services, as they do to their TV appliances at home, and expect a response time that provides an appealing experience to them. Thus, ensuring that setup and switching times are as short as possible is key to keep user attention towards the service.

Additionally, the emergence of new radio bearers such as MBMS, offering multicast support to IP-based applications (hence, a better spectral efficiency for delivery of media content to users willing to receive a specific –e.g.: live– stream). Release-8 and Release-9 standardization is hence putting effort in enabling RTSP-controlled media delivery over MBMS.

4.4 Recent 3GPP PSS standardization work

3GPP PSS based on RTSP and RTP was essentially completed around Releas-8 and Release-9 timeframe. In subsequent releases (Release-10, Release-11, Release-12 and recently completed Release-13) focused in some slightly different areas.

First and foremost, the explosive growth of media streaming over the Internet, initially based on proprietary technologies, and later on adopting HTML5 over HTTP, eventually led to the broad adoption of HTTP-based streaming capabilities to mobile devices. Recognizing such broad industry trend, 3GPP extended the PSS service with HTTP download capabilities. After initial work around a “Progressive Download” PSS profile, a full HTTP-based streaming architecture was defined in 3GPP DASH (Dynamic Adaptive Streaming based on HTTP) and specified in [42].

A high level architecture of 3GPP DASH is depicted below.

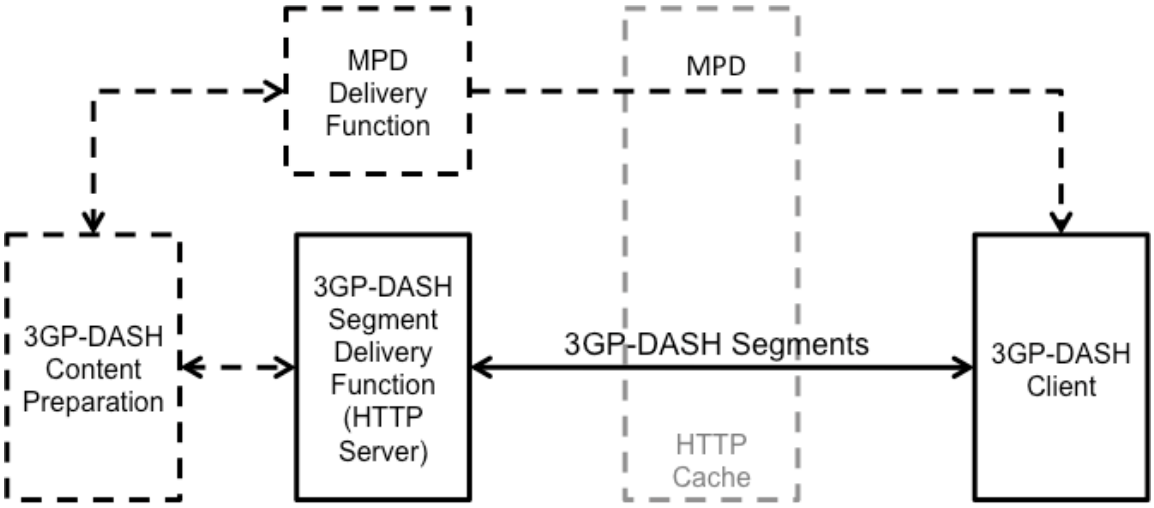

Figure 20. 3GPP PSS architecture based on Progressive Download and DASH [42].

In subsequent work during Release-10, Release-11, Release-12, Release-13 3GPP DASH new features have been incorporated, such as new QoE reporting mechanisms, incorporation of H.265 HEVC CODEC profile and new content protection and DRM mechanisms [43].

After the thorough review of the PSS provided in the previous section and the baseline TCP/IP performance evaluations of the previous chapters, in the rest of chapter 4 we will put everything together and describe the author's contributions to the area of Packet-switched Streaming over 3GPP networks, with focus on two main areas, namely:

- Session-information based admission control adaptation of PSS over 3G 4.5.
- Optimization of RTSP session setup time through SDP templates and enhanced pipelining 4.6.

4.5 Strategies to enhance Packet-switched Streaming (PSS) Admission Control (AC) procedures

4.5.1 Introduction

After the overview of the 3GPP PSS service our initial investigation work aimed at enhancing the experience of the service, focused on the aspect of Admission Control (AC) for PSS over 3GPP networks considering a generic 3GPP PSS architecture with QoS control over the Go interface as depicted below.

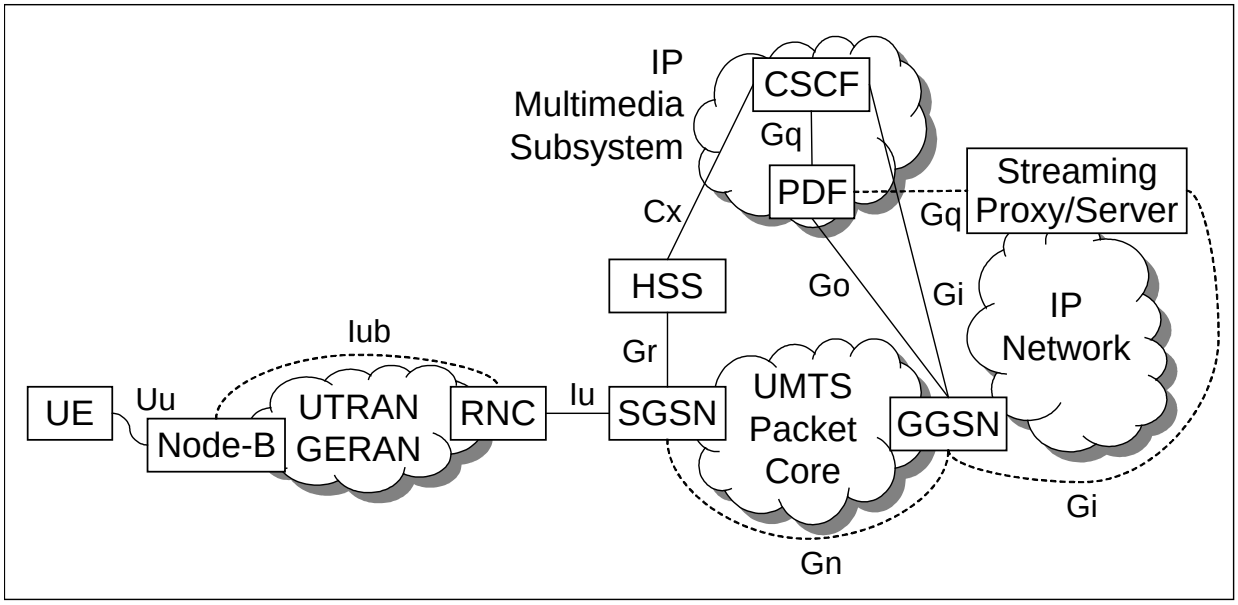

Figure 21. Reference 3GPP PSS architecture with QoS control through PDF.

AC mechanisms are not generally specified by standardization bodies as it is left as a field of differentiation between different vendor implementations. However, AC strategies in general and their application to cellular 3G systems is a field of intensive research [44] [45].

In the particular case of 3G networks, AC is a rather complex procedure that takes place in a multistage and distributed fashion, since it relates to radio access network admission control (RAN AC), core network admission control (Core AC) and PSS service admission control (Service AC).

In our discussion in [16] we argue how information about expected time duration of a streaming session can be used to further optimize the admission control process. Obviously, this approach is dependent on knowing expected session duration, which mostly applies to pre-recorded content or time-boxed live streaming. Hence, the assumption is made that such type of content (where expected duration can be estimated) plays a relevant role in the overall traffic mix, so that an enhanced AC strategy based on the knowledge of this traffic profile brings value to the overall AC system.

4.5.2 RAN AC overview

RAN AC mechanisms try to ensure proper performance of all ongoing and future sessions if the new requested service is accepted. One approach is to estimate the minimum E_b/N_0 level that is required for each session in order to meet that particular session's QoS requirements. A given Mobile Station (MS) *i* located in a cell *k* should accomplish the following constraint [44]:

$$\left.\frac{E_b}{N_0}\right)_{i,k} = \frac{W}{r_{i,k}} \frac{P_{i,k} h_{i,k}}{I_{\text{int},k} + I_{ext,k} + \eta_0 W} \geq \gamma_i$$

Where E_b is the bit energy, N_0 is the total interference density received at the Base Station (BS), η_0 the thermal noise spectral density, I_{int} and I_{ext} are intracell and intercell interferences, γ_i is the minimum E_b/N_0 value for which the required QoS can be guaranteed, P_{ik} is the transmitted power associated to MS i in cell k, $L_{i,k}$ is the path loss between the MS and the BS and the ratio $W/r_{i,k}$ is the spreading factor associated to the WCDMA transmission.

The basic idea of RAN AC mechanisms is to decide whether admitting a new session i at cell k, having a certain $E_b/N_{0i,k}$ constraint, will let the network keep accomplishing all $E_b/N_{0j,k}$ ($j \neq i$) constraints for already ongoing sessions.

Intensive research considering parameters such as single cell and multicell scenarios [4], downlink or uplink based mechanisms has been performed to enhance RAN AC mechanisms.

4.5.3 Core and Service AC

Core network AC in 3G is based on traffic classification and resource availability. Even though manufacturers and operators may define different types of policies for AC, we assume that a certain amount of available bandwidth is reserved for Real-Tine services; a fraction of Real-Time traffic may be reserved for Conversational services (which may receive higher priority than Streaming applications). Statistical multiplexing allows Background and Interactive traffic to use the free capacity.

Once reserved bandwidth thresholds have been setup, we assume that AC is performed based on system load monitoring and evaluation of requested resources for incoming sessions: if the total aggregate of system load and requested resources falls below the threshold of the right category (e.g.: Real-Time services) the session is accepted; if admitting the session requires exceeding the threshold, the session is dropped.

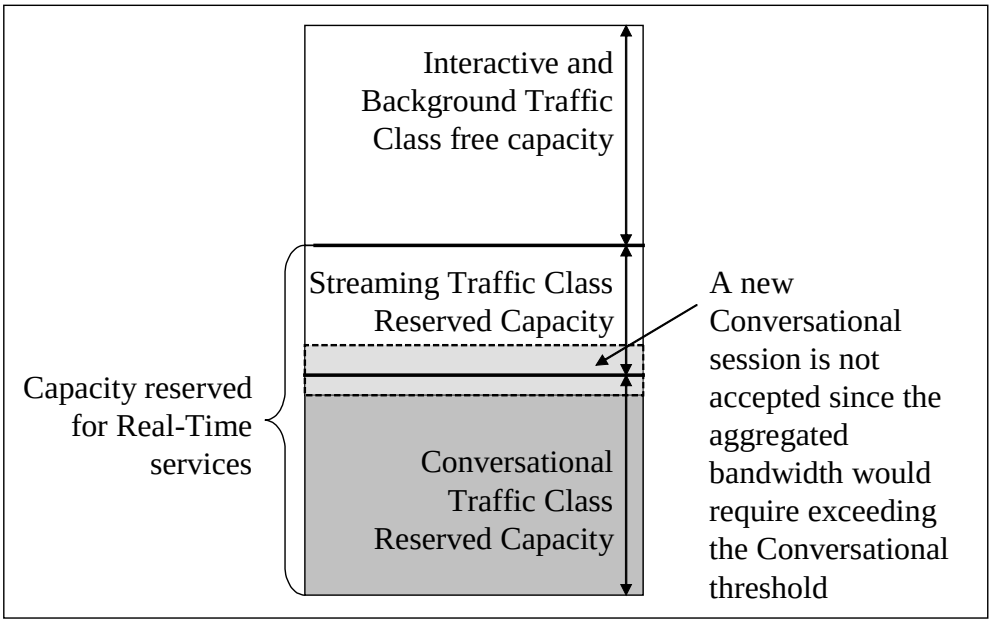

Figure 22. Example core network Admission Control.

In addition to radio and core network AC mechanisms, service-based authorization may be implemented as well. As an example, it must be decided if a given session can be admitted based on network policies, user profile and Home Subscriber Server (HSS) settings.

4.5.4 Mapping of session parameters to QoS parameters

During session setup stage, the MS has to indicate the required QoS profile parameters for a given media flow. These parameters should be included when activating a Streaming Packet-Data-Protocol context (PDP-Context), which is the logical connection established between the MS and the 3G network when a new session starts. The mapping between application-level requirements to parameters that let the network perform AC mechanisms may be implemented as follows:

1. A user requests a streaming session using the RTSP DESCRIBE method. Session description is delivered as an SDP message specifying, among other parameters, session duration and required bandwidth for each stream. RTSP messages may be exchanged using a PDP-Context that has been previously established.
2. The streaming application requests a new Streaming PDP-Context and indicates the required QoS profile:
   - Guaranteed and maximum bitrate are obtained through the SDP `b=` line indicating application requested bandwidth.
   - The type of audio and video CODEC used (included in the SDP message) can serve as the basis to request certain residual Bit Error Rate (BER) and BLock Error Rate (BLER) settings.

- Delay requirements may be calculated based on buffer settings.
- Additional parameters may be deducted and/or default values might be used for certain traffic classes

Once required information has been gathered, the request of a new PDP-Context takes place, which triggers different AC procedures to be started through the 3G network in order to determine if the new session can be authorized.

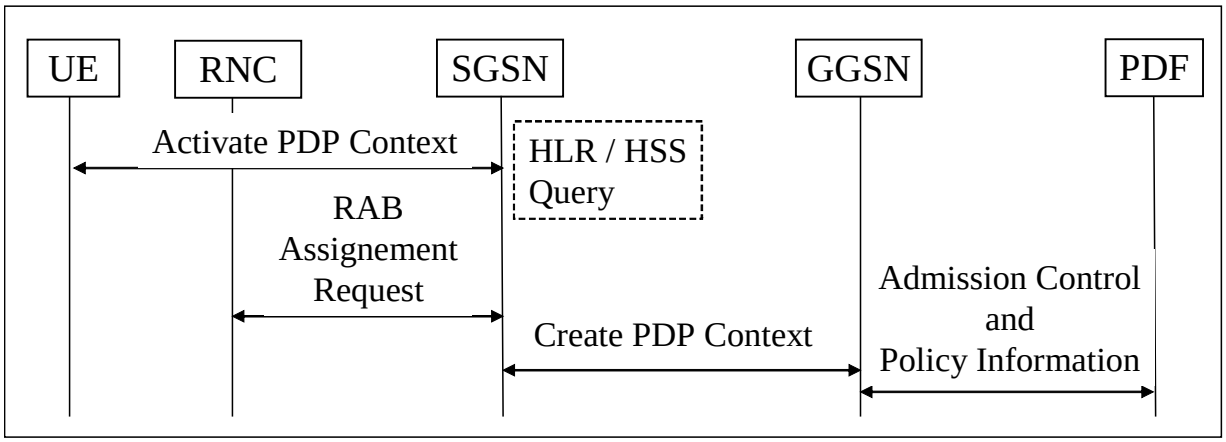

Figure 23. Example PDP-Context setup flow over 3G involving PDF interaction.

PDP-Context settings are sent over an `Activate_PDP_Context_Request` message to the SGSN, which performs core network AC (e.g.: check for available resources as explained in section II) and queries the Home Location Register (HLR) database to validate the session.

At a second stage, SGSN sends a `RAB_Assignment_Request` message to the 3G Radio Network Controller (RNC) asking for the creation of a new Radio Access Bearer (RAB) Service. RNC performs RAN AC procedures. Input parameters used for these operations are:

1. Spreading Factor requirements for the new communication. This parameter is calculated out of the Guaranteed and Maximum Bitrate parameters sent by the MS when asking for PDP-Context activation.
2. Power measurements performed at the BS.

Finally, a `Create_PDP_Context_Request` message is sent to the GGSN, it performs additional core network AC procedures, and finally the PDF is queried for service-based AC and policing. If all AC procedures are successful, the session is authorized.

4.5.5 Example scenario

In order to validate the enhancements of the admission control process proposed by [16], an example scenario with the streaming of a real audio & video streaming session over 3G was used.

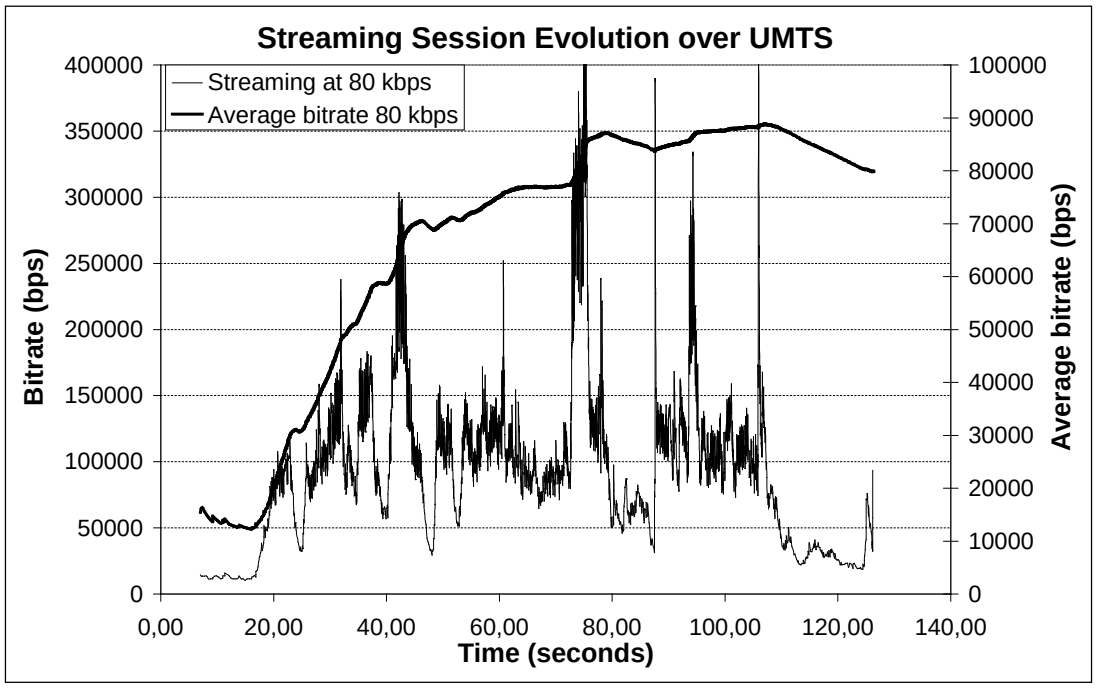

Figure 24. Example scenario. Streaming session evolution over a 3G network.

An observation of the above figure gives an idea of the high variability of streaming traffic. This behavior is directly related to motion characteristics of the media content. Some encoding and delivery mechanisms as those described in [41] may help to reduce instantaneous bitrate variability to a certain extent; however, it is likely that a degree of rate variation will prevail regardless of the delivery technique in use in many cases (assuming that a certain degree of motion images exists in the original content).

An observation of the previous figure from the AC point of view may lead us the following conclusions:

- If a new streaming session request arrives to the 3G network during the first 30 seconds, AC mechanisms based on averaged load measurements may make an erroneous decision by authorizing network access to the new session.

  Effectively, during the first 30 seconds, average bitrate of the already ongoing session falls below the 40 kbps range, thus being 40 kbps lower than the SDP signaled bandwidth. As a consequence, the network may erroneously determine that additional 40 kbps are available for new sessions, which is not the case (because average bandwidth of the started session will progressively rise to reach approximately 80 kbps).

- On the other hand, a session arriving approximately 108 seconds after the displayed connection, might be refused or downgraded during PDP-Context negotiation due to the fact that averaged bitrate of the previous session is around 80 kbps. However, the session is finalizing, and the residual content that is yet to be streamed will be far below 50 kbps on average.

4.5.6 Usage of session information to enhance AC Mechanisms for Streaming Services

Based on the example session, the core of [16] contribution proposes to leverage additional SDP information which may only be used at the application layer to further enhance Admission Control at different layers (e.g.: RAN, Core).

Essentially, by using timing information exchanged during session setup it is possible to make an estimation of expected "playtime" of all ongoing sessions. Effectively, it is possible to reuse the `t=` field in SDP to capture the total duration of a session during the original DESCRIBE / 200 OK. This of course applies specifically to pre-recorded sessions, rather than to live sessions that may last for an undefined time.

As we have seen above, not considering session duration and expected workload into the system may lead into sub-optimal admission control decisions, such as optimistic acceptance (e.g.: at the beginning of the stream depicted in the previous session) or pessimistic acceptance (e.g.: by not accepting new streams while some sessions are reaching end).

In this context, if we reuse information from t= SDP header, it is possible to configure the PSS AC mechanism to reuse t= SDP header in the following manner [16]:

1. Enhancement mechanisms start when a streaming session is authorized to utilize network resources. At this point, session starting time and approximate expected ending time can be calculated (based on current time and session duration indicated in the SDP message) and stored in the AC system.
2. Sessions that are likely to finish in a short period of time (e.g.: the next 5-10 seconds) are located, and for each about-to-end session, a percentage of signaled average bandwidth is deducted from current system load.
3. Most recently started sessions are located up to a certain threshold (e.g.: 30 seconds before current time, thus considering this as a session transient period). For these sessions, average bandwidth is calculated and compared with their SDP signaled average bandwidth.
4. If signaled average bandwidth for recently started sessions is higher than current session bandwidth, add the difference to current system load.

5. AC for the new incoming session can then be performed using the new estimated system load (i.e.: considering sessions that are about to finish, and having considered that some sessions' average bitrate might rise to reach a stable bitrate).
6. Sessions can be deleted from the system as they finish.

It can be seen that this AC schema better estimates future evolution of already ongoing sessions. It is still a conservative approach, because step 4 of the procedure only takes into account sessions that are below the signaled average bandwidth (sessions currently consuming more bandwidth than their average are not considered, thus leading to a "worse case" decision, which seems to be adequate). The procedure does not affect live streaming content (for which an ending time is not known and, therefore, is not considered for the algorithm) and conversational content, which transparently add on top of current system load.

In addition to the baseline mechanism, several enhancements are also proposed:

- Steps performed when a new incoming session arrives to the system should be also followed in case of other events, such as:
    - Users pausing an ongoing session.
    - Users fast forwarding, rewinding or repositioning the played content range.
- The procedure may be implemented by an element being able to monitor both RTSP/SDP signaling and current system load. As a consequence, it seems that the PDF element is the most suitable one to perform this enhanced mechanism, because neither the GGSN nor the RAN are able to interpret SDP signaling. It is a topic for further study to evaluate how this method might be ported to other network elements involved in AC procedures.
- Scalability of the method should be evaluated. However, assuming that already deployed monitoring mechanisms can be used, only a reduced set of additional information is required. Additionally, calculations are performed not in a periodical way, but only during session admission stage. Since typical streaming session setup delay may be in the range of 5-10 seconds, this may be affordable both by the system and by the user.
- In case that stored content does not represent a significant amount of traffic and/or high delay/jitter variations prevent the system to accurately estimate session ending time, usefulness of the procedure should be revised.

Since timing information is contained in RTSP/SDP info, it is the PSS/RTSP Server the one receiving this information. In order to leverage this information at the RAN and Core AC level (which may be more constrained than the application server) a mechanism to push such timing information downwards toward the GGSN / EPC and RNC / eNodeB. This means that an interface between the application layer and the

network layer is required to implement this enhancement at all levels. In a 3GPP Release-5 architecture, Gq/Go would be used for such purpose (by extending the COPS protocol). In Release-14/-15 environment, DIAMETER-based 3GPP Rx interface would be used.

As mentioned through the rest of section 4.5, our contributions described in this section was published in International Journal PIMRC [16].

4.6 Enhanced PSS session setup based on SDP templates and RTSP pipelining

After the completion of the session information based admission control work, our area of activity focused on the evaluation and optimization of session setup time by further optimizing the core of the RTSP protocol when used over 3GPP networks.

At the time the work was carried out, best effort 3G networks had already been rolled out. However, little application of 3GPP defined QoS procedures were in place yet. We took the QoS model defined by 3GPP and studied in detail how PSS session setup would happen over 3G networks with QoS support, and how to optimize it.

In particular, one of our main goals was finding ways to further optimize PSS session setup procedures in order to save time. One of the concepts we developed significantly was “RTSP pipelining”, a way to speed up session negotiation and setup in PSS by being able to send several RTSP requests “pipelined” without having to wait for an explicit answer to each request prior to sending the next one (e.g.: hence, reducing the number of RTT’s required to set up a streaming session).

This work crystalized in the form of two significant contributions. In the European Wireless paper [46] we analyzed the generic requirements that should be met in order to optimize PSS session setup based on RTSP. In the Wiley WCMC paper [18] we provided a detailed implementation description and further justification for the proposed optimization mechanisms.

In the rest of this section we will provide an overview of the contributions developed in the field of RTSP-based PSS session setup over 3GPP networks with QoS control.

4.6.1 Origins. PSS session setup over 3GPP networks with QoS support

As described through the chapter, PSS reuses standard protocols such as the RTSP for session description, setup and control (e.g.: pause, tear down, …), SDP (for media and connection negotiation, e.g.: addresses and UDP ports), and the Real-time Transport Protocol (RTP) for media delivery. Actual media is typically encoded using high quality CODECs such as Enhanced AACplus for audio or H.264 for video [30].

PSS sessions can be triggered in different ways, such as reception of an SDP file through MMS, Mail or other messaging services, or usage of the RTSP DESCRIBE method after content discovery through (e.g.)

web browsing [36]. Our study will focus on the full case including the DESCRIBE transaction, as this is a likely scenario, and the one involving larger volume of signaling information exchanged over RTSP. The following picture provides an overview of PSS session setup over a 3GPP network with QoS support.

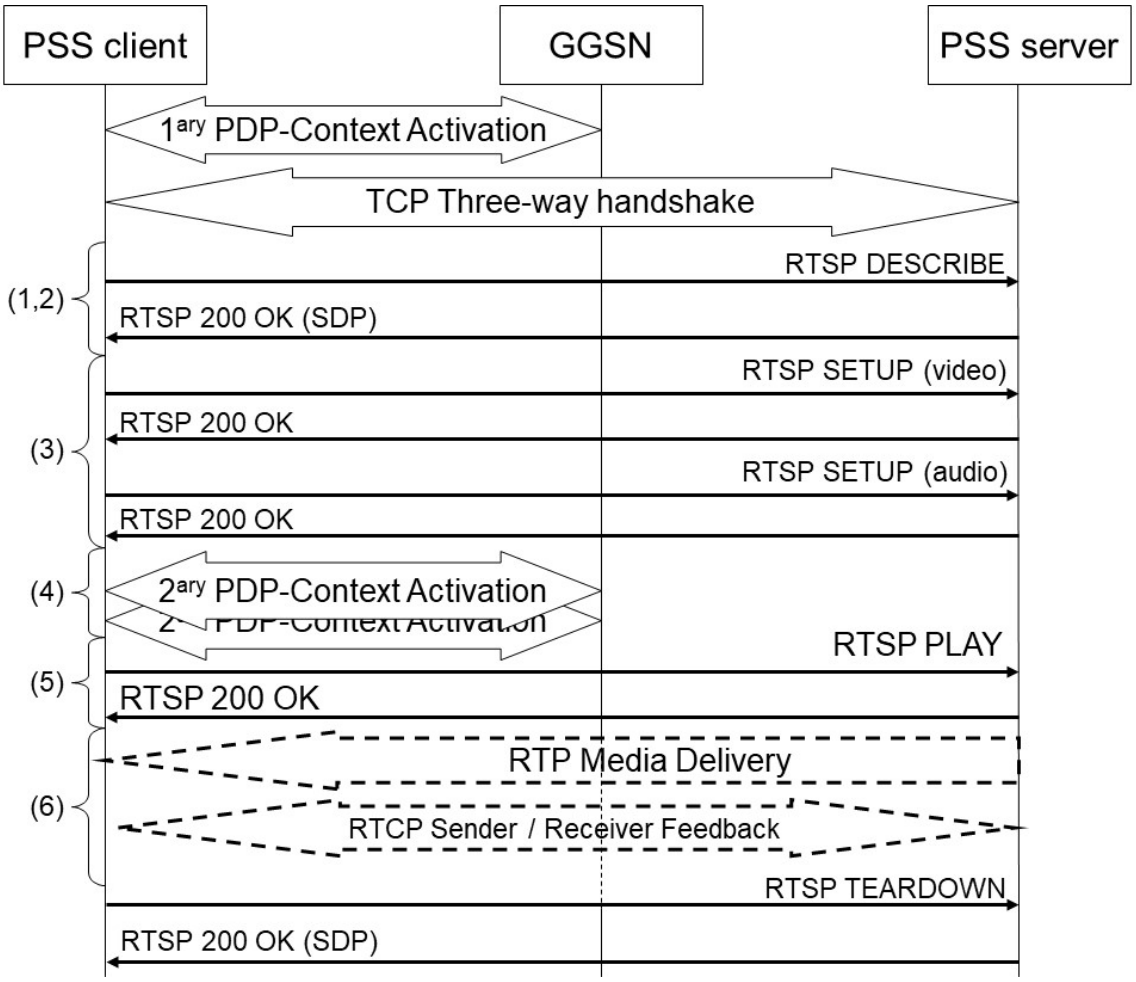

Figure 25. PSS session setup flow over a 3G network with QoS support.

First of all, the PSS application needs an active bearer to be able to start communicating with the PSS Server to trigger session setup. Such active bearer is the Primary PDP-Context, which is used to carry RTSP signaling information.

Once the Primary PDP-Context has been activated, the application will establish a TCP connection between the PSS Client and the PSS Server. This will require the traditional TCP three-way handshake, which generally involves one RTT to complete.

1. The first RTSP message sent by the MS requests the server a DESCRIption of the media content to be streamed based on the RTSP DESCRIBE transaction. The answer (200 OK) will contain

session information in the form of an attached SDP message. In general, we assume a session description with at least two streams, namely: an audio stream and a video stream.

2. The PSS client will use this description to decide which stream(s) –out of those which build up a multimedia presentation– are relevant for the upcoming session. The criteria used to perform this decision may involve processing parameters such as required bandwidth, client capabilities or user preferences. We will assume that the client intends to receive two streams (e.g.: an H.264 encoded video and an Enhanced AACplus audio stream).
3. The client issues one SETUP request per media stream it wishes to receive. In our case we will assume that two SETUP transactions are required. Each transaction is answered with a 200 OK message. Importantly, the 200 OK answer of the first SETUP message generally contains a Session: header with a unique identifier. From that moment onwards the client and the server will include this session identifier in all RTSP messages exchanged in relation to the same presentation (e.g.: starting, pausing, tearing down the session). This lets the client have one single URL to control the whole session, so that one command can be applied to several streams simultaneously (e.g.: to keep synchronized control of audio and media delivery).
3. Once all stream(s) is/are setup, the client has all required information to activate one or more Secondary PDP-Context(s) for media delivery.
4. After successful context activation, the client issues an RTSP PLAY message.
5. At this point, RTP media delivery takes place over the wireless network. Generally, RTCP information is exchanged end-to-end to monitor evolution of media delivery during the whole session.

In a nutshell, a session setup process involving TCP handshake, RTSP DESCRIBE transaction, two RTSP SETUP transactions and one PLAY transaction will generally involve 5 RTT's prior to media reception.

4.6.2 Setting up bearers with QoS support: the PDP-Context concept

3GPP defined an end-to-end QoS architecture originally in [37]. Later on, with the expansion toward 4G, part of the complexity of the original 3G QoS architecture was simplified and the Quality-of-Service Class Indicator (QCI) was defined in the new policy and charging architecture [47]. However, some of the key concepts such as the definition of QoS profiles (QCI's), usage of dedicated bearers for different types of IP flows or the definition of a process to request activation of bearers with specific QoS requirements are essentially similar across 3G and 4G. In the rest of this section we will present the procedures as described originally in [46] [18].

The mechanism implemented by the 3G network to transfer information over the Packet Switched domain is the *Packet Data Protocol* Context(PDP-Context). A PDP-Context is a logical association between a Mobile Station (MS) and the 3G infrastructure, so that the network:(a) provides end-to-end routing of user data

between the MS and an IP network (e.g. the Internet), and (b) allocates network resources to guarantee the QoS requirements requested by the application. When an application is aware that it may require exchange of IP flows with substantially different QoS requirements (e.g.: delay, jitter, reliability, bandwidth guarantees) and the network supports different QoS profiles, the application may set up several PDP-Contexts with different QoS requirements. As an example, applications with a clear split between the control and the user planes may use a Primary PDP-Context for signaling exchange (e.g. RTSP or SIP), and one or more Secondary contexts for user plane data delivery (e.g. RTP).

When two or more PDP-Contexts are used, one of them is the Primary PDP-Context, while the other ones are Secondary PDP-Contexts. In general, the signaling flow (RTSP) is carried over the Primary PDP-Context, while media flows are carried over one or more Secondary PDP-Contexts.

The basic idea behind the Secondary PDP-Context concept is that, when a handset requires receiving two or more IP flows with different QoS requirements, it must have the ability to indicate to the 3G network a hint to distinguish between those flows. The network must then ensure that each flow is treated according to its requirements (e.g.: in terms of bandwidth, delay, jitter, reliability, …).

The following diagram outlines a Secondary PDP-Context setup procedure by a PSS Client.

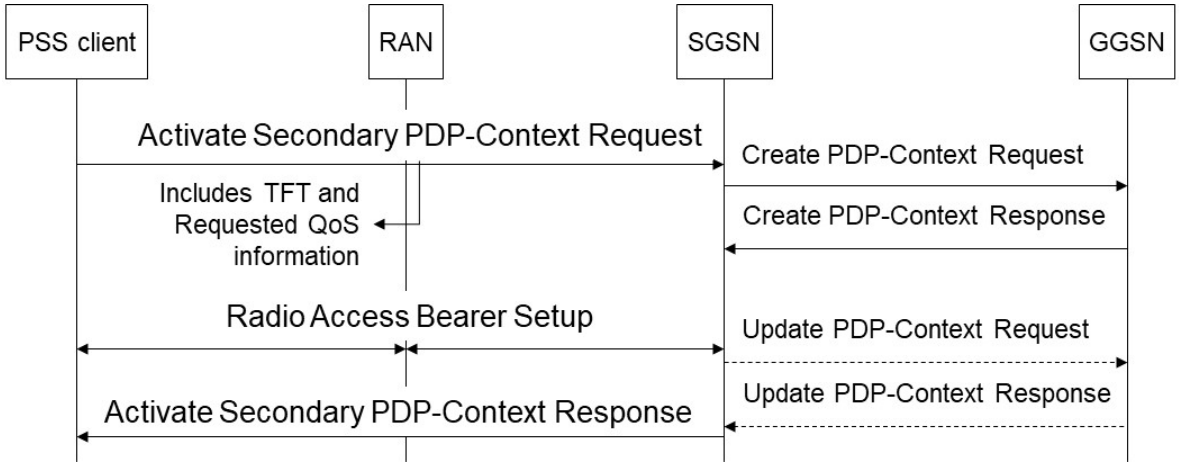

Figure 26. Secondary PDP-Context setup procedure.

The PSS client (or any application requiring specific QoS-enabled bearers) requests a Secondary PDP-Context by issuing an "Activate Secondary PDP-Context Request" message. In a 3G network, this message is terminated at the SGSN, which in turn queries the GGSN to activate the PDP-Context.

The "Activate Secondary PDP-Context Request" message contains enough information to define the flow for which QoS is specifically requested. To achieve this, the MS must include a structure in the signaling messages sent to the mobile network when setting up the Secondary PDP-Context. This structure is called

a Traffic Flow Template (TFT). The TFT contains information that should let the 3G network unambiguously distinguish new IP flows being setup (e.g.: media delivery over RTP) from existing flows already carried over the Primary Context (e.g.: signaling information over RTSP).

Upon receiving the "Create PDP-Context Request" the GGSN will query the PDF to request the bearer, which is eventually accepted. Note that all this procedure is internal between the UE and the 3G network, without intervention of the PSS Application Server. In a 4G context the PDP-Context request is terminated by the EPC, while flow authorization is managed by the PCRF. Furthermore, other scenarios such as network initiated QoS are possible, in which case the network is capable of assigning certain QoS-enabled bearers to IP flows, without the UE having to explicitly request or be aware of it. Network initiated QoS is out of the context of this thesis work.

The MS must also include information about the requested QoS settings that the new flow will require. These may include the Traffic Class, the required bandwidth, delay or jitter constraints, and requested reliability. The network will use this data to decide which Radio Access Bearer (RAB) and core network resources should be allocated to that flow.

The following table shows an example of the differences, in terms of QoS requirements and TFT information, between RTSP signaling and RTP media flows.

Parameter	RTSP traffic	RTP media
PDP-Context	Primary	Secondary
Example 3G Traffic Class	Interactive	Streaming
Example 4G QCIs	5	4
Example TFT info **- IP addresses** **- Protocol** **- Client Port / Server Port**	- Client IP / Server IP - TCP - 4,000 / 554	- Client IP / Server IP - UDP - 5,000 / 30,000
QoS requirements	Non-GBR (e.g.: 16kbps) <100ms delay 10^{-6} BER	GBR (e.g.: 284kbps) <300ms delay 10^{-6} BER

Table 1. Example PDP-Contexts for RTSP and RTP respectively.

Note that once the client has received media description over SDP during the DESCRIBE transaction it can initiate the setup of one or more Secondary PDP-Contexts of *Streaming* Traffic Class (or QCI 4 in LTE networks). When streaming two or more RTP streams (e.g.: audio and video) it is up to the implementation to set up several Secondary PDP-Contexts (e.g.: one for audio and one for video) or to use one single PDP-Context to aggregate all media delivery.

4.6.3 Requirements for the optimization of RTSP. The need to define pipelining support

As described in the previous section, the canonical PSS session setup (RTSP over TCP, setting up two streams) generally involves 5 RTT's prior to starting media reception. Furthermore, when session setup occurs over a 3GPP network with QoS support, the setup of one or more Secondary PDP-Contexts needs to happen prior to media reception, which may further impact session setup delay. Depending on how long it takes to set up a Secondary PDP-Context the procedure may further delay overall setup time. We will review the reasons for such calculation in deeper detail.

When the client issues the DESCRIBE message it shall receive an SDP description as a consequence. In general, by means of several structures of the form parameter=<value>, SDP provides information about the session name, the session originator, the media contained in the multimedia presentation (e.g. audio, video...) and mechanisms to allow proper identification of each media that may be streamed from the server. Figure 22 contains an example SDP file for reference.

The SDP provides information such as media types, CODECs and encoding information, required bandwidth, it also provides a control URL for each stream as well as for the whole presentation (notice the '*' attribute in the SDP file). As an example, the presentation described in Figure 22 contains one MPEG4 stream encoded at 88kbps. This video content can be streamed coupled with a narrowband audio track based on AMR-NB that consumes 14kbps, or alternatively, a high audio track at 64kbps can be used. Since we know AMR-NB highest encoding rate is 12.2kbps we can deduct that the b=AS:<value> parameter includes header overhead in addition to encoded media.

```
v=0
o=StreamingServer 3315947612
1106957545000 IN IP4 162.158.131.222
s=ClipName.3gp
u=http://www-entel.upc.edu/
  streaming.html
e=streaming@entel.upc.edu
c=IN IP4 0.0.0.0
t=0 0
a=control:*
a=range:npt=0-161.200
m=video 0 RTP/AVP 96
b=AS:88
a=rtpmap:96 MP4V-ES/90000
a=fmtp:96 profile-level-
id=0;config=000001B002000001B50EA0
   20202F00000100000012000C788BA9
   850584121463F
a=mpeg4-esid:301
a=x-envivio-verid:0001516B
a=control:trackID=65737
m=audio 0 RTP/AVP 97
b=AS:14
a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align=1
a=mpeg4-esid:201
a=x-envivio-verid:0001516B
a=control:trackID=65637
m=audio 0 RTP/AVP 98
b=AS:64
a=rtpmap:98 MP4A-LATM/24000
a=fmtp:96 profile-level-id=1;
bitrate=64000; cpresent=0;
config=9122620000
a=mpeg4-esid:101
a=x-envivio-verid:0001516B
a=control:trackID=65537
```

Figure 27. Example SDP file describing an audio / video multimedia presentation.

At this stage the client only counts with a description of the presentation. The client and the server have not agreed yet to start any streaming session. In order to do that the following things need to happen:

- The client needs to inform the server that it intends to SETUP a streaming session.
- As a consequence of this, the Server will include a Session: identifier in the first 200 OK response it sends to the SETUP request from the client. From that moment, the client and the server have an ongoing server concept, that they can reuse in the future.

- Once the client has received the first Session identification, it will include such identification in all subsequent messages sent in the context of the same multimedia presentation.
- The SETUP / 200 OK messages are also used to negotiate the IP address and UDP port where the client wishes to receive media, as well as the IP address and UDP ports from where the server intends to send media.

$$\begin{aligned}&\texttt{Transport}: \texttt{RTP/AVP}; \texttt{unicast};\\ &\texttt{client\_port} = 53680 - 53681;\\ &\texttt{source} = 217.226.43.54; \texttt{server\_port}\\ &\qquad = 6970 - 6971; \texttt{ssrc} = 00001753\end{aligned}$$

Figure 28. Example "Transport:" RTSP header exchanged during SETUP / 200 OK transaction.

Hence, until all SETUP transactions have not been completed it is not possible to issue the PLAY command. Importantly, in a 3G context with QoS support, session setup delay will be delayed even further, because prior to issuing the PLAY command the client has to perform bearer allocation based on the "Activate Secondary PDP-Context" procedure. Note also that this procedure can only happen once all SETUP / 200 OK transactions have completed (because they carry the necessary information to be used in filling in the TFT).

The reader may refer now to Figure 20 to understand that no other alternatives exist to the 5 RTT delay commented above, when setting up a multimedia streaming session composed of two media components. Note also that in addition to the 5 RTT's, in a QoS context, additional delay is incurred due to setting up one or more Secondary PDP-Contexts.

The main root causes of such prolonged session setup time are two:

- The delayed delivery of the Session identifier in the first SETUP transaction the client and the server quickly establish a session context.
- The necessary information to set up all the PDP-Contexts (in particular, Server ports allocated for each stream) are not known until all SETUP transactions complete.

Actually, the RTSP protocol refers to a "pipelining" concept, which allows a client to send several RTSP requests without having to wait for a response to the previous ones. However, this concept is not further developed. In fact, an implementation of RTSP pipelining according to [11] would lead to interoperability issues because:

- No pipelining can happen before the RTSP DESCRIBE transaction is completed (as the client still would not know the description of the streams to be requested in subsequent SETUP transactions)
- No pipelining can happen before the second RTSP SETUP request has been completed (the client cannot issue a PLAY message without knowing that the streams have successfully been accepted by the server).
- No pipelining can happen between the first and the second SETUP message (as the client should wait for the answer to the first SETUP message in order to receive the session identifier -by means of a Session: header- that must be placed in the second SETUP message).

Given the above limitations, it is not strange that there are no pipeline implementations based on RTSP 1.0 [11]: even though the concept is outlined in the spec, its practical implementation is simply not possible by design.

In conclusion, in order to reduce the number of RTT's and optimize RTSP/PSS session setup delay, there are two key requirements that need to be met:

- Ensuring the delivery of a session identifier during the RTSP DESCRIBE transaction so that subsequent messages can be pipelined as early as possible.
- Ensuring the delivery of endpoint information (e.g.: RTP media source and destination ports) during the RTSP DESCRIBE transaction so that Secondary PDP-Context activation can be triggered as early as possible.

Until now we have mostly reviewed the limitations of the RTSP / PSS setup procedure. We have also outlined some of the requirements that should be met by new procedures aiming to optimize session setup time. These requirements form the bulk of contribution [46] by the author. The rest of section 4.6 will provide a detailed description, additional context and expansion of the mechanisms proposed to implement the above requirements, into further optimizing RTSP / PSS session setup delay over 3GPP networks with QoS control, as contributed by the author in [18].

4.6.4 Proposed optimization. Enabling RTSP pipelining

4.6.4.1 Introduction

In the previous section we have described why pipelining is desirable (saving RTTs and setup time), the limitations of the RTSP protocol in order to support pipelining and the requirements that would need to be fulfilled in order to enable better pipelining support in RTSP (early sharing of a session-id concept and early sharing of transport endpoints information).

In our context, enhanced pipelining will lead to setting up multimedia streams earlier than with regular RTSP session setup. From this perspective, we will talk about *RTSP pipelining* or *RTSP early setup* interchangeably.

As we have seen in the previous section, setting up a PSS session in a 3GPP offering QoS requires establishing a Secondary PDP-Context. This procedure takes time and can increase session setup delay when compared to a 3GPP with no QoS support (in which case both signaling and media are carried over a default Primary Internet PDP-Context). In a nutshell, in a context where overall service delivery should offer a better experience (as the network offers QoS) one important key performance indicator (session setup time) is degraded precisely due to the way how RTSP and the QoS procedures in the 3GPP network behave and interact. The procedure is based on RTSP feature tags to let clients and servers synchronize about each other's capabilities.

One key goal of our *RTSP Early Setup* contribution is enhancing RTSP behavior to avoid the negative impact of the interaction between standard RTSP session setup and the Secondary PDP-Context setup procedure.

The main goal of the *early setup* concept is to provide the MS with the relevant information required to fill out the TFT at the earliest possible stage. This way, Secondary PDP-Context activation process can be triggered in parallel with remaining RTSP signaling, therefore reducing —potentially eliminating—impact in total session setup delay. The mechanism is considered to be an early setup procedure in the sense that it performs some of the tasks of the SETUP stage during the RTSP DESCRIBE transaction. Another benefit of the proposed mechanism is that, by sharing more information at an earlier stage, we can also save some RTT's from the standard RTSP session setup procedure, thus improving setup time regardless of whether QoS is used or not. We will describe the proposed enhancements in the following paragraphs.

4.6.4.2 Detailed description

In a 3GPP network with QoS our main interest is to trigger the Secondary PDP-Context procedure as early as possible and in parallel with the rest of RTSP session setup. To do so, the MS needs to know as much information about the intended (e.g.: audio and video) media tracks that will be streamed, as early as possible. This includes:

- CODEC to be used (exchanged during DESCRIBE transaction over SDP).
- Required bandwidth (exchanged during DESCRIBE transaction over SDP).
- Client and Server IP addresses and UDP ports allocated to each RTP stream (exchanged in the Transport header during the SETUP transaction for each stream).

- Finally, during the first SETUP transaction typically a session-id is provided by the Server so that both client and server share a common view and use a single control URL for all tracks exchanged during a presentation.

Our goal with the early setup process will be to exchange all this information as early as possible in order to speed up session setup, as well as to enable early Secondary PDP-Context activation in parallel with the rest of RTSP signaling.

In a nutshell, the mechanism works as follows:

1. The MS must indicate to the server the capability and willingness to support the "early-setup" mechanism
2. The "early-setup" flag exchanged in 1 informs the Server that the Client is willing to start media delivery immediately. When it receives this information, the Server must reserve resources for this session. Furthermore, it must furnish enough information during the RTSP DESCRIBE transaction, so that the MS is able to complete a relevant TFT to setup the Secondary PDP-Context(s) as early as possible (i.e.: at the end of the DESCRIBE transaction, as opposed as starting it at the end of the last SETUP transaction).

In the following paragraphs we will provide a detailed description of the procedure.

To trigger the Early Setup process, a MS starting a PSS session, will first signal support of this mechanism by adding a `Supported` RTSP header [12] with the early-setup tag to the DESCRIBE request, as follows:

```
Supported: early-setup
```

This tag must be used only when the client intends to start a streaming session immediately (i.e. the DESCRIBE message is to be immediately followed by the SETUP and PLAY sequence). The client may include a `Bandwidth` header in the DESCRIBE request to inform the server about current link characteristics. This helps the server determine which streams are relevant among those available in a given multimedia presentation. When receiving a DESCRIBE request containing the early-setup tag, the server will:

1. Reserve resources for the new session to be started. The server may evaluate the `Bandwidth` header of the request, if present, to determine which stream(s) are relevant to the MS.
2. Provide a 200 OK response with the following features:
    a. It must contain a `Supported` header indicating the `early-setup` tag.

b. It should contain a `Session` header with a session-id value. This communicates to the MS that the server has allocated a new session, and the MS may refer to it in subsequent RTSP messages. This header helps to speed the process further up, if combined with RTSP pipelining, as all subsequent messages can already use this new session-id. A timeout value can be inserted as well, to avoid blocking reserved resources for a long period of time.
c. It must contain an SDP body as described in the next paragraph.

3. The delivered SDP body must implement the following extra features:
   a. The connection (c=) header must include the unicast address of the RTP source (i.e. the server), in case that this is different from the RTSP end-point. This is in line with standard SDP [9], but not fully aligned with how SDP is used in RTSP [11]. This information is used to inform the MS about the actual source of RTP media at the DESCRIBE stage. In case media and signaling are terminated at the same address, the server may keep the null value to stay aligned with [11]. This is expected to be the general case.
   b. It must provide a UDP port number for each media line (m=) that has been reserved in Step 1. This port number will be the source port used by the server to send RTP packets delivered for each particular stream. Putting a non-null port value in (m=)edia lines breaks the RTSP standard [11]. However, given that in general, RTSP implementations leave the UDP port value in the SDP media line to 0, this new usage should have no relevant impact.

Summarizing, the server must provide the same set of information—using SDP connection(c=) and media(m=) lines— as it would do later on when inserting a `Transport` header in the 200 OK answer to a SETUP request.

Observe that SETUP message(s) must be sent anyway in order to inform the server about the MS port settings and to let any intermediate element (e.g. firewall) get all required information from the RTSP level. However, once the three steps described above have been implemented, the MS is able to uniquely identify each media stream that has been allocated for the session. This could involve, e.g., one video and one audio stream. At this stage, the MS can trigger activation of the Secondary PDP-Context(s) in parallel with the RTSP SETUP and PLAY messages, as opposed to doing it sequentially in regular RTSP/PSS session setup.

In case that the early-setup extension is not supported by the server, it would not recognize the early-setup feature tag. Hence, a regular DESCRIBE transaction would be completed, thus keeping backwards compatibility with standard RTSP protocol [11].

4.6.4.3 Conclusions

The Early Setup provides a mechanism to let the MS uniquely identify RTP streams that will be sent by the streaming server once the session is fully setup and started. This will let the MS trigger PDP-Context

activation in a faster way. Additionally, Early Setup introduces the Session concept (and header) at the earliest possible stage, thus easing RTSP pipelining as defined in [11]. Pipelining is a powerful concept to further reduce session setup delay. Although not described in detail in [11], it seems reasonable that messages sent without waiting for an explicit response from the server should have a common 'session-id' in order to ensure proper tracking and state transitions at both endpoints. The main drawback of the mechanism is that it requires slight breaks of the RTSP protocol as defined in [11][2]. Usage of the `Supported` feature tag minimizes implementation issues, as only endpoints implementing it would use the proposed procedure. Acknowledged misalignments from [11] are:

- Usage of the SDP connection field (c=) as representing the source of RTP media packets. As this field is recommended to be set to null for unicast streaming sessions, this is perceived as a not being acritical issue.
- Usage of the media line field (m=) to indicate source UDP ports for RTP media packets. This field may already be used by servers to recommend destination UDP ports to the client. This could be a mechanism used by service providers to recommend usage of firewall-friendly UDP ports and hence, losing this capability is a matter for further study before commercial implementation.
- Sending of the `Session` header within the 200 OK answer to the RTSP DESCRIBE message. RTSP identifies a session as the message sequence taking place between a SETUP and a TEARDOWN transaction. If this feature is implemented, this rule is overridden. This is not perceived as critical, as the server uses it just to reinforce the fact that it is aware that a PSS session will be started. Finally, backwards compatibility with legacy clients and servers is ensured by using a new feature tag.

4.6.5 Proposed optimization. SDP templates

4.6.5.1 Introduction

In the previous section we have described two important contributions: the requirements that should be met by an RTSP protocol optimization to speed up session setup time, and secondly we have proposed an implementation of the Early Setup concept that indeed improves session setup time in the context of the

[2] As we will note later on, it is interesting to see that when IETF decided to update the RTSP protocol in the new RFC that would be approved in 2016 (RFC 7826 [12], it also put particular focus in reducing the total number of RTTs and sharing a session-id concept as early as possible in the signaling flow, but we will focus on this comparison in section 4.6.7

3GPP PSS service. As a follow up, we will present theoretical results of the achievable benefits of these mechanisms in section 4.6.6.

Prior to that, we will still propose yet another enhancement of the RTSP/SDP protocol implementation in the context of the PSS. In this case, as part of our quest for optimizing multimedia setup protocols, we focus our work in reducing the amount of information exchanged over the air interface during the DESCRIBE phase.

This work was presented in our contribution at Wiley's Wireless Communications and Mobile Computing [18], where in addition to the Early Setup implementation proposal, we also present an additional optimization, so called "SDP Template". The SDP template mechanism is suited in cases where content providers may have a number of independent contents that have been encoded with common encoding parameters (e.g.: CODEC, encoding rate, available streams, …). As a consequence, when serving these contents, it can be expected that SDP information delivered by the streaming server to answer the RTSP DESCRIBE message will —in many cases— be similar across sessions and clips (i.e. as different clips are encoded using the same settings).

On the other hand, it is clear that not all SDP parameters will be exactly the same in all cases (e.g. not all clips will share parameters such as total clip length or source UDP ports used). In case each and every SDP parameter was equal and shared among sessions, a standard mechanism such as Content Indirection already defined for SIP-based services [48] could be easily adapted for RTSP to save signaling bandwidth.

However, it does not seem likely that in the scope of a 3GPP PSS service all and every session share a common SDP description, hence PSS will require a higher degree of flexibility about the contents of the SDP message. The SDP template proposal described here brings together the benefits of the Content Indirection concept without losing the flexibility of having a dedicated and different SDP message per content to be delivered. The PSS service already contain a procedure called "Capability Exchange" described in annex A of [30]. This mechanism lets clients inform the server about client capabilities without having to send a large amount of information (essentially a URL to a repository of client characteristics is stored centrally, so that only the URL is exchanged over the air interface). This mechanism does not let clients share "fractions" of SDP files, but is more aimed at sharing client capabilities (e.g.: supported CODECs, profiles, screen resolutions, …).

SDP templates as proposed in [18] applies a similar idea to SDP information in the case that a substantial fraction of SDP information is shared across a large number of multimedia presentations and remains stable over a long period of time. The main idea is to implement a generic 'template' mechanism that contains all SDP information which is common across a number of sessions. This 'template' file is downloaded once and can be cached as long as it remains applicable to streaming content. Once the SDP template content is

available in the UE it can be reused for future presentations that make use of the same template. If an SDP template becomes obsolete, future presentations may simply provide a URL to a different SDP template, which will again be downloaded once and be applied to all presentations that share the "core" SDP parameter described in the new SDP template. This is applicable, for example, if a content provider decides to upgrade its content (e.g.: upgrade to HD, or 4K, upgrade from MPEG4 to H.264, …) to a new baseline, which has an impact on a previously downloaded template.

Essentially, the idea is that by combining a static set of SDP information (contained in a so-called SDP template) with a set of dynamic SDP fields received when setting up a PSS session toward a specific content, the PSS Client will end up handling a full, standard and complete SDP description of the presentation. This way a total match between client and server SDP information related to a given PSS session is ensured.

Similarly, to the early setup concept, usage of SDP Templates can be managed through the `Supported:` header as well. If both endpoints support the mechanism, only differential SDP information will have to be exchanged, thus saving bandwidth and setup time, as stable SDP information is not transferred over the air.

In the following sections we present the SDP template mechanism that we propose. In turn, this work has been published as well as reference [18].

4.6.5.2 Detailed description

First of all, in order to ensure backwards compatibility, the following feature tag is proposed:

```
Supported:sdp-template
```

An example SDP template follows:

```
Sdp-template:http://www.entel.upc.edu/pss-3g-sdp-t.01.sdp
```

These headers can be sent both in the UE → Server RTSP DESCRIBE request, as well as in the Server → UE 200 OK response to the DESCRIBE request. The meaning in each case is described as follows:

- In the UE → Server direction it indicates UE willingness to use the SDP template mechanism. Also, the UE provides the template (or set of templates) it has available, which can be used for the content to be streamed. As an example, the template(s) URLs may have been received from the same PSS Server when streaming content in the past, so the UE assumes it is likely that new presentations will reuse the same template. However, the exact criteria why a given UE decides that a given (set of) template(s) is used is implementation decision.

- In the Server → UE direction it indicates Server willingness to use SDP templates. Furthermore, the `sdp-template:` header contains the specific SDP template that has been used to encode a given response. In the Server → UE direction, only one SDP template URL is allowed, because the rest of the SDP document provided in the 200 OK response [to the DESCRIBE request] is complementary to the SDP template provided in the SDP-Template URL.

In case the server detects a match between a URL sent by the client and the one sent by the server, the full mechanism described below will be used; otherwise, a regular complete SDP message will be sent to the client. In such case the Server may still decide to send an SDP template anyway. The Client may then download the template and cache it for use in future requests.

If a match between one Client provided `sdp-template` and the single Server provided SDP Template occurs, the Server only includes differential SDP information in the body of the RTSP 200 OK response to the DESCRIBE request. The client shall combine the cached content of the SDP template with the differential content received in the RTSP body to build up a complete SDP message.

In the following section we describe how an SDP template is built so that reconstruction of the complete SDP message can be achieved by the PSS Client.

4.6.5.3 Description of a template file

A template file must be a text file containing a regular SDP message as described in [9]. There may be some SDP fields, however, which will vary from session to session and will, therefore, be sent within the 200 OK response body to the DESCRIBE request. For such variable fields, the SDP template must contain an empty SDP tag (in the form `<character>=`), thus indicating that this field will be completed when getting the particular information of a session.

The following considerations must be taken into account:

- All differential fields (i.e. empty in the SDP tem-plate) must be filled out within the 200 OK message answering the RTSP DESCRIBE request.
- All fields must be sent in the same order as specified in the SDP template.
- If a field remains empty for a particular session, the `<character>=` sequence must be sent in any case.
- A field must not be split among ‘variable’ and ‘fixed’ part. Therefore, a given field must be sent completely either within the SDP template or within RTSP signaling.
- Additional SDP fields may be sent during the DESCRIBE transaction, although in that case the MS will assume that those fields are located at the end of the SDP message. Therefore, it will typically not be possible to modify any session-level attribute using this mechanism.

The above-described precautions will enable the client to build a fully standard SDP message without any uncertainty, therefore getting the session description once the DESCRIBE transaction is fully completed. For this purpose, the client needs only to match the template description with the differential fields received via RTSP signaling. An important advantage of the SDP template mechanism is that it is generic, and there is no explicit need to specify which SDP fields must be 'fixed' or 'variable'. The streaming server can decide how to perform the split. This flexibility allows supporting different types of SDP messages, and even proprietary SDP headers inserted within the session description.

4.6.5.4 Example operation

For the purpose of this section, the following example SDP message will be used (fields marked in bold are assumed to be dynamic and change from presentation to presentation):

```
v=0
o=StreamingServer 3315947612110695754500 IN IP4 162.158.131.222
s=ClipName.3gp
u=http://www-entel.upc.edu/streaming.htmle=streaming@entel.upc.edu
c=IN IP4 0.0.0.0t=0 0
a=control:*
a=range:npt=0-161.200
m=video 0 RTP/AVP 96
b=AS:88
a=rtpmap:96 MP4V-ES/90000
a=fmtp:96 profile-level-id=0;
      config=000001B002000001B50EA020202F00000100584121463F
a=mpeg4-esid:301
a=x-envivio-verid:0001516B
a=control:trackID=65737
m=audio 0 RTP/AVP 97
b=AS:14
a=rtpmap:97 AMR/8000
a=fmtp:97 octet-align=1
a=mpeg4-esid:201
a=x-envivio-verid:0001516B
a=control:trackID=65637
m=audio 0 RTP/AVP 98
b=AS:64
a=rtpmap:98 MP4A-LATM/24000
```

```
a=fmtp:96 profile-level-id=1;
      bitrate=64000; cpresent=0;config=9122620000a=mpeg4-esid:101
a=x-envivio-verid:0001516B
a=control:trackID=65537
```

This SDP document contains session-level and media-level fields, and provides information about a session consisting of two audio streams (AMR-NB and MPEG4 AAC) and a video stream (MPEG4). Variable fields are marked in bold font, and will be 'filled out' through the DESCRIBE / 200 OK transaction. Therefore, the SDP template would contain the whole SDP file except fields marked in bold (for which only the SDP parameter tag would be included). The following fields will be sent in a 'per-session' basis:

- The Range attribute, as it does not seem natural that total session duration is equal for a number of different contents.
- Media description line.
- CODEC specific configuration parameters which may change among sessions.

The above omitted fields are only provided for example purposes: each implementation may perform a different split, provided that the final outcome of the process lets the client build the original, complete and standard compliant SDP message again. Given the described SDP template, the RTSP 200OK message answering the DESCRIBE request would look like the one below (fields marked in bold are the ones that directly relate to the SDP template mechanism).

```
RTSP/1.0 200 OK
Server: DSS/5.0.1.1 (Build/464.1.1;Platform/Win32; Release/5;)
Cseq: 1
Last-Modified: Fri, 28 Jan 200523:23:19 GMT
Cache-Control: must-revalidate
Date: 03 Mar 2005 19:08:37 GMT
Expires: Thu, 03 Mar 2005 19:08:37GMT
Supported: sdp-template
Sdp-template: http://www.entel.upc.edu/pss-3g-sdp-t.01.sdp
Content-Type: application/sdp
Content-Base: rtsp://217.226.39.95/Hole1000.3gp
Content-length: 181

a=range:npt=0-161.200
m=video 0 RTP/AVP 96
a=fmtp:96 profile-level-id=0;
```

```
config=000001B002000001B50EA020202F00000100012000C788BA9850584121463F
m=audio 0 RTP/AVP 97
m=audio 0 RTP/AVP 98
```

This message assumes that the client previously notified support of the sdp-template mechanism, and provided a URL to the correct SDP template, which had been previously downloaded and locally stored at the client.

In the case of the SDP media line (m=), observe that a server implementing the Early Setup mechanism may be forced not to include it in the SDP template, as it would contain information about (potentially dynamic) UDP ports to be used for media delivery, this is why we assume that in general m= line fields will not be present in the static part (the template) of the SDP message.

Taking into account this example SDP message and the proposed template, we can roughly estimate the savings we can obtain by using SDP templates, in terms of less signaling bytes exchanged during session setup when compared with full SDP message transferred in each DESCRIBE transaction. Note that while the results will be specific to the example we are providing in this section, they can be trivially generalized to any SDP message and associated template used in real life.

In our particular case, the associated SDP message, SDP template and differential SDP info size values are calculated in the following table.

Parameter	Value
Total SDP body size[3]	806B
SDP template size	632B
Additional RTSP header info[4]	81B
Differential SDP info sent[5]	192B

Table 2. Example sizes of RTSP and SDP info when SDP template mechanism is used.

Considering that the SDP template is transmitted once and reused a number of times, the savings in total information sent over the air interface (for a given PSS/RTSP session) can be estimated as:

[3] In case no SDP template mechanism is used.

[4] Additional RTSP headers and info sent if SDP template mechanism is used.

[5] The contents of the SDP information sent in the 200 OK response to the DESCRIBE request, that complements the SDP template cached by the client.

Savings due to SDP Templates (Bytes): [806 bytes] − [192 bytes+2∗81 bytes] = 452 bytes[6]

In order to assess the benefit that the savings in signaling bandwidth may represent in terms of session setup time reduction, the reader is referred to Section 4.6.6.

4.6.5.5 Summary of considerations on SDP-Template mechanism

The SDP template mechanism aims at reducing session setup delay. While Early Setup targets early activation of a PDP-Context, SDP template focuses on minimizing the amount of PSS signaling exchanged over a (potentially) narrowband channel. Although SDP template inherits part of the rationale that motivated similar approaches such as SIP Content Indirection [48], the server must implement it in an 'intelligent' way: observe that effectiveness of the SDP template depends on the actual split between 'fixed' and 'variable' fields, and the overhead introduced in RTSP signaling to support this feature. In particular, the following condition must be met to ensure that this mechanism at least does not impact session setup delay negatively:

$$\left[\begin{matrix}Variable\ SDP\\ Fields\ Size\end{matrix}\right] + 2\left[\begin{matrix}RTSP\ Support:\\ Header\ Size\end{matrix}\right] + 2\left[\begin{matrix}RTSP\ SDP-Template\\ Header\ Size\end{matrix}\right] < \left[\begin{matrix}Full\ SDP\ Message\\ Size\end{matrix}\right] \qquad (1)$$

Equation 1. Threshold of usability of SDP templates vs. regular SDP body.

As the left-side of the above expression approaches the right side, the SDP template mechanism becomes less interesting.

4.6.5.6 Combining the RTSP Early Setup and SDP Template Mechanisms

Figure 24 presents an example signaling flow that showcases how the two mechanisms can be combined to setup a streaming session.

[6] RTSP `Supported` and `sdp-template` headers and contents are counted twice, as it is assumed that they are sent both by the RTSP Client and the RTSP Server during the RTSP DESCRIBE / 200 OK transaction.

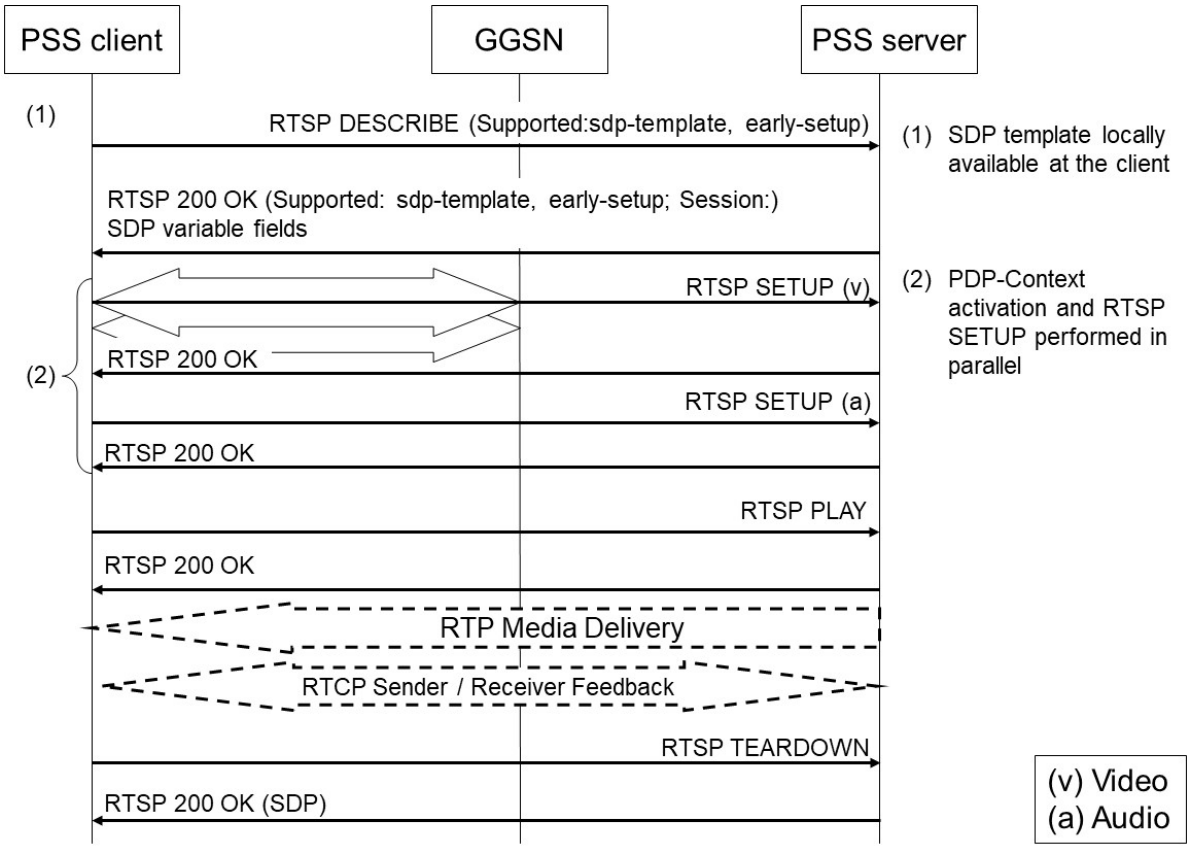

Figure 29. Example signaling flow implementing simultaneously *Early Setup* and *SDP Template* mechanisms.

By using the Early Setup and the SDP Template mechanisms in the same session, the best degree of optimization will be achieved. At the RTSP DESCRIBE stage, the client notifies the server that it is capable of supporting both features, using the `Supported` header. The server will then send a 200 OK message indicating which SDP template (among those indicated by the client) is effectively used for this particular session, it will include a `Session` header and it will only send differential SDP fields —those which are not included in the SDP template. At this stage, the client may trigger Secondary PDP-Context activation in parallel with RTSP SETUP signaling, thus avoiding extra delay due to limitations of the standard procedure, as described in section 4.6.1.

The benefits achieved by combining these two mechanisms are:

- Smaller amount of (SDP) information exchanged over a narrowband channel.
- Performance of two tasks in parallel (RTSP SETUP messages together with Secondary PDP-Context(s)activation). Whole in traditional PSS setup the two tasks contribute to overall session setup time, with Early Setup only the slowest task impacts session setup delay.

In the following section, we estimate which is the degree of improvement that can be achieved over a narrowband channel in a 3GPP network with QoS support, when these two mechanisms are used.

4.6.6 Theoretical evaluation of achievable setup improvements with enhanced RTSP

4.6.6.1 Introduction

In this section we will provide an evaluation of the theoretical improvements that are achievable if the two proposed enhancements (RTSP Early Setup and SDP Templates) are implemented over a 3GPP network with QoS support. In order to demonstrate such improvements, the link layer theoretical model described in [49] [50] is reused and extended, to adapt it to fit 3GPP PSS session setup modelling. The main extension precisely consists on the consideration of exchange of RTSP signaling over a narrowband WCDMA bearer.

It is important to note that if we consider RTSP as a signaling protocol (similar to SIP in an IMS domain) and we take into account the Control Plane vs. User Plane splitting that 3GPP defines in many services (e.g.: PSS, MCPTT, VoLTE / MMTEL, …), such signaling traffic may be exchanged over a narrowband, or shared, or non-guaranteed bearer. In a WCDMA environment 3GPP defines typical signaling bearers from 1.7kbps up to 33.2kbps on a 3G WCDMA network [51, 52]. In a 4G / 5G environment, IMS signaling (which we can associate to RTSP signaling, for the sake of our argument) would be carried over a RAB with QCI 5 with no guaranteed bitrate. Hence, the results of the study described in this section could well apply to a 4G or 5G environment in cases where signaling is transmitted in a shared bearer with bandwidth restrictions or in a narrowband bearer up to 33.2kbps.

After a description of the model, this is used to estimate the performance improvement that can be achieved by implementing the two PSS / RTSP session setup enhancements presented above (Early Setup and SDP Templates). Before entering such study, we will review the elements that contribute to the overall user perceived start-up delay in a PSS session deployed over a 3GPP network with QoS support:

- Primary/Secondary PDP-Context activation delay.
- Media player and Operating System activation delay. When the media player is invoked there is a startup delay until the application is fully activated.
- TCP three-way handshake between the client and the Server, to establish the RTSP signaling channel connection.
- RTSP and SDP session description and initiation (as described in section 4.6.1).
- Media buffering in the handset prior to playback. This delay contribution is highly dependent on network performance and buffering configuration at the handset (e.g. pre-decoder buffer [30]). At this stage, communication is fully setup, although the first frames or voice samples will not be displayed until the buffer reaches a minimum threshold.

Our focus will be on evaluating how the proposed RTSP modifications could help to reduce first, third and fourth contributions of the above list, as these are directly related to the signaling connection. Media buffering and rate control procedures are objects of study elsewhere [41] [53].

4.6.6.2 Link Layer Model

In this section, the model initially proposed in [49] [50] is adapted to the case of PSS signaling exchange over a narrowband signaling 3GPP channel. The focus will be put on how the 3GPP RLC protocol [52] is used. RLC is the link layer that governs the radio interface between the UE and the eNodeB. The RLC can work in different configurations (e.g.: unacknowledged mode, acknowledged mode, transparent mode, …). Each mode has different performance and reliability characteristics. In UM for example, a frame loss is not managed by the RLC protocol layer, but it has to be managed by upper layers (e.g.: TCP, application layer), while in AM the RLC protocol takes care of link layer retransmissions. Each RLC mode is suited for different types of usage (e.g.: interactive protocols, streaming delivery, …). As a consequence, RLC has a key influence in the performance of the radio link[7]. The link layer model will essentially simulate how RLC behaves in carrying RTSP interactive signaling traffic. For a more detailed description how RLC behaves in 3G and 4G networks, refer to [52] [54] [55] [56].

In particular, the following assumptions are made:

- The connection uses the Radio Link Control (RLC) protocol [52]. The RLC Acknowledged Mode (AM) is used to provide reliability at the Link Layer. The effect of link layer retransmissions due to lost or corrupted PDUs will be considered by calculating the performance of the system under different BLER conditions[8].
- RTSP messages are encapsulated into TCP segments over IP packets. These are the input Signaling Data Units (SDU) delivered to the RLC protocol, which are in turn converted into the RLC Protocol Data Units (PDU) transferred over the wireless link.
- The RLC sender transmits PDUs as long as new frames or requested retransmissions are available at the transmission buffer.

[7] Note that for the sake of completeness it is important to mention that RLC generally works in combination with two other link layer protocols, namely MAC and PDCP. For the sake of simplicity, we will focus our study on RLC, which –except under severe congestion conditions– is the one which has greater influence in the behavior and performance of the link layer.

[8] It should be noted that RLC AM is generally used for interactive (request-response) flows such as RTSP or SIP signaling. This does not imply that RTP media is actually delivered using the RLC AM. Depending on the type of bearer assigned to RTP data, other RLC modes may be used to deliver streaming media

- Since RTSP is a transaction based client-server protocol, the effects of the TCP window mechanisms will not be considered in this section (e.g. slow start, congestion avoidance). Observe that at the application layer, each segment must be answered by the remote endpoint in order to proceed with the session setup (e.g. RTSP DESCRIBE, 200 OK,). Hence, this seems an accurate assumption.

Based on the above assumptions it is possible to define an analytical model for the link layer that exists in the radio interface between the UE and the eNodeB. In general, the sender will fragment any SDU delivered by the upper layers (e.g. a TCP segment) into a set of n PDUs. Each PDU will contain P bytes of upper layer data plus the RLC AM header, which consists of H bytes. In the absence of errors, the only delay incurred during the communication would consist of transmission of the n PDUs and the corresponding propagation/processing delay (RTT/2) until the entire PDU is reassembled into the original SDU. When link errors are considered, some PDUs will have to be retransmitted. The following picture provides an overview of the RLC behavior at a glance.

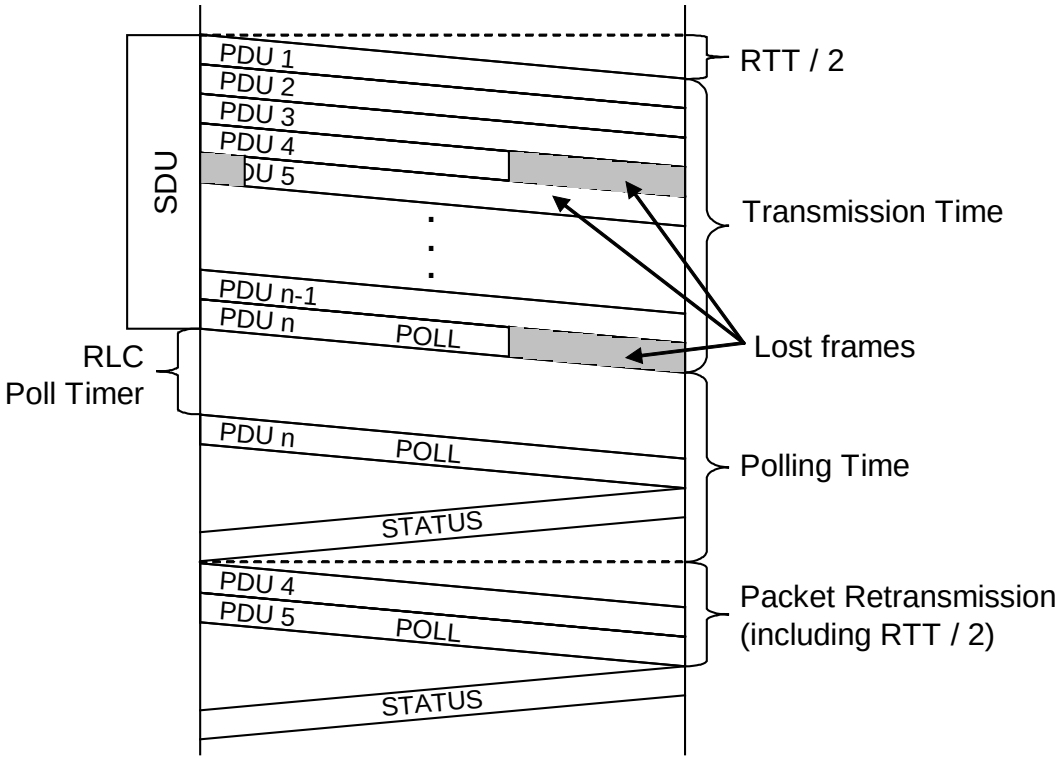

Figure 30. Overview of the RLC model.

In particular, if the Block Error Rate (BLER) over the radio link is ε and a given number of frames need to be transmitted (n), the actual number of frames transmitted over the air interface (n_{tx}) will increase to [50]:

$$n_{tx} = \frac{n}{(1-\varepsilon)} \qquad (2)$$

Given that some PDUs will be lost, the need for retransmission arises. The probability of not requiring any retransmission is simply:

$$P_0=(1-\varepsilon)^n \qquad (3)$$

Let's assume for a moment that once the sender sends n PDUs, the receiver will be able to detect which PDUs were lost or corrupted (if any). At this stage, the receiver could inform the sender accordingly and trigger a round of retransmissions of lost PDUs. Assuming that this generic mechanism exists, let's observe that the probability of requiring one single round of retransmissions is equal to the probability that in the first transmission, at least one frame was lost, while in the second iteration, all transmitted frames are correctly received. This covers all cases between: (a) one frame lost at the first transmission and all (n−1) frames transmitted correctly at the retransmission and (b) all n frames lost in the first transmission and all them received correctly at the second one. Analogously, the probability of requiring *i* retransmissions is equal to having between 1 (at least) and all n PDUs transmitted incorrectly in *(i−1)* iterations, but having all remaining frames correctly delivered in the *i-th* round. From [49] and [50] it is not difficult to observe that such probabilities can, therefore, be calculated by making use of the binomial distribution [50]:

$$\begin{aligned} P_i = & \sum_{k_1=1}^{n} \sum_{k_2=1}^{k_1} \cdots \sum_{k_i=1}^{k_{i-1}} \left[\binom{n}{k_1} \varepsilon^{k_1} (1-\varepsilon)^{n-k_1} \binom{k_1}{k_2} \right. \\ & \varepsilon^{k_2} (1-\varepsilon)^{k_1-k_2} \cdots\cdots \binom{k_{i-1}}{k_i} \\ & \left. \varepsilon^{k_j} (1-\varepsilon)^{k_{i-1}-k_i} (1-\varepsilon)^{k_i} \right] \\ = & (1-\varepsilon^{i+1})^n - (1-\varepsilon^i)^n \end{aligned} \qquad (4)$$

The average number of retransmissions can be calculated by:

$$\begin{aligned} N_{\text{retx}} &= \sum_{i=1}^{\infty} i P_i = \sum_{i=1}^{\infty} i \left[\left(1-\varepsilon^{i+1}\right)^n - \left(1-\varepsilon^i\right)^n \right] \\ &= \sum_{i=1}^{\infty} \left[1 - (1-\varepsilon^i)^n \right] \end{aligned} \qquad (5)$$

It must be observed that typical RLC implementations work with a limited number of allowed RLC retransmissions (i.e. every lost frame is retransmitted only up to a pre-defined finite maximum number of times, which we can call *Max_Ret*). As a consequence, the above expressions should only be evaluated up to *Max_Retx*. At the practical level, the *Max_Retx* parameter would lead to SDUs being lost at the link

layer, in which case TCP re-transmissions would be triggered. However, assuming a sufficiently high value of *Max_Retx* and sufficiently low of BLER probabilities, this effect is negligible.

We will follow this approach in order not to develop an unnecessary complex model for the case under study. As an example, for a segment size of 1460 bytes, BLER value of up to 10% and a *Max_Retx* value of 10 allowed retransmissions, the probability that TCP / Application layer retransmissions are required is less than 4×10^{-10}. The above assumption, however, may not hold true for scenarios in which BLER peaks are experienced (e.g. handover periods or users located at the limit of a cell). However, since our intention in [18] was to provide an initial estimation of the benefits of the RTSP enhancements, we did not develop the model further to cover corner cases such as high BLER scenarios. We believe the model proposed at [49] [50] provides a fairly accurate estimation of the general case: transmission of data over a 3GPP channel controlled by an RLC protocol that operates under certain target BLER conditions.

In order to trigger retransmission of lost PDUs, RLC allows a number of configurable mechanisms, including timer based, receiver and sender driven strategies. For the sake of this study, the approach in [49] is used: the main mechanism to trigger PDU retransmissions is the activation of a Polling bit when the sender transmits the last PDU available for (re/)transmission (the RLC 'Poll on Last PDU in buffer' mechanism) [52]. Upon reception of the 'Poll PDU', the receiver will answer back by sending a 'StatusPDU' which contains information about correctly received, lost and/or next expected PDUs.

In addition to the 'Poll on Last PDU in buffer' mechanism, a timer must also be implemented by the sender. This is to avoid that the loss of a 'Poll PDU' leads to the permanent loss of data. Therefore, upon the expiration of a given 'Poll Timer', the sender will re-issue a 'Poll PDU' until a 'Status PDU' is received back. Once a 'Status PDU' is received, the RLC sender determines if lost frames need to be retransmitted. Given that these two mechanisms are used (Poll PDUs and timers) and that a non-negligible probability that either 'Poll' or 'Status' PDUs may get lost in their way, the average time required to trigger a re-transmission should be calculated.

Without loss of generality for the purpose of our study, we assume that the 'Poll Timer' is configured to be equal to the estimated RLC RTT. If we observe Figure 25, it is clear that—considering apart propagation delay of transmitted and re-transmitted PDUs— the polling mechanism adds an extra delay of at least RTT/2 (when both 'Poll' and 'Status' PDUs are correctly received at each endpoint). We assume that 'Poll' PDUs can be lost with probability ε (the 'Poll' PDU is a regular AM PDU with the same loss probability as traffic PDUs). Initially, we will assume that 'Status' PDUs can be lost with probability ε_S.

Then we can calculate the 'Average Polling Time' as [49]:

$$T_{\text{poll}} = \text{RTT}_{\text{RLC}}\left(\frac{1}{2} + \frac{1-(1-\varepsilon)(1-\varepsilon_S)}{(1-\varepsilon)(1-\varepsilon_S)}\right)$$
$$= \text{RTT}_{\text{RLC}}\left(\frac{1}{(1-\varepsilon)(1-\varepsilon_S)} - \frac{1}{2}\right) \quad (6)$$

While initially ε_S does not have to be equal to ε (e.g.: due to eNodeB being able to transmit at higher power than the UE) we will simplify our study by assuming that the loss probability is equal in uplink and downlink directions, hence $\varepsilon = \varepsilon_S$. If so, the above expression can be simplified as follows:

$$T_{\text{poll}} = \text{RTT}_{\text{RLC}}\left(\frac{1}{(1-\varepsilon)^2} - \frac{1}{2}\right) \quad (7)$$

By inspection of Figure 25, once the sender understands that a retransmission is required, the additional delay incurred due to each retransmission itself is simply RTT/2. This is so because the transmission delay of the retransmitted frames is already considered in the n_{tx} value of expression (2). Hence, the overall delay for the transmission of a SDU can be calculated as:

$$T_{\text{SDU}} = \frac{n_{\text{tx}}(H+P)}{r} + \frac{1}{2}\text{RTT}_{\text{RLC}}$$
$$+N_{\text{retx}}\frac{1}{2}\text{RTT}_{\text{RLC}} + N_{\text{retx}}T_{\text{poll}} \quad (8)$$

Where H, P and r are the PDU header overhead, PDU payload size and channel bitrate, respectively. By replacing the expression for T_{poll} and reordering, we reach:

$$T_{\text{SDU}}(n) = \frac{n(H+P)}{(1-\varepsilon)r} + \frac{1}{2}RTT_{RLC}$$
$$+N_{\text{retx}}\text{RTT}_{\text{RLC}}\frac{1}{(1-\varepsilon)^2} \quad (9)$$

We use the above expression to perform an estimation of the improvement in RTSP setup delay provided by the proposed enhancements. To achieve this, the following two assumptions are made:

- Fixed network delays and packet losses are negligible when compared to the wireless link.
- Since RTSP requires exchange of application level data in uplink and downlink direction, both the handset and the 3G RNC / 4G eNodeB must implement an AM RLC transmitter and receiver. Given that both entities operate at different frequencies, it is not precisely accurate to assume that $\varepsilon_{UL} = \varepsilon_{DL} = \varepsilon$. However, in order to ease calculations, we will assume that these values are identical. This is similar to assuming ideal closed-loop power control available at each endpoint.

Under the above assumptions, the overall setup delay of a 3GPP PSS session setup over a WCDMA channel governed by the RLC protocol in Acknowledged Mode can be simply calculated as:

$$T_{\mathrm{PSS}_{\mathrm{SETUP\_DELAY}}} = \sum_{i=0}^{N} \left[T_{\mathrm{SDU}}(m_i)|_{\mathrm{UL}} + T_{\mathrm{SDU}}(n_i)|_{\mathrm{DL}} \right] \quad (10)$$

Where m_i is the size of the *i-th* SDU sent in uplink direction (e.g. RTSP DESCRIBE) and n_i is the size of the *i-th* SDU sent in downlink direction (e.g. RTSP 200 OK). *N* is the total number of RTSP transactions required to setup the PSS session. The convention has been made that *i=0* represents the initial setup of the TCP connection.

4.6.6.3 Estimation of PSS Session Setup Delay

After having developed the theoretical model, this will be applied to estimate the performance improvements that can be achieved when deploying the enhancements proposed before (RTSP Early Setup and SDP Templates).

The goal is to use the theoretical model described in the previous section to estimate overall PSS session setup delay over a 3G/4G channel using RLC AM as link layer protocol, comparing the proposed RTSP enhancements (i.e. RTSP Early Setup and SDP Templates) with regular RTSP setup procedure. A 3GPP network setup with QoS support is assumed, hence implementing the Secondary PDP-Context mechanism to deliver RTP media over a high capacity Streaming TC radio bearer. Observe that on top of RTSP session setup calculated in this section, the user will not start viewing the first clip frames until the handset buffer is filled up, which may approximately represent between 1 and 3 seconds of additional delay [30].

In order to apply the model, the following input parameters will be used.

Parameter	Value
Channel bitrate	8, 16 and 32 kbps
BLER (ε)	1%, 10%
PDU payload size (P) [49]	320 bits
RLC AM Header Overhead (H) [52]	12 bits
Secondary PDP-Context setup time[9]	1500ms

Table 3. Input parameters used in defining the 3GPP link layer and application level information.

Example RTSP message sizes (including TCP and IP header overheads) used in the calculations are shown in the table below.

Transaction	Regular RTSP		Optimized RTSP	
	UL (bytes)	DL (bytes)	UL (bytes)	DL (bytes)
TCP handshake	52	52	52	52
DESCRIBE	277	1054	358	440
SETUP (v)	389	525	389	525
SETUP (a)	414	517	414	517
PLAY	252	308	252	308

Table 4. Example RTSP message sizes used in the calculations.

Sample RTSP message sizes are taken averaging out the sizes measured in 40 tests performed with a 3GPP PSS client. Optimized RTSP values have been calculated by applying the criteria described in the Early Setup and SDP Template concepts. Additional bytes are considered when adding the necessary header information (e.g.: Supported: header). When using the SDP template mechanism, the actual SDP message and SDP template splitting as described in section 4.6.5.3 is used. Additional clarifications:

- We assume a multimedia presentation essentially consisting of an audio and a video track. Both tracks need to be set up (RTSP SETUP transactions for video and audio respectively). Once all tracks are setup a single RTSP PLAY request toward the shared control stream will trigger media delivery for both tracks simultaneously.

[9] Since PDP-Context setup runs over a Radio Resource Control (RRC) connection, a static value is used. It is consistently lower than measured times when setting a Primary PDP-Context with a 3G / 4G USB modem (in the range of 2.5–3.5th s). This is to take into account that Secondary PDP-Context setup should be fairly faster, as it reuses information partially available from the Primary context (e.g. allocated address).

- In the initial three-way TCP handshake, ACK message size is added to the DESCRIBE transaction (since final TCP ACK and RTSP DESCRIBE request are pipelined).
- When using the Early Setup concept, the two SETUP transactions are pipelined, since the Session identifier is assigned already during the RTSP DESCRIBE / 200 OK transaction.
- A Secondary PDP-Context is setup as early as possible in the process, in particular after completing the RTSP DESCRIBE transaction. Hence, Secondary PDP-Context operation typically starts in parallel with RTSP SETUP transactions to set the audio and video streams.

Taking all these assumptions and parameters into consideration, and applying the theoretical model described in the previous section, we estimate PSS session setup delay under different conditions. We assume that a signaling bearer is used to exchange RTSP signaling. The following three graphs display estimated session setup delay when an UL/DL bearer @8kbps, @16kbps and @32kbps is used respectively. Note that these values are generally aligned with the ones described in [51] for typical 3GPP signaling bearers.

The first picture depicts session setup delay when a UL/DL @8kbps signaling bearer is used, with different BLER conditions.

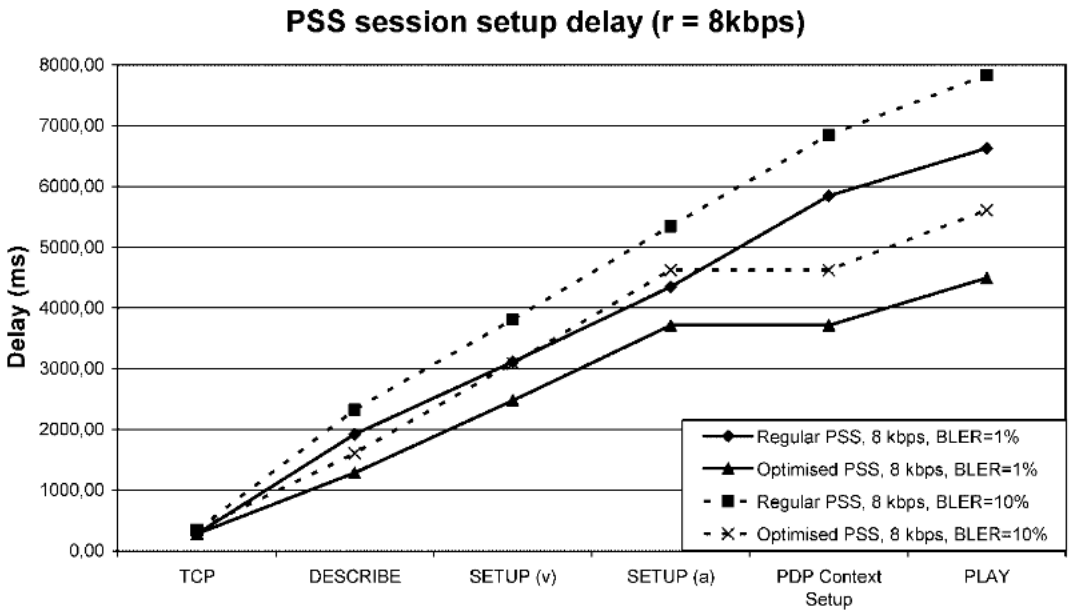

Figure 31. PSS session setup time over an @8kbps signaling cannel under BLER 1% and 10% respectively. Standard RTSP vs. Optimized RTSP applying Early Setup and SDP Templates are displayed.

In Figure 26, we can evaluate the evolution of session setup delay @8 kbps. Under BLER=10%, the non-optimized regular PSS mechanism requires almost 8 seconds to complete the whole setup process (excluding media delivery). On the other hand, optimized RTSP (i.e. including the Early Setup and the

SDP Template mechanisms) achieves the same result in approximately 5.6 s. Similar behavior is observed for BLER=1%, with 6.6 seconds (non-optimized) and 4.5 seconds (optimized), respectively.

Note that in the optimized case (in particular, when using Early Setup) Secondary PDP-Context process starts after completing the RTSP DESCRIBE transaction, and we have associated a total delay of 1.5 seconds to the Secondary PDP-Context setup procedure. When a narrowband channel of 8kbps is used to carry RTSP signaling, completion of the RTSP SETUP transactions takes more than 1.5 seconds, hence the Secondary PDP-Context setup procedure does not impact overall session setup time when the proposed RTSP optimizations are used. This is one of the scenarios when the proposed optimizations can be most beneficial, since –effectively– we are removing at least 1.5 seconds delay from the overall session setup time by performing two key operations (RTSP SETUP transactions and Secondary PDP-Context activation) in parallel, as opposed to regular RTSP / PSS, where such operations can only be implemented sequentially.

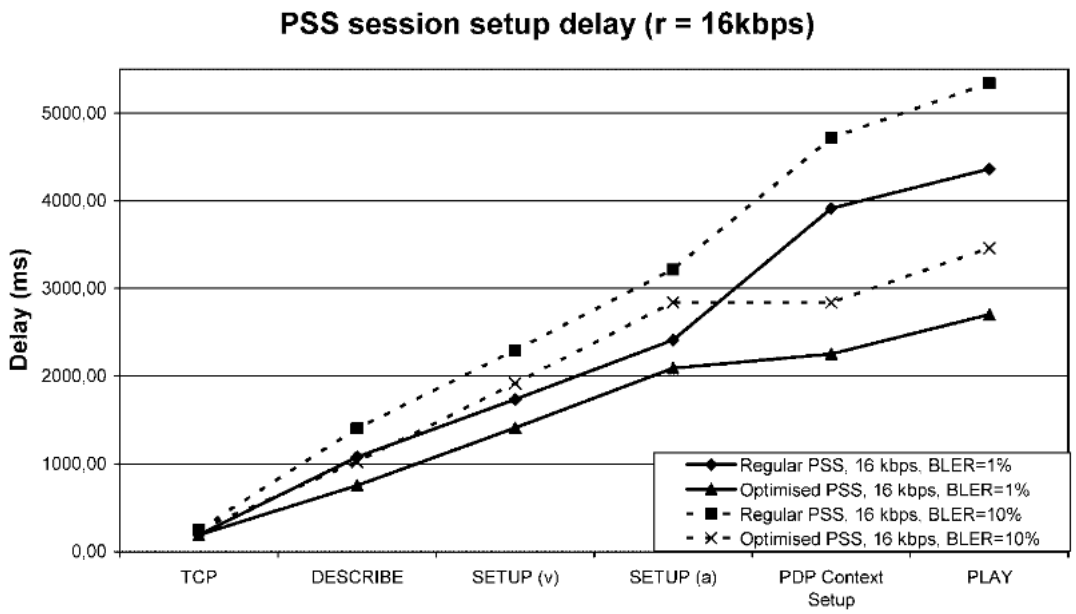

Figure 32. PSS session setup time over an @16kbps signaling cannel under BLER 1% and 10% respectively. Standard RTSP vs. Optimized RTSP applying Early Setup and SDP Templates are displayed.

Evaluation of the system at a rate of 16 kbps leads to similar conclusions. The main difference being that, in this case, as channel rate is higher, the RTSP SETUP transactions are completed in less than 1.5 seconds. There-fore, the PDP-Context activation procedure has certain impact over total session setup delay in the RTSP-optimized case. Still, the effect is significantly smaller than in the non-optimized case (where the Secondary PDP-Context setup contributes with the full extra 1.5 seconds delay). Session setup is completed

in approximately 5.3 seconds (regular PSS setup, 10% BLER) and 4.4 seconds (regular PSS setup, BLER 1%). When optimized PSS / RTSP is used (i.e.: Early Setup plus SDP Templates), total session setup time is reduced to 3.5 seconds (BLER 10%) and 2.7 seconds (BLER 1%) respectively.

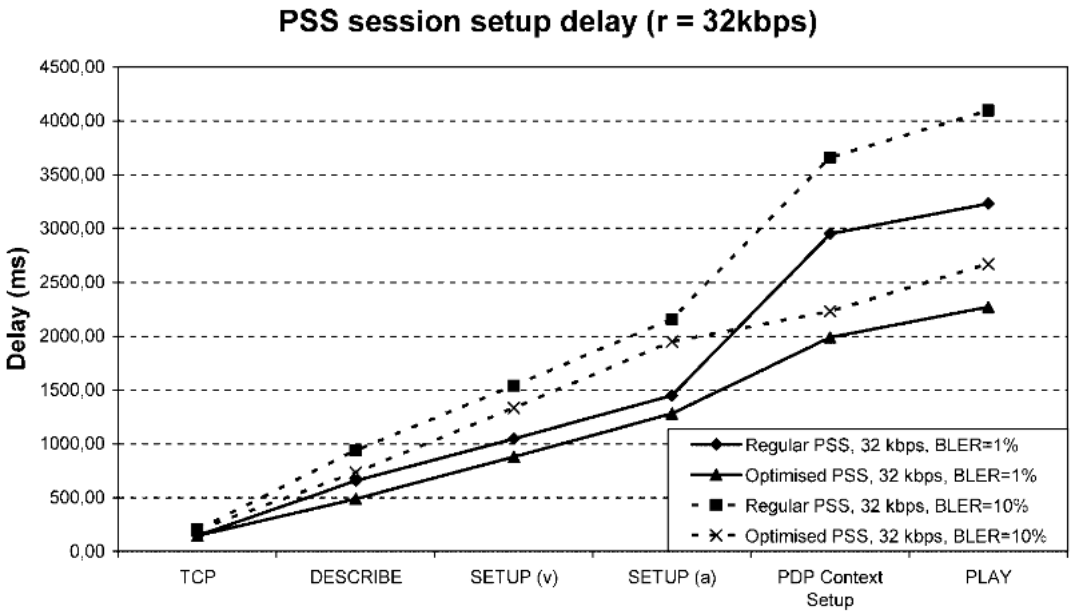

Figure 33. PSS session setup time over an @32kbps signaling cannel under BLER 1% and 10% respectively. Standard RTSP vs. Optimized RTSP applying Early Setup and SDP Templates are displayed.

When a @32kbps signaling bearer is used similar results are observed. The main difference with @8kbps and @16kbps is that, in this case, the Secondary PDP-Context procedure contributes with additional delay, because the faster signaling bearer lets the client and the server complete the two SETUP transactions significantly faster than when channels with less capacity are used. In particular the reader may note that there is a steeper slope in the graph from the SETUP (a) step until the PDP-Context Setup step in Figure 28 than in Figure 27 and Figure 26.

A comprehensive summary of results obtained when applying the link layer model calculations under different conditions is presented in the table below.

Channel conditions	Regular PSS (time, s)	Optimized PSS (time, s)	Improvement (%)	Improvement (time, s)
Setup time (@8kbps)				
BLER 1%	6.628s	4.494s	32.20%	2.134s
BLER 5%	7.164s	4.984s	30.43%	2.180s
BLER 10%	7.832s	5.611s	28.36%	2.221s
Setup time (@16kbps)				
BLER 1%	4.363s	2.703s	38.04%	1.660s
BLER 5%	4.804s	2.948s	38.63%	1.856s
BLER 10%	5.341s	3.462s	35.18%	1.879s
Setup time (@32kbps)				
BLER 1%	3.231s	2.271s	29.70%	0.960s
BLER 5%	3.624s	2.443s	32.59%	1.181s
BLER 10%	4.095s	2.669s	34.82%	1.426s

Table 5. Results comparison. Regular vs. Optimized PSS. @8kps / @16kbps / @32kbps. BLER 1% / 5% / 10%.

An overall evaluation of the results shows that optimized RTSP mechanisms (i.e. RTSP Early Setup and SDP Templates) provide a significant improvement in terms of reduction of PSS Session Setup time. Depending on channel conditions and available rate, the achievable figures range between 30% and 38% (relative) and 0.9s and 2.2 s (absolute). The main reasons for such an improvement are:

1. Usage of SDP templates requires the exchange of a smaller amount of data over a potentially narrowband (signaling) channel. This of course leads to shorter transmission time. An indirect consequence of the above is that for a given BLER value, exchanging less information requires a lower number of PDUs to be transmitted. Furthermore, the average number of retransmissions required is also reduced, leading to even shorter setup time.
2. Usage of the Early Setup mechanism allows that the PSS client triggers setup of the Secondary PDP-Context for media delivery as soon as the RTSP DESCRIBE transaction has been completed, and in parallel with the RTSP SETUP subsequent message(s). This allows that the effect of PDP-Context activation procedure on overall setup delay is either partially or completely hidden in the

optimized case. This is not possible when using regular PSS/RTSP implementations. Note that the benefit of the proposed SDP Template mechanism is maximized when the likelihood of using an already available template is high.

Note also that, while Early Setup and SDP Templates present an improvement when compared to regular PSS setup, the results presented in [18] did not incorporate the assumption of pipelining RTSP requests. Rather, the paper focused on the benefit of being able to trigger Secondary PDP-Context in parallel with other operations. In the following section we will expand the conclusions from [18] with the addition of the RTSP pipelining effect.

4.6.7 Conclusions

In the above sections we have described regular PSS/RTSP session setup process and indicated the challenges, in terms of session setup time, when using regular RTSP over a 3GPP network with QoS support. Those challenges mainly relate to the verbosity of the RTSP/SDP protocol as well as to the fact that, by protocol design, at least 4 RTT's are required to set up a PSS multimedia session prior to receiving real content.

In order to overcome these challenges, we have presented the Early Setup and SDP Template concepts in detail, and applied a theoretical model to describe the benefits that can be achieved when using them. Early Setup consists in providing enough media information to the PSS/RTSP Client to enable early activation of the Secondary PDP-Context. SDP Templates aims at minimizing the amount of SDP information transferred over the air interface, by gathering certain unlikely-to-change fields in a template that can be stored at the client. Both mechanisms minimize setup delays in a way that does not generate a big impact on RTSP and SDP protocol implementations. Applying the link layer model from [49] we can achieve benefits in the range of 30% in terms of overall session setup time due to RTSP signaling (i.e.: excluding delivery of first RTP packets and buffering effects).

In addition to this, it is still possible to get some additional benefit by incorporating the pipelining effect into the model, with the following assumptions:

- Since the Early Setup concept enables sharing a session id during the DESCRIBE request, it is hence possible to pipeline all subsequent SETUP (a), SETUP (v) and PLAY requests without having to wait to each individual response.
- As a final enhancement, we will assume that Secondary PDP-Context activation runs in parallel with the rest of pipelined requests. We will assume that if the PLAY request complete before the Secondary PDP-Context has been activated, the first RTP frames may flow over a default 3GPP bearer. Once the PDP-Context activation is completed, subsequent RTP frames may flow over a Streaming Traffic Class bearer (e.g.: QCI 2 or QCI 4 in LTE). This is

consistent with the fact that (e.g.: in a 4G / 5G network) it is possible to modify the QoS characteristics (e.g.: QCI mapping) of a bearer [47] [57].

Taking the above assumptions into account, using the packet size values described in table 4 and applying equations (5), (9) and (10) it is possible to estimate the minimum setup time when all proposed optimizations are combined. In the following figure we assume a 16kbps UL/DL channel used for RTSP signaling. Figures depict setup time under BLER 1% and 10% respectively. Three scenarios are displayed:

a. Regular PSS / RTSP procedures (including Secondary PDP-Context procedure impact).
b. Early Setup and SDP Templates (including Secondary PDP-Context procedure impact, but not applying RTSP pipelining).
c. Early Setup and SDP Templates including RTSP pipelining and assuming parallel Secondary PDP-Context setup procedure.

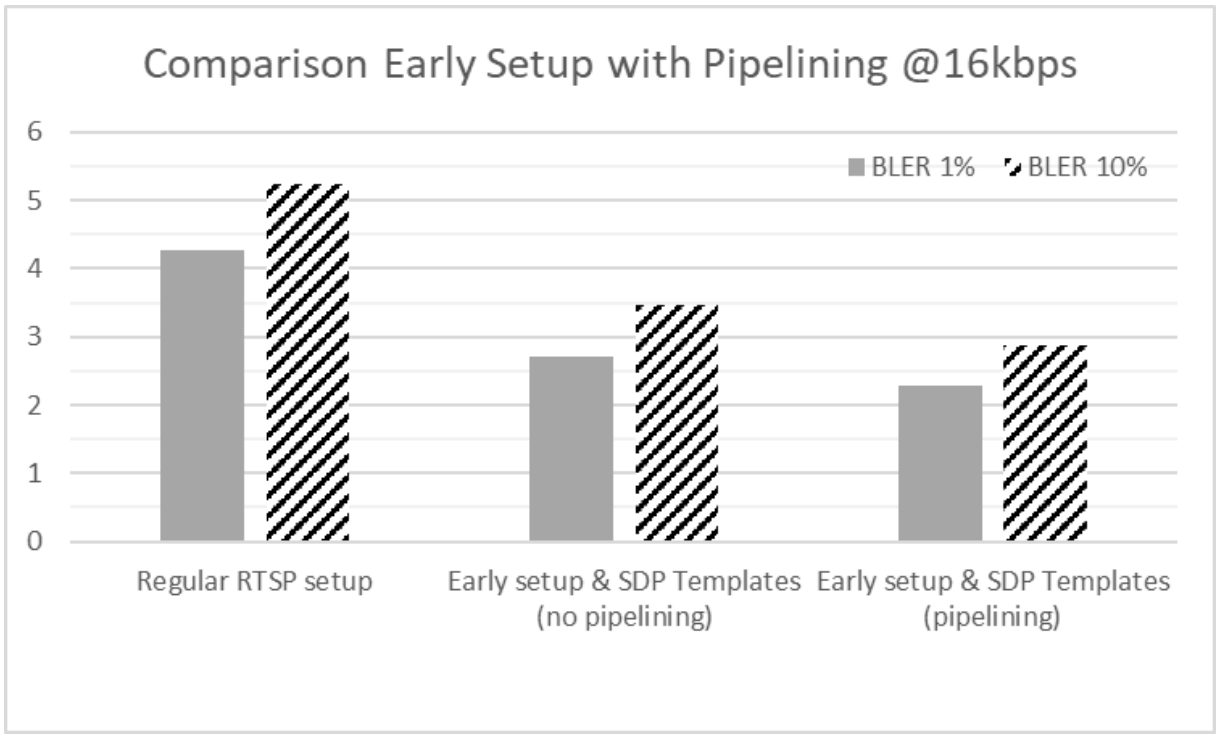

Figure 34. Minimum setup time comparison when RTSP pipelining is used.

As we can see above, minimum setup time when RTSP signaling is exchanged over a 16kbps bearer and BLER conditions are 1% is 2.221 seconds. This represents a further 17.8% improvement when compared with plain Early Setup and a 49.1% improvement when compared with regular PSS/RTSP session setup procedures. In a rich multimedia delivery context such as IP TV over 4G or 5G, setup time may be required when a user zaps through different media sources or channels. In such context, optimizing session setup time (i.e.: channel switching time) down to 2.221 seconds can help content providers and operators

significantly improve service interactivity, minimize channel switching time and better capture end user attention toward content delivery.

Finally, the reader is referred to Annex A, where more detailed example RTSP signaling flows combining all proposed optimizations are described.

4.7 Conclusions

This chapter has presented in detail and described the optimizations proposed by the author into the 3GPP PSS / RTSP session setup context, namely Session Information Based Admission Control [16], PSS/RTSP Early Setup and SDP Templates [46] [18]. Some of the main benefits of these contributions are:

- Enhancing network resource allocation due to better planning of future consumption based on session duration and bandwidth requirements, as specified in [16].
- Reducing media session setup time in 3GPP PSS by enabling RTSP pipelining of several RTSP SETUP messages [46].
- Reducing session setup and switching time in 3GPP PSS by caching frequently used unchanged information such as session description documents based on SDP format [18].

The proposed enhancements have been focusing on a PSS service mostly based on the RTSP / RTP protocol suite as originally defined by 3GPP [28]. Later on, during 2010-2014 timeframe, 3GPP Release-10 defined HTTP-based streaming as the 3GP-DASH service. In such context, both content and signaling are based on HTTP, as opposed to RTSP and RTP.

Regardless of this significant modification, the validity of some of the key concepts described in [16] [46] [18] expands beyond the scope of such contributions and remains still valid in the broader PSS concept as defined and developed by 3GPP in subsequent releases until present [58] [59] [60].

If we consider first [16], described in section 4.5 we outline the benefits of being able to leverage resource usage / capacity limits information from the network layer, with PSS session information, including the possibility to manage resource reservation based on the information known at the application layer about PSS sessions (e.g.: likelihood of session termination based on session duration, …).

In this context, it is worth noting that in a 3GPP network context not only an interface exists to push application-layer information toward the 3GPP network layer (e.g.: the PCRF) –in the framework of the Policy and Charging architecture– to allow for network enforcement of policies based on application level information, but also additional capabilities specifically developed from a streaming perspective have been recently incorporated by 3GPP.

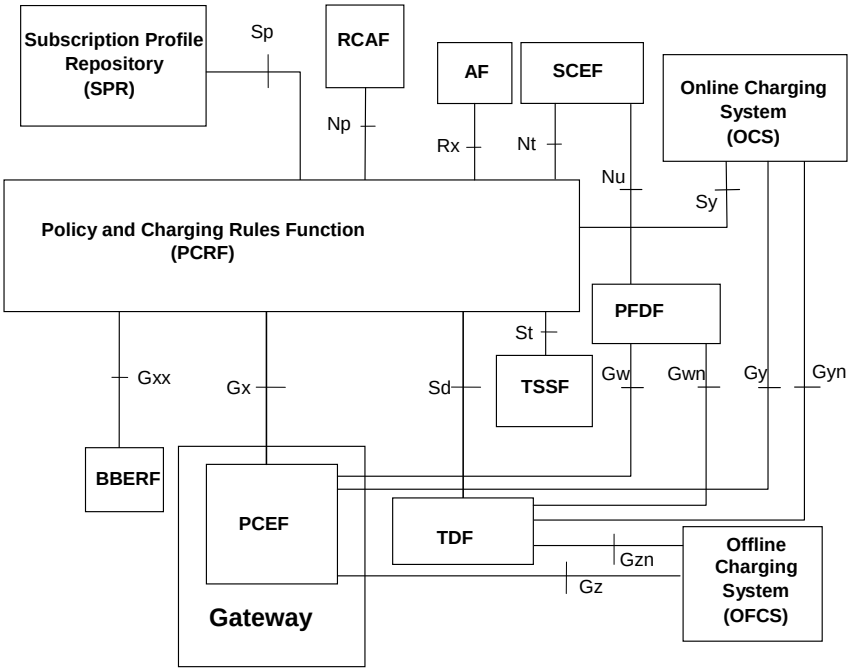

Figure 35. 3GPP Overall PCC architecture, including AF-PCRF Rx interface [61].

In a 5G context, the above architecture evolves into 5G System Architecture, in which still an interface between the Application Function and a Policy Control Function exists, evolving the Rx interface into the to-be-defined N5 interface

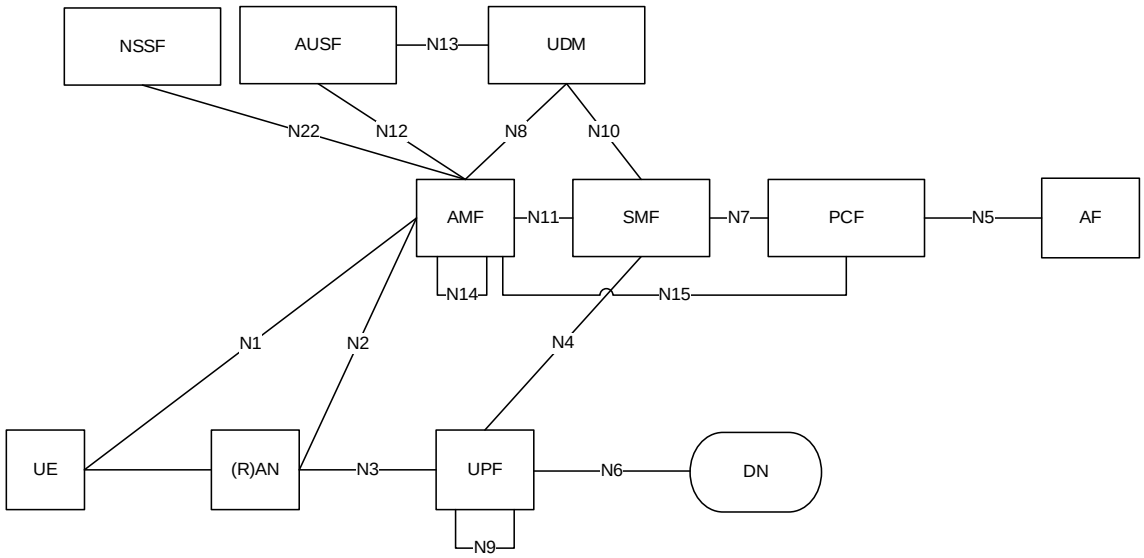

Figure 36. 5G System Architecture including AF-PCF N5 interface [62].

In addition to the general possibility to connect an application function with the network layer to provide policy information through the Rx interface, in a PSS context 3GPP DASH has adopted the so-called SAND and DANE concepts, which we will comment briefly.

As described above, from Release-10 3GPP has focused on the DASH concept as an HTTP-based streaming technology. Such mechanism is well suited to how Internet media streaming has evolved in recent years. However, HTTP-based streaming lacks all the adaptation capabilities that RTSP and –particularly– RTP-based streaming are able to support [41]. Under variable network conditions, RTP/RTSP-based streaming is generally capable of better adapting to them when compared to general HTTP-based streaming / 3GP-DASH.

With this in mind, the original 3GP-DASH architecture has been extended to define the possibility to deploy DASH-Aware Network Elements (DANE). A DANE is a network element involved in service delivery (e.g.: EPC, …) which –through some interface – is capable of signaling information about media delivery, characteristics, likelihood of future events, … When a DASH service is deployed on top of DANE elements, we talk about SAND (Server And Network Assisted DASH), in which both the Server and the 3GPP Network collaborate to deliver the best possible streaming service under certain (varying) network conditions.

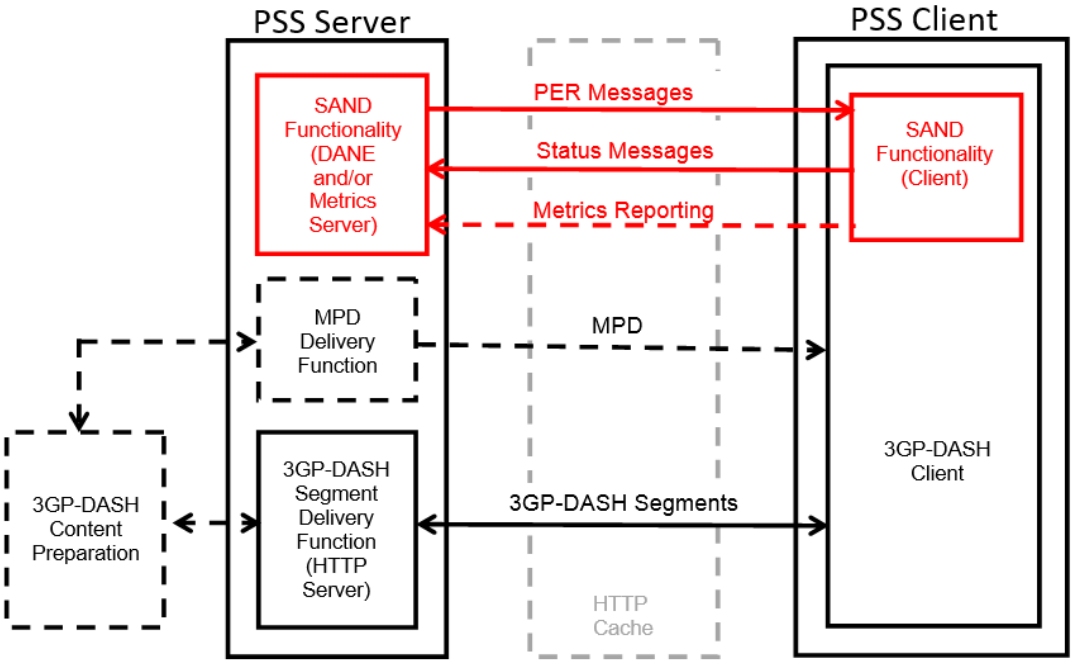

Figure 37. System architecture for SAND over PSS [60].

Obviously the concepts of DANE and SAND are much beyond the originals contributions by the author. However, already in [16] the importance of being able to leverage session information at the network layer was outlined.

When it comes to fast session setup and content switching, which were two of the main goals of [46] [18], these have been also considered by 3GPP. Just after the completion of both [46] [18] 3GPP expanded the scope of PSS in Release-7 / Release-8, to specifically improve session setup and channel switching among sessions. One of the main mechanisms proposed by 3GPP consists on the incorporation of a new RTSP header to enable early pipelining of SETUP requests. The new header was called `Pipelined-Requests` [63]. The main idea was that –after having completed the DESCRIBE transaction– the PSS/RTSP Client can issue pipelined requests using this new header. The Server should use this header to understand that those requests belong to a common session concept. Actually, this header is used while the Server has not allocated a context. The Server typically replies the first pipelined request with the allocated Session-Id. As soon as the client receives the Session-Id it may start using the `Session` header instead of the Pipelined requests header.

Eventually, the `Pipelined-Requests` concept was also adopted in the new RTSP 2.0 specification that would end up being approved in 2016 [12] by IETF in RFC 7826. Note that, from a formal and design perspective, there are a lot of similarities between the Early-Setup concept proposed in [18] and the `Pipelined-Requests` header [12]. In the following table we provide a high level comparison of both concepts.

Early-Setup vs. Pipelined-Requests	
Early Setup	**Pipelined Requests**
Backwards compatibility ensured through `Supported`: feature tag	Backwards compatibility ensured through `Supported`: feature tag
Requires DESCRIBE transaction to create context	Independent of DESCRIBE transaction
Based on RTSP header	Based on RTSP header
Allows pipelining all SETUP requests	Allows pipelining all SETUP requests
Transport info. shared during DESCRIBE	Transport info. shared during SETUP
Allows RTP punching after DESCRIBE	RTP punching after SETUP
Session info provided by Server during DESCRIBE transaction	Special temporary Session info. created by Client during SETUP transaction and put into `Pipelined-Requests` header

Table 6. Early-Setup vs. Pipelined-Requests procedures.

Each mechanism has a relevant benefit when compared to the other one: in the Pipelined-Requests context, if the client has a content description, the mechanism does not have a dependency on the DESCRIBE

transaction (while Early Setup does). On the other hand, since Early Setup allows sharing all relevant information during the DESCRIBE transaction, it is possible to send punch RTP packets to open NAT and firewall devices in parallel with the SETUP transactions (which is not possible with `Pipelined-Requests`, since the Server ports are not known upfront).

Further information about the exact reference to the Pipelined-Requests RTSP header in 3GPP PSS is described in Annex B.

Note also that in addition to the new features to pipeline requests described in 3GPP PSS and RTSP 2.0, 3GPP TS 26.234 also contains a number of enhancements aimed at optimizing or suppressing the transmission of SDP information (e.g.: refer to section 5.5 from [59]), which was also one of the goals of the SDP Template mechanism proposed by the author.

After the completion of RTSP/RTP-based PSS in Release-8 by 3GPP, with the introduction of IMS-based streaming, 3GP-DASH / DANE or IPTVoLTE services in subsequent 3GPP releases, the importance of interactivity, service-network collaboration, caching, fast setup and fast content switching have become of utmost importance, and the new architectures are being designed with such requirements in mind [64] [65] [66] [67].

At present, 3GPP is considering the creation of 3GPP 26.501 [68] to serve as the new 5G Packet-switched Streaming baseline spec, replacing 3GPP 26.233 [58].

5 Optimizations and evolved architectures to support IMS-based 3GPP Services and Mission Critical Push-to-Talk (MCPTT)

5.1 Introduction

After the completion of the research work in the area of RTSP-based streaming, our scope of investigation evolved slightly. Effectively, several market and technology trends made us progressively move our focus from RTSP toward SIP. We will briefly comment this evolution in this section. The rest of chapter 5 will provide an overview of our activities in this area.

First of all, it is important to note that RTSP and SIP are protocols that have a good share of common background. Not only were both of them standardized by Henning Schulzrinne, the father of all relevant Internet multimedia protocols, but also they are Client-Server, Text-based, Multimedia Signaling protocols. While RTSP is mostly aimed at multimedia content delivery from a content server and SIP was initially conceived to support peer-to-peer VoIP communications, there are a lot of commonalities among both protocols, with SIP having evolved in a way that has even taken over some of the use cases that were initially envisaged for RTSP at the conception of its first release back in 1996 – 1998.

At the completion of PSS Release-8 / Release-9, back in 2009 some industry trends were already happening, namely:

- In the web domain, the growth of the Flash© technology allowed many content and application providers to implement full web-based applications on top of the plugin from Adobe©. This technology enabled multimedia content delivery from a browser (no need to install a dedicated application after the plugin was enabled in the browser). This technology included also some advanced NAT traversal mechanisms which made it particularly convenient to use. Since Flash created a sandbox environment, applications ended up implementing their own proprietary signaling protocols, not necessarily relying on standards-based IETF protocols.
- The appearance of Apple iOS© and the Android© Operating System initially sponsored by the Open Handset Alliance and Google©, followed by the launch of the corresponding marketplaces, also opened up the possibility to dynamically and frequently install new applications in mobile devices. This was a dramatic shift when compared with the traditional mobile device ecosystem, based on a relatively stable number of actors and a combination of semi-open (e.g.: Symbian, BlackBerry OS) and closed Operating Systems (e.g.: BREW) with very limited programming capabilities (e.g.: J2ME, …). In such dynamics, in which also Adobe Flash was available in mobile

devices initially (as it was supported by a number of releases of the iOS platform), again the deviations from RTSP to build up content delivery solutions were very significant [69].

- The evolution of HTML and the initial works toward HTML5 were aimed precisely at enabling native support for real-time (audio and video) delivery directly from the browser, without the need for installed applications or plugins [70]. This would let streaming applications enjoy standard firewall / NAT traversal capabilities of HTTP, which have been in place for more than thirty years now. Furthermore, the implementation of the HTML5 DataChannel mechanism, that would allow usage of virtually any standard or proprietary signaling protocol, further opened up the possibility to use alternatives to RTSP.

With all these considerations, the options for massive adoption of RTSP as streaming signaling protocol in all possible environments, and hence the need for its evolution and extension did not ramp up significantly (it took 14 years for RTSP 2.0 to evolve from draft-00 [71] until approved RFC [12]). It is hence no surprise that the latest releases of 3GPP PSS leveraged on HTTP and evolved into 3GPP DASH / SAND as described in chapter 4.

With all this in mind, and with all the background and good work accumulated in investigating RTSP in particular and IETF-based multimedia signaling protocols in general, it was interesting to focus our attention on SIP-based services and architectures. There were a number of good reasons why SIP services would not be affected in the same way as RTSP by the market and technology trends described above, namely:

1. At the time when Android, iOS or Flash were disrupting the market, there was a large deployment and industry inertia towards deployment of SIP solutions as the baseline platform for next generation networks and interconnection of POTS / ISDN networks based on SIP trunking. This was mostly a need from the industry, as opposed to end users or end user devices, but helped to have a critical mass and adoption that set the foundation of consistent and sustained SIP evolution over time, particularly in terms of standardization at IETF and 3GPP as well as other fora.
2. Furthermore, during this period a good share of enhancements to support NAT traversal had been built into the SIP architecture [72] [73] [74]. This was a fundamental difference to RTSP, where the lack of traction never enabled the incorporation of proper NAT traversal into the core protocol.
3. SIP had been selected as the multimedia signaling protocol for the 3GPP IP Multimedia Subsystem (IMS) core platform. In particular, in the future evolution of 3GPP networks towards the All-IP paradigm, SIP was selected as the baseline signaling protocol for 3GPP services. As a matter of fact, native 3GPP calling (VoLTE) is heavily based on SIP and the IMS architecture (VoIMS). This provides a formidable ecosystem and installed base of SIP-enabled devices (natively implemented in the device and the OS in most cases). Further, SIP-based VoIMS will still be

present in the evolution towards 5G, hence enabling future evolution and extension of SIP to support the next 3GPP releases that will define the future of mobile networks.

4. Finally, it is also worth noting that through this journey, 3GPP has also defined and standardized the 3GPP Mission Critical Push-to-Talk service (in the framework of 3GPP Mission Critical Communications –MCC– services). While MCPTT / MCC represents a niche market aimed at Public Safety and Industry Critical applications, this market has an outstanding and fundamental need for interoperability, since critical communication users cannot afford proprietary solutions or vendor lock (as they have suffered over the last forty years). Hence, MCPTT / MCC is also a strong driver of standardized usage of SIP-based services over the IMS / 4G / 5G networks of the future that will serve the community of First Responders of the XXIst Century.

With all these considerations in mind, and with SIP having inherited a substantial share of design concepts for RTSP, we embarked in the journey of researching SIP-based architectures and services over 3GPP networks, with particular focus on Presence, PTT and MCPTT as we will describe in the following sections.

5.2 Research and standardization of SIP-based services over 3GPP networks

5.2.1 Standardization overview

During Release-5 / Release-6, 3GPP defined a fundamental architecture that would influence all future releases, from 3G down to 5G and beyond. Effectively, the selection of SIP as the signaling protocol for future All-IP based 3GPP networks, and the definition of the IP Multimedia Subsystem represented the foundation for the evolution of traditional circuit switched networks, that would expand into the fixed domain as well.

In turn, the IP Multimedia Subsystem (IMS) was based on the foundation of the multimedia protocols and technologies that had been created under the IETF umbrella at the end of the XXth century and first years of the XXIst, mostly led by Dr. Henning Schulzrinne, in strong collaboration with Jonathan Rosenberg (e.g.: SIP, SDP, RTP, XCAP, …).

IMS standardization evolved in Release-6 and Release-7, with the addition of QoS capabilities to support Real-Time communications over the packet-switched domain and a number of optimizations to support easy deployment of innovative multimedia services following a “plug&play” paradigm.

SIP was adopted as the application layer signaling protocol by 3GPP to support a new generation of multimedia services based on the packet-switched domain. It was envisaged that the future of circuit-switched telephony would involve the progressive migration of communication capabilities towards a single, converged, IP-based network [75]. 3GPP adopted this approach for the new generation of multimedia services that would eventually replace the circuit-switched 2G infrastructure. Having a set of

mobile-specific issues to be tackled, 3GPP had to consider some extensions or adaptations of the protocol to cope with them (e.g.: mobility management, security, scalability, roaming, domain interconnection, user identity, QoS, policing, …). At some point, the amount of profiling 3GPP was demanding to SIP triggered intense discussions, so as not to consider "3GPP SIP" as a SIP compliant implementation anymore. However, 3GPP and IETF signed a collaboration agreement [76], by which 3GPP should work as a "requirements generation" body towards a "protocol specification" body (IETF). This way, IETF could develop generic solutions, still taking into account 3GPP needs as an input. Only when a clear lack of an IETF-defined solution would exist, 3GPP would develop tailor made solutions, with the aim of replacing them as new IETF compliant solutions would become available.

Within this framework, and with the goal of defining the new range of mobile multimedia communication services based on the packet-switched domain, 3GPP started IMS standardization.

The 3GPP IMS architecture is depicted below.

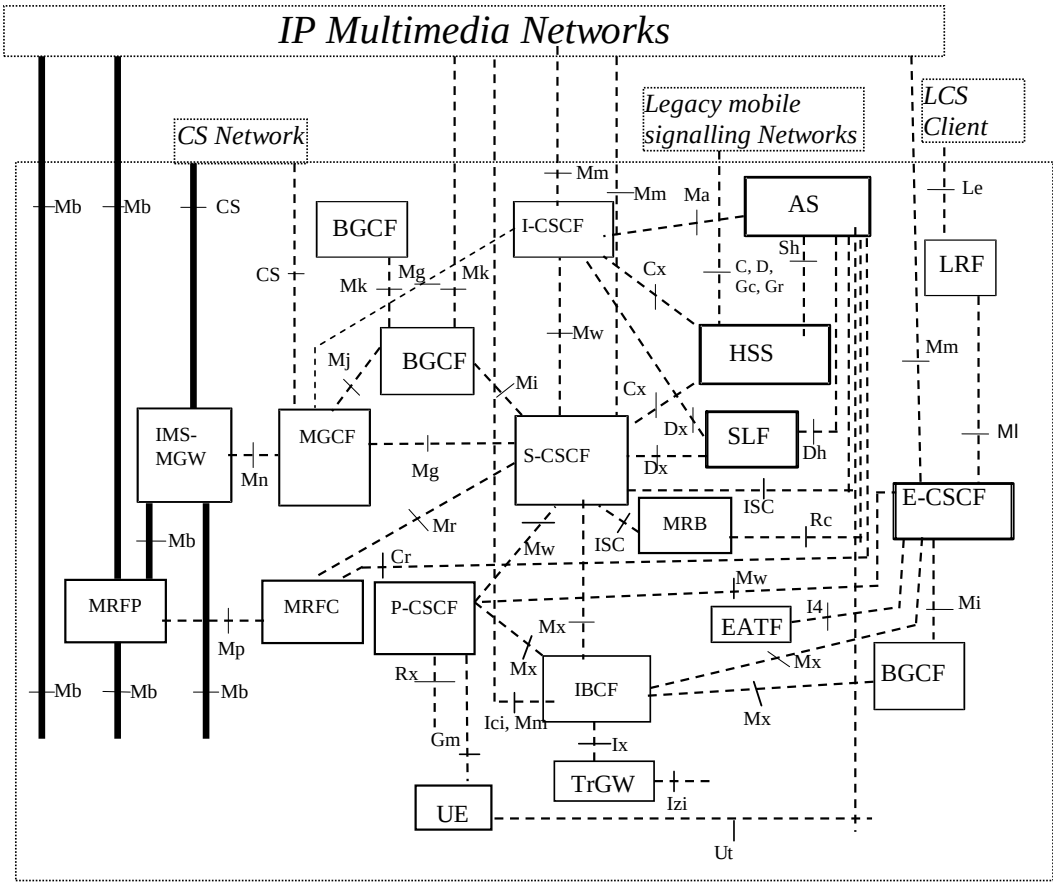

Figure 38. 3GPP IMS architecture [13].

On top of the initial 3GPP IMS architecture, a number of new features have been incorporated in subsequent releases, including but not limited to:

- The development of an access-agnostic IMS concept, based not only on 3GPP radio access networks but on any IP Connectivity Access Network (IP-CAN).
- Incorporation of Multimedia Telephony (MMTel / VoLTE), incorporating new media streams such as video or whiteboarding, which eventually evolved toward the Voice-over-LTE (VoLTE) concept, eventually replacing traditional Circuit Switched Telephony over 2G and 3G networks [77], including supplementary services support.

In the recent years, the incorporation of emergency sessions to IMS including enhanced location capabilities to locate emergency callers, Voice-Call Continuity (VCC) capabilities between VoLTE and CS calls, the support for unlicensed access to IMS (e.g.: WiFi, LTE-U, …), incorporation of policing and charging Diameter-based interface enabling for dynamic QoS / QoE, support for Mission Critical services and multicast capabilities, and support for WebRTC access have enabled the definition of a truly global communications enabler [78].

In parallel with the definition of the 3GPP IMS Core, the Open Mobile Alliance (OMA) was created to define application layer enablers which could be delivered on top of 3GPP architecture. 3GPP and OMA agreed on proceeding with service layer standardization activities at OMA, rather than at 3GPP (with OMA taking care that the enablers it would define should properly interoperate with 3GPP networks, and 3GPP defined IMS in particular). This split of activities lead to the definition of a split architecture, where OMA enablers would implement the role of IMS Application Servers. The first enablers to be defined by OMA would be Push-to-Talk over Cellular (PoC), XML Document Management (XDM) and Presence. An example architecture presenting PoC, Presence, XDM and 3GPP IMS elements is depicted below.

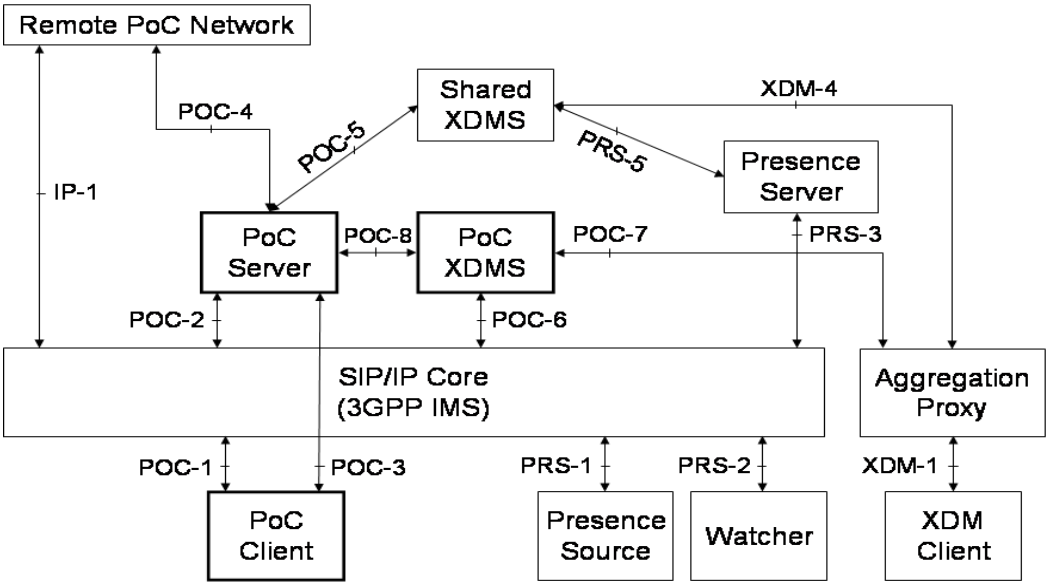

Figure 39. PoC, Presence, XDM and Presence SIMPLE architecture.

On the application plane, OMA developed the definition of a set of SIP/IMS-based service enablers: the Push-to-Talk over Cellular (PoC) enabler [79] [80], that would provide a walkie-talkie experience to groups of users, based on half-duplex VoIP sessions initiated via SIP, the XDM enabler [81] [82], that would enable users the definition of policies and service-related groups via the management of XML documents, and the Presence service [83] [84], enabling the delivery of real-time information about remote contacts' availability to communicate and communication capabilities. In addition, OMA also standardized the Instant Messaging (IM) service [85] [86] based on the SIP/SIMPLE architecture leveraging SIP and MSRTP for session-based IP messaging. In turn, SIMPLE IM was leveraged by the GSM Association (GSMA) to develop the so-called Joyn / Rich Communications Suite (RCS) service in order to develop a competing proposition with Over-the-top (OTT) Messaging applications such as WhatsApp, Lime, SnapChatt, Facebook.

OMA PoC, Presence, IM, XDM and GSMA RCS can be deployed partially or fully on top of a 3GPP IMS Core architecture. As opposed to the delivery of circuit-switched services in the PSTN or ISDN in the past, MMTel, Instant Messaging, PTT or Multimedia Streaming involve the provision of Real-Time or Near-Real-Time services over the packet-switched mobile domain.

While OMA Presence, IM and GSMA RCS were initially conceived as horizontal enablers that could target any type of consumer segment (consumer, business, government), the PoC service was already initially conceived as a business-oriented enabler.

During Release-13 3GPP has completed the definition of the Mission Critical Push-to-Talk (MCPTT) service, based on IMS, LTE and some of the concepts defined previously by OMA. In turn, 3GPP has also

"connected" the MCPTT service with other enablers such as eMBMS to enable efficient delivery of Mission Critical group communications over a broadcast / multicast bearer, as well as interconnection with the lower network layers through the PCRF for charging, rating and policy pushing.

The following picture depicts 3GPP MCPTT high level functional architecture as per [87].

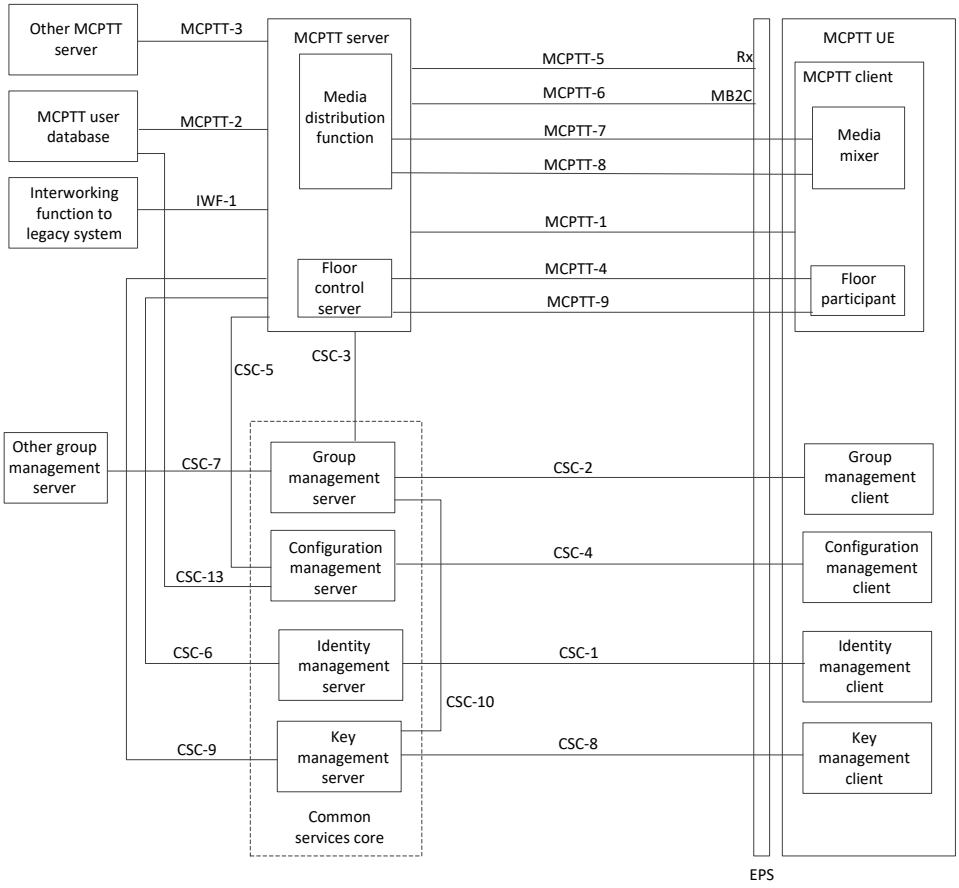

Figure 40. 3GPP MCPTT architecture [87].

Over time, with the incorporation of new service enablers defined by 3GPP, OMA or the GSMA, 3GPP networks have evolved incorporating services that involve significant data consumption (e.g.: PSS), real-time interactive delivery of information (PSS or SIP-based services), voice services (MMTel) and group voice communications among a large number of session participants (e.g.: PoC, MCPTT). Once 4G LTE networks are capable of handling such demanding services, it makes little sense to keep the traditional splitting between the packet domain and the circuit domain.

Simultaneously, it is worth noting that 3GPP-based networks, today and in the future, will emulate the services traditionally delivered by completely separate and independent networking systems. Effectively, massive content distribution was traditionally carried by the terrestrial broadcasting system, real-time voice and video communications were initially delivered by circuit-switched systems such as the ISDN and the

GSM networks, and large real-time group communication services have typically been supported by dedicated LMR / PMR systems. It is hence no surprise that significant evolution, optimization, research and standardization will be required in the coming years to optimize and adapt future 3GPP systems to support such diverse services, as well as those still to be envisaged.

As an example, it is obvious that these services will be faced with a number of challenges that are specific to the characteristics of the mobile environment or the wireless link. While IETF, 3GPP and OMA standards have focused over the time into definition of a set of interoperable technologies, a number of items have been left out of standardization scope, as a matter of differentiation among different vendor solutions, or as features for future study in later standardization releases.

While there is a broad research bibliography related to the performance of network and transport layer protocols over wireless links, there is still a broad space for application-layer optimization and adaptation to the specifics of the mobile domain, where users, applications and devices may experience issues such as bandwidth scarcity, delay variations, problems derived from fast mobility, screen display limitations or –simply– a general need to transmit application data in the most efficient way in terms of (e.g.) battery consumption, data volume generation or interference among mobile stations.

As an example, significant effort has been put since the definition of OMA Presence enablers, given the possibility that significant amounts of traffic could be generated by such service. [88] highlights the amount of traffic generated by Presence NOTIFY messages and develops a mathematical model to estimate and optimize it, while [89] develops mathematical models to derive Presence traffic estimates out of end user behavior who generates such traffic.

In this context, the author has focused his efforts in the evaluation, enhancement and definition of architectural or protocol enhancements in relation to two main services, namely:

- SIP-based OMA Presence over 3GPP networks.
- OMA Push-to-Talk over Cellular and 3GPP Mission Critical PTT (MCPTT) services.

These contributions are mainly described in sections 5.3 and 5.4 respectively.

5.3 SIP-based Presence optimizations over 3GPP networks

5.3.1 Introduction

Presence services have been a topic of research over the last 20 years. Appearance of SIP as the next generation signaling protocol for multimedia Internet services boosted interest in new communication models and paradigms. SIP related extensions were created under the umbrella of

so-called SIP SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions). Given the large scope of SIP SIMPLE, the main concepts are actually presented in umbrella RFC [90], however additional work has been carried out after its release.

The IETF SIMPLE Working Group has created a framework of SIP extensions that define what Presence information can be shared among users, how SIP is used to share this information, and how network entities should manage the Presence service [91] [92]. The Open Mobile Alliance (OMA) in turn specified a Presence service [83] [84] that reused SIMPLE work and defined additional implementation aspects, to ensure interoperability and adaptability to 3GPP networks (without losing generality to other IP-CANs).

Due to the "always on" nature of the Presence service, concerns raised among researchers, particularly related to service scalability, the amount of traffic it generates and battery consumption implications, specifically when Presence becomes ubiquitous and it gets deployed and used by many users and devices simultaneously [93] [94] [88] [95] [96].

One of the key mechanisms in OMA/IETF Presence to help reduce the amount of traffic and the number of simultaneous subscriptions maintained by a client is the "Presence List" subscription, based on the *Resource List Server* (RLS) concept [97]. We will provide guidance to calculate the amount of Presence traffic generated in different scenarios, and demonstrate that the Presence List mechanism outperforms the basic (individual) subscription mechanism in specific circumstances, but not as a general case. We also define calculations that can be performed by Presence clients to decide the optimal mechanism based on the observed traffic patterns and switch among them (Presence List vs. Basic) in a dynamic and adaptive way.

In the rest of section 5.3 we aim to provide a detailed description of SIP-based OMA Presence, as well as provide numerical guidance when certain mechanisms (e.g.: RLS based subscriptions vs. individual Presence subscriptions) are optimal, based on a number of assumptions and parameters.

5.3.2 Overview of the OMA Presence Service

OMA defines an *access agnostic mobile-friendly* Presence service [3] based on SIMPLE.

OMA Presence is based in the SIP Subscription–Notification framework, where a SIP client (a *Watcher*) *subscribes* to receive timely information of its interest (e.g.: Presence) about a remote

contact. When new relevant information is available, a *notification* (SIP NOTIFY) message is delivered. NOTIFY messages carry XML body with (e.g.: Presence) information for the Watcher.

The notification mechanism provides Watchers information about *availability* and *willingness* to communicate of their contacts. Contacts are called *Presentities* (Presence Entities). When a Presentity modifies its Presence status, a PUBLISH message containing XML encoded Presence information is sent to populate the new information. PUBLISH messages are sent by *Presence Source* applications *on behalf* of each Presentity. This information is translated into NOTIFY messages delivered to Watchers.

Communication between Presence Sources and Watchers are handled by the *Presence Server* (PS). The PS is responsible for accepting subscriptions (from Watchers), receiving Presence publications (from Presence Sources) and delivering the corresponding Presence notifications to all subscribed Watchers. An example scenario is shown below.

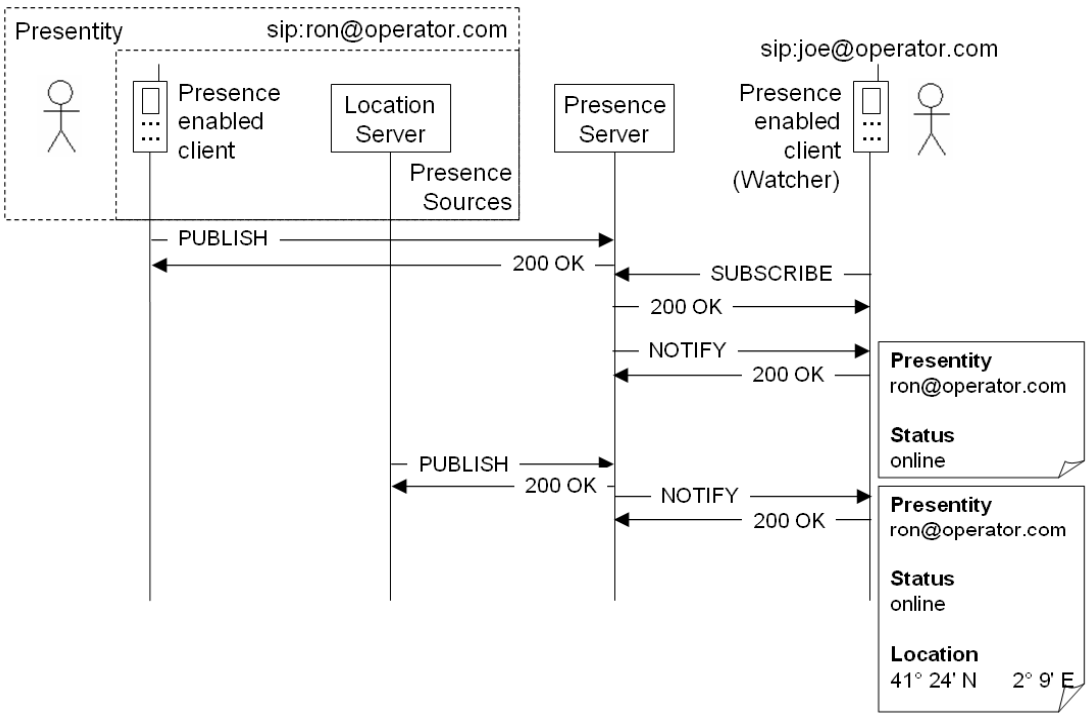

Figure 41. Example Presence Signaling Flow.

In this example two Presence Sources (a client device and a location server) publish Presence information on behalf of entity sip:ron@operator.com.

A multitude of Presence Sources may publish Presence information about one or more Presentities [3], thus generating relatively frequent notification messages to Watchers. Hence, the interest by researchers in reducing overall Presence traffic [93] [94] [88] [95].

5.3.3 The OMA Presence Architecture

Figure 2 shows OMA Presence architecture. Presence Server, Watcher and Presence Source functions were presented in section 2.1, so only new functions are described below.

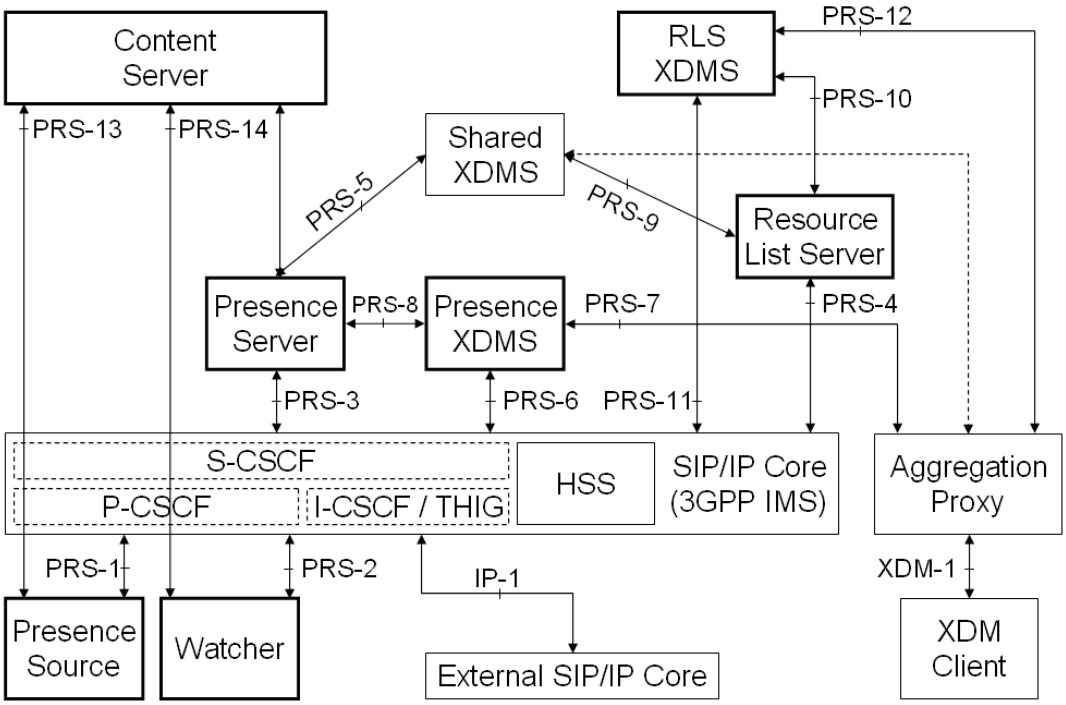

Figure 42. The OMA Presence Architecture.

The Presence XDM Server stores Presence and privacy policies defined by Presentities.

The Resource List Server (RLS) supports subscriptions to Presence Lists [97]. Presence Lists are XML documents stored in the RLS XDMS, and used to send a single SUBSCRIBE message to request information about a set of Presentities.

3GPP IMS is the standard SIP Core in cellular networks. It supports registration, SIP routing and interconnection of SIP domains.

5.3.4 Presence data format

Presence information may range from simple short documents stating “I am online” into more complex details, such as “I am available for IM and willing to chat. I am at home, listening to my

favorite song". Regardless of its complexity, Presence information is encoded in the form of simple [98] or richer XML documents [99].

5.3.5 Related Work

Since SIP-based Presence systems have been standardized by IETF and, particularly, since their consideration for the wireless environment by 3GPP and OMA, the interest about Presence traffic scalability and impact in network and service performance has grown up [94] [100]. One of the foundation studies about scalability and capacity impacts of the IMS-based Presence service was [94], where an overview of Presence capabilities and a calculation of the Presence-related message rates in an IMS network was presented and evaluated. This reference provides a good overview of the IMS Presence service, including end-to-end Publication and Notification processes, as well as the authorization and privacy considerations. Furthermore, it provides a detailed analysis of Presence service message rate at IMS network nodes such as P-/S-CSCF. In this chapter we will extend such approach by focusing on the specific OMA Presence architecture, including both RLS and PS cases (thus covering a more advanced architectural scope when compared to [94]) and we provide more detailed calculation for actual bandwidth usage based not on the number of exchanged messages but also on the actual sizes of these messages.

The work proposed by Chi [101] focuses on Presence traffic modeling from a different perspective. It analyses how aggregated Presence traffic impacts PS performance. It derives an analytical model and validates it through measurements from a real deployment. This chapter provides an alternative perspective, by providing the complementary Watcher perspective and the traffic implications in the client-server wireless link.

In [93] [95] [100] specific interaction of Presence traffic with other services is analyzed. The scope of these papers demonstrates a nontrivial correlation between Presence traffic and services such as VoIP [93] (where a down prioritization of Presence is suggested in order to avoid impact in VoIP traffic capacity) and Instant Messaging (IM) [95] (where latency increase due to large SIMPLE message traffic is observed). In turn, IETF work in [100] demonstrates that relevant cross-domain Presence traffic exists when interconnecting large Presence enabled domains. The authors have also contributed to [100] by pinpointing to the potential of using the *Content Indirection* mechanism [48] in the RLS – Presence Server path, to reduce interdomain traffic. All these conclusions serve as a basis to justify proper client design and optimization, which is at the core of the present paper.

In [96] several proposals are introduced to minimize unnecessary (redundant) Presence traffic during idle client periods (e.g.: active screen saver or locked keypad). However, the concepts presented in [96] require substantial extensions to existing standards. In contrast, the results of the work presented in this paper are not subject additional standardization work prior to their "implementability", but rather can be deployed immediately based on existing frozen standards after proper analysis by the relevant stakeholders.

Finally, it is worth referring to [102], which compares performance of individual subscriptions and RLS-based subscriptions. While [102] provides a deeper analysis of certain SIP/SIMPLE optimizations, our main contribution when compared to [102] is twofold: on the one hand, while [102] focuses on empirical results, we provide an analytical context to estimate in what conditions individual or RLS subscriptions are optimal; secondly, we compare IMS and regular IETF SIP profiles in the evaluation of SIP subscription procedures.

5.3.6 Estimation of load traffic generated by Presence subscription procedures

5.3.6.1 Introduction

In section 5.3 we will provide calculations to showcase the impact of Presence traffic based on different configurations or options, namely individual subscriptions and RLS-based subscriptions.

For the rest of this chapter we will use the parameters described in the following table. We use a simple scenario considering IETF-based Presence subscription, as well as a more complex scenario where specific 3GPP IMS signaling is used. In such case, generally 3GPP adds some additional headers and information into SIP messages, hence the different size when comparing [92] and [103].

Symbol	Description	Simple scenario [92]	IMS scenario [103]
T_{OBS}	Observation Time	-	-
T_S	Average subscription expiration time	3600 s - 43200 s	3600 s - 43200 s
T_U	Average time between Presence updates for each Presentity	-	-
C	Number of contacts in the buddy list	-	-
S	Average size of a SUBSCRIBE message	384 B	850 B
N	Average size of a NOTIFY message	412 B	632
O	Average size of a 200 OK message	284 B	372
P [3]	Average size of a PIDF document	650 B	650 B
R	Average size of a RLMI document	200 B	200 B

Table 7. Input parameters used in calculations.

In the following sections we provide an overview to estimate traffic load generated in both cases and, in turn, we compare the basic Presence subscription mechanism vs. RLS-based subscriptions, as described in the following sections.

5.3.6.2 Individual Presence Subscriptions

In this section we describe how the basic subscription mechanism works. Subscriptions are placed by Watchers to request Presence updates about Presentities, as follows:

1. The Watcher sends a SIP SUBSCRIBE message containing the identity of the Presentity for which Presence information is requested
2. The SUBSCRIBE message is routed to a Presence Server (PS), that eventually accepts the subscription by sending back a 200 OK message.
3. The PS sends a NOTIFY message to the subscribing Watcher. This message contains a PIDF document representing the status of the Presentity upon subscription activation.
4. While the subscription is active the Presence Server will keep sending NOTIFY messages whenever the Presence status changes. The Watcher acknowledges all received NOTIFY messages by sending back a 200 OK response.
5. The subscription may expire or be terminated explicitly by the Watcher. Alternatively, the Watcher may extend the subscription prior to its expiration, by issuing a new SUBSCRIBE message that triggers steps 2-5 again.

For a Watcher that subscribes to Presence about *C* contacts, steps 1-5 are repeated *C* times.

Now, we assume that each contact (Presentity) in the *buddy list* of the Watcher application modifies its Presence status every T_U seconds, on average –hence, a SIP NOTIFY message is received by the Watcher every T_U seconds on average. Considering this, and the fact that Presence subscriptions are refreshed every T_S seconds on average (step 5 above), we can derive an expression to determine the amount of information exchanged between the Presence service and the Watcher[10] (I_{BASIC}), over a given observation time (T_{OBS}).

$$I_{BASIC} = \frac{T_{OBS}}{T_S} C\left[S + N + P + 2 \times O\right] + \frac{T_{OBS}}{T_U} C\left[N + P + O\right] \quad (11)$$

5.3.7 Presence Lists

The goal of *Presence List* subscription mechanism [97] is to reduce the amount of traffic exchanged in the interface between the Watcher and the Presence service, and to minimize the amount of SIP dialogs required to subscribe to Presence information [97]. The underlying concept is using single subscription to *a list* of Presentities, so that the all notifications are consolidated into a single SIP dialog, as shown in the below figure.

[10] I_{BASIC} refers to the fact that this calculation shows the amount of traffic exchanged when using the "basic" subscription mechanism, as opposed to usage of the RLS optimization.

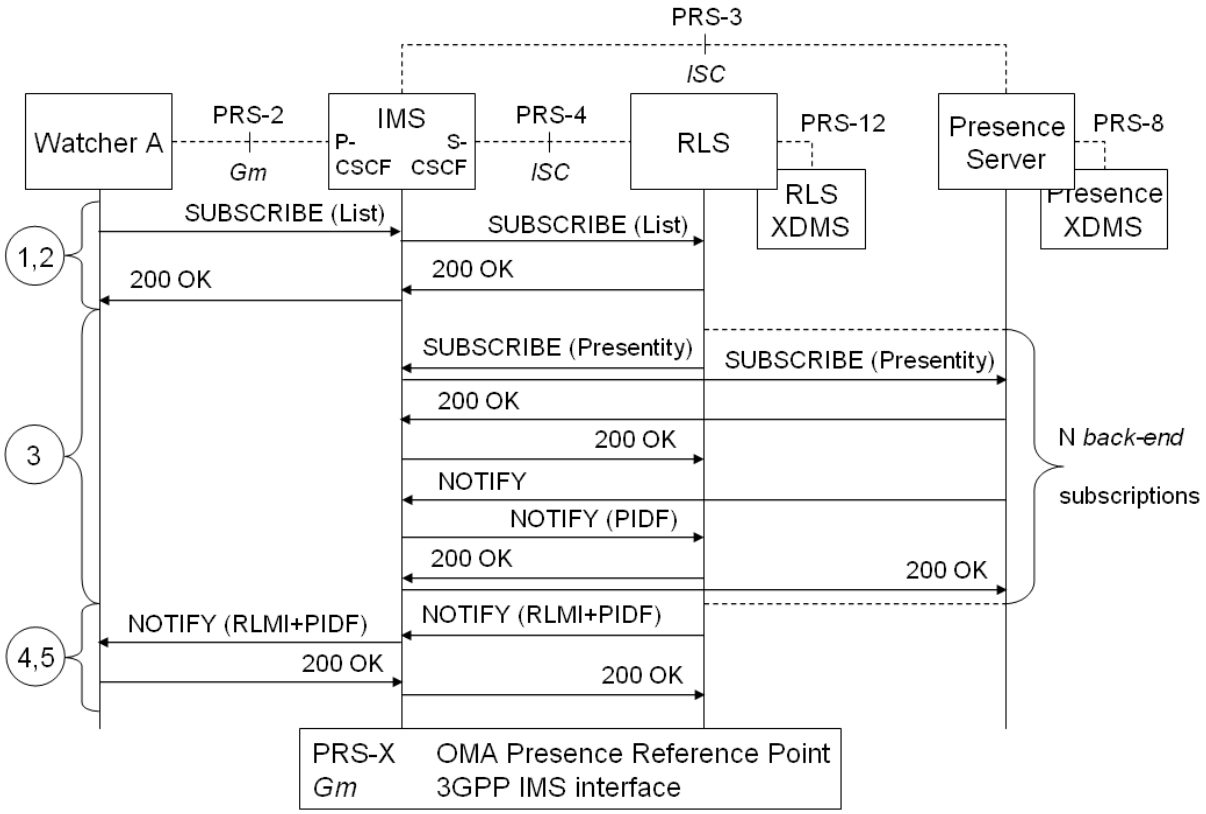

Figure 43. RLS Subscription mechanism.

A Presence List document that contains the list of Presentities is stored in the RLS XDMS. Once the list is created the following process activates the subscription:

1. Watcher A sends a SIP SUBSCRIBE message to the RLS, via IMS, to subscribe to the Presence List (sip:ron@operator.com;pres-list=mypreslist).
2. The RLS accepts the subscription and retrieves the Presence List from the RLS XDMS.
3. For each entry in the document, the RLS sends a *back-end* subscription to the PS.
4. Once the RLS has received information about all Presentities, it sends a single multiparty SIP NOTIFY request to the Watcher that contains:
    a. A *Resource List Meta-Information* (RLMI) documents describing the status of each subscription (accepted/rejected).
    b. A PIDF Presence document for each subscribed Presentity.
5. Upon reception of the NOTIFY message, the Watcher displays Presence information to the user and issues a 200 OK response.

As in the basic case, while the subscription is active, a SIP NOTIFY message will be sent to the Watcher every time a Presentity changes its status. In addition to the Presence PIDF document, all

NOTIFY messages sent by the RLS contain the RLMI document as well, conveying *back-end* subscriptions status.

An RLMI document contains a set of XML elements that represent the status of each *back-end* Presence subscription that the RLS has initiated on behalf of the Watcher. For our purpose, we will assume that the average RLMI document size (*R*) can be estimated as $R = C \times r$, where *r* is the average size of the RLMI section devoted to each Presentity. The NOTIFY message contains a full RLMI document (i.e.: with size R) when the Watcher initiates or refreshes a subscription. On the other hand, a *partial* RLMI document (with size *r*) is sent each time a real Presence update occurs due to a specific Presentity modifying its status.

Putting all these elements together, it is possible to derive an expression for the amount of information exchanged over an observation period T_{OBS} between the Watcher and the RLS, when the Presence List subscription mechanism is used:

$$I_{RLS} = \frac{T_{OBS}}{T_S}\left[S + N + C \times (P + r) + 2 \times O\right] + \frac{T_{OBS}}{T_U} C\left[N + P + r + O\right] \qquad (12)$$

The component associated to $\frac{T_{OBS}}{T_S}$ in expressions (1) and (2) is "background traffic" due to Presence subscription activation and successive refreshes (observe that it does not depend on activity from Presentities). On the other hand, the component associated to $\frac{T_{OBS}}{T_U}$ represents the amount of traffic due to notifications carrying "real" new Presence information from Presentities.

5.3.8 Traffic comparison of Presence subscription mechanisms

5.3.8.1 Introduction

As mentioned above, the goal of Presence Lists is to reduce the amount of traffic exchanged, particularly over wireless links [97]. Effectively, at any given time, a Watcher using this mechanism only needs a single SIP dialog to handle Presence information about a potentially large number of Presentities. On the other hand, a Watcher using the individual mechanism will have to establish C dialogs. In expression (1), the amount of traffic required to establish and refresh dialogs has a linear dependency with C, while in expression (2) this dependency applies only to the specific PIDF document with size *P* that is consolidated into a single NOTIFY message. Furthermore, observe that in this expression there is an "extra" payload carried in the SIP NOTIFY

message: the full RLMI document –which is not required in the basic case– is used to update the status of the Presence subscription. The $C \times r$ component in expression (2) models the inclusion of an RLMI document whose size depends on the number of contacts the Watcher is subscribing to.

When it comes to Presence updates due to "real" activity from Presentities (the part associated to $\frac{T_{OBS}}{T_U}$ in (11) and (12)), both expressions are similar, the only difference being the presence of the partial RLMI document in Presence List subscriptions, which are carried in all notifications sent by the RLS in expression (12).

At first sight, it is not trivial to deduct under what conditions expression (12) offers better performance (i.e.: lower traffic exchanged over the wireless link) than expression (11). As we will see, different scenarios may happen as a function of several input parameters.

There are a number of mostly constant parameters in expressions (11) and (12): effectively, the size of SIP messages will largely be determined by the characteristics of the SIP service and clients being used, and will remain relatively static over the whole service life cycle. In practice, the two most relevant parameters when characterizing the service are: a) how many contacts –on average– does a Watcher application have in its Presence-enabled buddy list, and b) how often a Presentity updates its Presence information. Interestingly, the size of Presence documents (P) will end up being of no relevance, since at the end of the day both mechanisms deliver the same amount of Presence information to the Watcher.

5.3.9 Optimal mechanism selection

In this section we compare both mechanisms from the point of view of traffic generation based on how they behave as a function of the number of contacts (C) and the frequency of updates ($1/T_U$). By analyzing expressions (11) and (12) we observe that both equations have a linear dependency with C. Hence, both expressions grow up with constant and –in general– different slopes, so we can assume that there is a value ($C_{THRESHOLD}$) where both expressions cross each other. For $C<C_{THRESHOLD}$ and $C>C_{THRESHOLD}$, either (11) or (12) respectively will be the optimal option that generates less overall Presence traffic, and vice-versa.

Once the above conclusion is clear, we can process and obtain the value of $C_{THRESHOLD}$:

$$C_{THRESHOLD} = \frac{S + N + 2 \times O}{S + N + 2 \times O - r(\frac{T_S}{T_U} + 1)} \quad (13)$$

We can derive two conclusions from expression (13):

- As one could have expected, $C_{THRESHOLD}$ does not depend on P.
- If we make the assumption that S, N, O and r are constant values, then $C_{THRESHOLD}$ basically depends on the ratio between the frequency of updates ($1/T_U$) and the frequency of subscription messages (including initial and refreshing subscriptions ($1/T_S$)).

We will now compare the slope at which Presence traffic grows when using each mechanism. After some calculations we can derive both slopes (m_{BASIC} and m_{RLS}) into expressions (14) and (15):

$$m_{BASIC} = \frac{T_{OBS}}{T_S}[S + N + P + 2 \times O] + \frac{T_{OBS}}{T_U}[N + P + O] \quad (14)$$

$$m_{RLS} = \frac{T_{OBS}}{T_S}[P + r] + \frac{T_{OBS}}{T_U}[N + r + P + O] \quad (15)$$

The interpretation of the above expressions is that as the Watcher adds new contacts to its agenda, the subscription-notification traffic grows up linearly with slopes m_{BASIC} and m_{RLS} respectively, depending on what mechanism the Watcher uses to handle Presence subscriptions.

It is interesting to compare both slopes to determine what mechanism grows at a higher rate. If we compare m_{RLS} and m_{BASIC} and make some processing, we can derive expression (6):

$$\frac{T_S}{T_U} \geq \frac{S + N + 2 \times O}{r} - 1 \quad (16)$$
$$m_{RLS} \geq m_{BASIC}$$

Hence, the amount of traffic generated by the RLS grows up at a higher rate than the basic mechanism when the left side of (16) is higher than the right side. If we assume that S, N, O and r remain generally stable, then m_{RLS} is higher than m_{BASIC} when the relationship between the frequency of subscription refreshes ($1/T_S$) and the frequency of updates ($1/T_U$) goes beyond a certain constant value. This is particularly interesting, because the RLS mechanism was originally conceived as a generic traffic optimization [4], and we have demonstrated that under certain

circumstances the traffic generated by the RLS mechanism grows at a faster rate than the basic mechanism. We now combine both results (the $C_{THRESHOLD}$ value and the slope comparison) to clearly determine when each mechanism should be used to optimize overall traffic using an adaptive strategy. Putting together (13) and (16) we can build up the following table.

Optimal mechanism selection	**Slope Characteristics**	
Number of contacts	$m_{RLS} > m_{BASIC}$	$m_{RLS} < m_{BASIC}$
$C < C_{THRESHOLD}$	RLS	Basic
$C > C_{THRESHOLD}$	Basic	RLS

Table 8. Characterization of RLS and Basic mechanisms (generated traffic).

For a number of contacts below $C_{THRESHOLD}$, the RLS mechanism is optimal as long as equation (6) stands. The RLS is also the preferred choice for a number of contacts above $C_{THRESHOLD}$, when (6) cannot be verified. The basic mechanism generates less traffic in the other two cases.

5.3.10 Numerical evaluation

We have demonstrated that there are circumstances in which the basic mechanism behaves better than RLS. It is now the turn to numerically evaluate expressions presented above with real figures. This will help us assess how relevant the four situations presented in the above table are.

As we have seen in equations (11) to (16) SIP message sizes are a key factor in determining the optimal Presence subscription mechanism. For the rest of this section we will use two different scenarios with different typical SIP message sizes, namely: a) simple SIP messages with a minimum set of mandatory headers [91] [97] [92] vs. b) larger IMS SIP messages [103].

The input values used in the rest of the paper are taken from [92] [84] [103] and summarized in the rightmost columns of table 7.

With these values, we can plot $C_{THRESHOLD}$ as a function of T_S/T_U (Figure 39). This parameter is the relationship between the frequency of updates ($1/T_U$) and the frequency of subscriptions ($1/T_S$). Clearly, longer subscription timers (T_S) reduce signaling load caused by refreshing SUBSCRIBE messages. The default value for the “Expires” header in the Presence event package is 3600 seconds [92]. Consequently, a Watcher using a large T_S value will minimize “background”

Presence traffic due to re-subscriptions. On the other hand, the T_U parameter depends solely on how frequently Presentities modify their Presence status. As an example, a value T_S/T_U of 6 means that, on average, each Presentity updates its Presence information 6 times between two consecutive subscription refreshes issued by the Watcher.

Without loss of generality, we will use a T_S value of 2 hours [103]. Following the above example, observe that a T_S/T_U value of 6 indicates that each Presentity updates its Presence information approximately every 20 minutes.

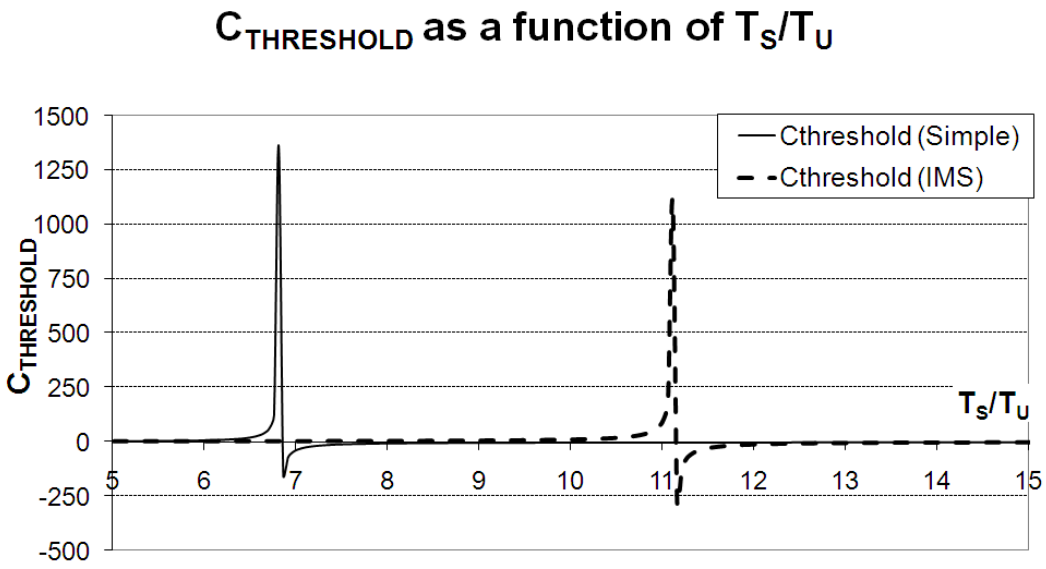

Figure 44. The $C_{THRESHOLD}$ curve as a function of T_S/T_U

There are several interesting conclusions we can derive out of Figure 39: first of all, observe that the $C_{THRESHOLD}$ curve has a discontinuity when the denominator of (13) becomes zero (that is: when both sides of equation (16) are equal): the slope of both mechanisms is the same, so both curves never cross. In such circumstance, the basic mechanism outperforms the RLS mechanism, because its value for C=0 is lower.

Furthermore, numerator in $C_{THRESHOLD}$ is a constant value (assuming approximately constant SIP message sizes), while denominator grows as T_S/T_U grows (i.e.: when activity from Presence sources grows compared to the subscription timer). Observe this effect in Figure 39.

$$\lim_{T_S/T_U \to \infty} (C_{THRESHOLD}) \leq 1 \qquad (17)$$

Combining expressions (16) and (17) we observe that, for T_S/T_U beyond a certain value, the basic mechanism consistently outperforms RLS for any nontrivial number of contacts that the Watcher application has in its buddy list (i.e.: $C \geq 1$). Such value is defined as follows:

$$\sigma \equiv \left.\frac{T_S}{T_U}\right|_{THRESHOLD} = \frac{S + N + 2 \times O}{r} - 1 \qquad (18)$$

We can interpret (18) as follows: when Presence Sources generate a rate of Presence status updates ($1/T_U$) that is equal or higher than σ times the rate of Presence re-subscriptions ($1/T_S$), then the basic Presence subscription mechanism is always the preferred choice in terms of minimal Presence traffic generation.

Interestingly, we can evaluate the value of σ for both the "Simple" and "IMS" message types. Applying the values in table 1 we obtain σ_{SIMPLE}=5.81 and σ_{IMS}=10.11 respectively.

If we take the "Simple" SIP messages case as a reference, the above results effectively mean that if the average ratio of Presence updates generated by each Presentity is at least 5.81 times higher than the ratio of re-subscriptions, the Watcher should use the basic mechanism to subscribe to Presence information about its contacts, instead of the RLS mechanism. As an example, if we take a re-subscription timer of 2 hours, then the basic subscription mechanism becomes the optimal choice if Presence Sources generate new status updates every 20:39 minutes approximately.

On the other hand, if we take the "IMS" case as a reference, the basic mechanism becomes more effective only if Presence Sources generate an update within less than 12 minutes.

Depending on the actual environment or application, having a Presentity send a Presence update with a frequency of less than 20 minutes, may seem a rather high updates frequency. However, there are two aspects we need to consider to put these values into perspective, namely:

- The values calculated for σ_{BASIC} and σ_{IMS} are general values. If each Presentity sends an update with an average frequency σ times higher than the frequency of re-subscriptions, the basic mechanism will perform better *regardless* of the number of contacts the Watcher subscribes to. Hence, even if the Presentity sends updates with lower frequency the basic mechanism may *still* perform better, as long as the number of contacts C is below the $C_{THRESHOLD}$ value that can be calculated using (13).

- The second important aspect is that we have selected a re-subscription timer of 2 hours. It is however largely acknowledged that using longer re-subscription timers represents an important traffic reduction. Effectively, the re-subscription traffic does not bring any extra information to the Watcher application, since in most cases notifications due to re-subscriptions contain basically the same information that was available to the Watcher due to reception of a previous "real" notification.

In particular, if we focus on the second bullet above, observe that as long as we make T_S higher, the first element in (11) tends to zero. We can calculate the σ values with different re-subscription timers. We present such values in table 4, where we compare the Presence re-subscription time with the Presentity minimum activity level that makes the basic subscription mechanism the preferred option for any value of C, in terms of generated Presence traffic.

T_S timer (s)	**Minimum T_U value for $T_U/T_S > \sigma_{SIMPLE}$ (s)**	**Minimum T_U value for $T_U/T_S > \sigma_{IMS}$ (s)**
3600 (1h)	620	316
14400 (4h)	2478	1424
28800 (8h)	4957	2847
43200 (12h)	7435	4273

Table 9. Maximum average time between notifications required for the basic subscription mechanism to outperform RLS subscriptions for all C>0 values.

Hence, observe that if we take a re-subscription timer of 12 hours, the basic mechanism becomes more efficient if every Presentity updates its status information at least once every 2h04' on average. In many applications this value will fall well within typical update frequencies of regular Internet Presence-enabled applications. A Presence update, as an example, may be issued when a user starts a multimedia call, or when she enters a meeting, or goes for lunch, or when her device remains idle for a specified amount of time. Hence, if the average time between consecutive Presence-relevant events is equal to or less than 7435 seconds, selecting the RLS Presence subscription mechanism will *not* be the optimal choice from the traffic savings perspective.

If we take the "IMS" case as a reference and a 12 hours re-subscription timer, the basic mechanism will be the best choice, as long as Presentities update their Presence information at least once every 1h11'.

On the other hand, if a Watcher application uses shorter re-subscription timers (e.g.: one hour) the RLS mechanism will certainly be the best option in most cases: effectively, as long as the average time between Presence updates from each Presentity is longer than 5'16'' (in the "IMS" case) the RLS mechanism will consistently outperform the basic mechanism. In typical Presence usages, it seems quite reasonable to assume that most users update their Presence information less frequently than such value.

In summary, we have proven that within certain frequencies of Presence updates either the basic or the RLS mechanisms may be the optimal choice for all C values.

5.3.11 Advanced Presence optimizations

In addition to the basic and RLS subscription mechanisms, there are several enhancements defined at IETF, 3GPP or OMA, aimed at controlling the rate or generated traffic load of Presence notifications. Usage of such advanced features may have an impact in expressions presented in the previous sections. Now we will briefly outline some of the more relevant optimizations applicable to our discussion:

1. The *Partial Notification* [104] mechanism aims at notifying shorter Presence documents, by delivering to the Watcher not full PIDF documents but XML fragments that contain only the subset of Presence information that has changed since last update.
2. The *Presence SigComp dictionary* [105] incorporates support for compression of PIDF documents within the SigComp framework. The SigComp family of specifications defines a set of mechanisms aimed at reducing SIP signaling overhead. With [105] an extra degree of efficiency can be achieved by using a particularly optimized dictionary to compress not only the bearer SIP messages, but also actual XML content that, in the case of standard Presence documents, contains a significant share of redundant or static information.
3. The *Conditional Event Notification* [106] proposes a reduction in re-subscription traffic, by eliminating the subsequent NOTIFY message that is sent by default after a successful re-SUBSCRIBE. When refreshing a subscription, the new NOTIFY message generally contains Presence information already known to the Watcher. In such case, the

SUSBCRIBE request is answered with a special success code (*204 No Notification*) that indicates there is no need to send a redundant NOTIFY.

4. There are several less mature initiatives aimed at reducing the rate at which Presence notifications are sent to subscribed Watchers. In particular, the *Throttling* mechanism [107] enables Watchers and Notifiers to agree about a maximum (or minimum) rate of notifications. As a particular case, this mechanism also allows Watchers to temporarily pause subscription without terminating them. This lets Watcher clients reduce the traffic overhead during low activity periods (e.g.: device locked). The throttling mechanism reduces Presence traffic, at the expense of losing timeliness of stored Presence information.

Although it is not this paper's aim to be exhaustive about impact of advanced Presence optimizations into expressions (11) to (18), it is possible to perform an initial estimation about expected modifications into equations and results calculated so far. Effectively, if we take *Partial Notification* and the *SigComp* Presence dictionary mentioned above, we observe that in principle both mechanisms should have an impact in reducing the amount of PIDF information delivered to the Watcher application. Hence, we can imagine a new value P' resulting from the application of both mechanisms, with P' < P. However, this has little impact on the obtained results. Effectively, as explained in section 5.3.8, the optimal mechanism selection is independent of P, since all procedures end up delivering the same amount of Presence information to the end user.

Conditional Event Notifications [106] actually remove redundant NOTIFY messages exchanged during the SUBSCRIBE refresh process if no Presence update has occurred since previous NOTIFY messages. We can estimate the new expression for the basic subscription mechanism when [9] applies[11]:

$$I_{BASIC}\big|_{ETAG} = \frac{T_{OBS}}{T_S} C[S+O] + \left[1 + \frac{T_{OBS}}{T_U}\right] C[N+P'+O] \qquad (19)$$

and also in the RLS case:

[11] In these expressions we make the assumption that during the Presence re-subscription process no Presence status update occurs. Instead, all status updates are delivered via asynchronous NOTIFY messages not triggered by the re-subscription process.

$$I_{RLS}\big|_{ETAG} = \frac{T_{OBS}}{T_S}[S+O]+\left[1+\frac{T_{OBS}}{T_U}\right]C[N+P'+r+O] \qquad (20)$$

In both equations (19) and (20) the first component, dependent on T_S represents the "background" traffic due to Presence subscription, while the second part represents the actual notification traffic, due to initial subscription plus average activity from Presentities. Observe that now the component associated to T_{OBS}/T_S does not include the traffic due to redundant Notifications sent when refreshing the subscription, since that redundant information has generally been delivered before as a "real" Presence information update.

In expression (20) we observe that –due to the single RLS subscription– the background re-subscription traffic does not depend on C anymore. This is actually not the case of expression (19), because in the basic case even though NOTIFY messages are eliminated, C SUBSCRIBE messages have to be sent anyway. Hence, with *Conditional Event Notification*s, the RLS procedure may significantly outperform the basic mechanism in a broader range of T_S / T_U cases.

On the other hand, if we understand that having a large subscription timer (T_S value) leads to less Presence traffic, observe that when $T_S \rightarrow \infty$, there is a certain value T_S for which the basic mechanism still behaves better. It is a matter of determining actual T_S / T_U relationship that leads to definition of relevant $C_{THRESHOLD}$ and σ values.

Without getting into more detail about new optimizations still being discussed in IETF and OMA, we have demonstrated that even if complex Presence optimization mechanisms are used, it is possible to derive the corresponding expressions that let a Watcher application estimate which is the Presence subscription mechanism that best optimizes wireless bandwidth usage.

5.3.12 Applicability

The calculations in sections 5.3 demonstrate that it is possible for a Watcher application to estimate the amount of Presence traffic that its subscription mechanism (basic or RLS) may generate, by using a reduced set of parameters which can be measured over the time as the Presence service gets used. As an example, by measuring the number of contacts in the "buddy list", plus average number of updates per Presentity and using real SIP message sizes, all expressions (11) to (18) can be calculated.

A Watcher application can hence decide the optimal subscription mechanism, based on empirical data (its measurements) and analytical models (the expressions proposed by the authors). Furthermore, as traffic patterns evolve or the number of contacts grows, the client may perform proposed calculations in a regular basis, thus dynamically adapting the mechanism selection based on service and usage evolution. Furthermore, the presented results let an Operator minimize overall Presence traffic generated by all subscribers, since at any given time each client would be using the optimal mechanism. Of course, further enhancements can be achieved when Throttling or SigComp dictionaries are used. Regardless, for a given set of mandatory and optional Presence subscriptions features, the calculations discussed in this paper offer a way to decide the subscription procedure that optimizes overall traffic.

As an example, imagine for example a Presence service deployment serving millions of subscribers. Ensuring that each Presence Watcher uses the optimal mechanism at any given time may lead to nontrivial bandwidth savings at the radio, core and service layers.

Another indirect benefit of letting clients decide the best subscription mechanism is that the RLS subscription mechanism is based on a Watcher application retrieving Presence information from a network node (the RLS) located in its home domain. With the proposed calculations, if a client determines that direct subscription to the Presence service is optimal, then the Presence Server that hosts the Presentity (the agenda “contact”) is the one that directly serves Watcher subscriptions. This architectural change may actually lead to more reliable performance, because it eliminates a *single point of failure* (the home RLS) and to more relaxed performance requirements into the RLS.

A third positive outcome of using the optimal mechanism is of course the possibility of using network and battery resources in a more efficient way, which leads to more sustainable and less costly services. Furthermore, when Presence traffic shares radio and network resources with real-time Telephony traffic in the future (e.g.: with large scale LTE radio deployments) the fact that Presence traffic uses a reduced share of radio resources will have a positive impact in telephony signaling traffic not being delayed due to the overhead of inefficient Presence traffic [93].

A fourth benefit of clients automatically selecting the most efficient Presence subscription mechanism is related to charging: for those cases where Presence traffic is charged in a “per

volume" basis, this optimal mechanism selection will obviously represent a benefit to the user. This is of particular interest when users are roaming to a visited network abroad.

Finally, it is worth mentioning that besides the four benefits presented above, the basic SIP subscription mechanism has the drawback that it requires maintaining C parallel SIP dialogs and associated state by the client, as opposed to a single dialog in the RLS case).

5.3.13 Conclusions

Although it is generally assumed that the RLS-based subscription reduces overall Presence traffic exchanged between the Presence service and the Watcher application, an important contribution of section 5.3 is to demonstrate that such assumption cannot be generalized. In fact, depending on traffic patterns, message sizes and number of contacts of the Presence-enabled application, we have demonstrated that the basic Presence subscription mechanism may outperform the RLS subscription in terms of generated traffic. This conclusion stands even if Presence changes occur within less than 2 hours, which seems to be a reasonable Presence update time for many activities (at meeting, at lunch, at home, at cinema, chatting).

The second contribution when compared with previous work in the area is the introduction of analytical expressions that can be used to estimate the total Presence traffic exchanged over the wireless interface. Such expressions take into considerations the different architectural variants introduced by the OMA Presence service [84] and their implications in the user-to-network communication. We can therefore calculate overall estimated traffic, rate of growth of each mechanism, and determination of the optimal mechanism taking traffic measurements as input parameters. Another deliverable is the mathematical expression to calculate the *threshold* ($C_{TRHESHOLD}$) –as a function of the number of *buddies*– at which both subscription mechanisms generate the same traffic, thus easing the calculation of the optimal mechanism based on whether the actual number of contacts (C) is greater or smaller than the threshold.

The presented mathematical expressions can be used by Watchers to determine the optimal Presence subscription mechanism at any given time. Such calculations can be performed in real-time and in an adaptive way, by monitoring only three input parameters: a) SIP message sizes, b) arrival rate of notification messages, and c) number of contacts in the *buddy list*.

It is remarkable that we do not propose a proprietary Presence extension. Rather, we propose to exploit information that is readily available at the Watcher application to select, among existing

standardized solutions, the optimal one in terms of generated wireless traffic. Hence, applicability of the proposals does not depend on new mechanisms having to be incorporated into standardization. Rather, the proposed calculations can be used immediately to optimize applications based on existing standards.

All this work can be extended in the following directions:

1. Evaluation and proposal of new expressions that cover advanced mechanisms ([106] [107]) which may get eventually incorporated into OMA Presence v2 [19].
2. Profiling Presentities based on their level of activity. This work aims at finding whether a combination of basic and RLS mechanism subscriptions for two different subsets of Presentities based on their level of activity may help further optimize overall traffic.
3. Determination of new analytical models based on new scenarios, such as the *un-subscription* effect, the inclusion of frequently changing Presence attributes (e.g.: Location information) or considering traffic models where some correlation exists between Watcher actions and arrival of Presence updates from Presentities.

5.4 Further optimizing Presence subscriptions

In addition to the description provided above, as a consequence of our work around OMA Presence, IMS and SIP/SIMPLE based services, we also developed another enhancement of the service, namely a mechanism to provide a Presence service about the status of a Watcher user.

The idea is that a substantial fraction of Presence notifications delivered to a Watcher may be irrelevant. As an example, if a Presentity modifies its Presence status when a Watcher application is running in a smartphone device that has its screen off and is placed in the pocket of a user, such notification may consume network and radio resources, but may never become useful if the end user does not “see” such Presence information.

The proposed enhancement consists on a Watcher application being able to update the subscription status to the Presence server and inform it when the Watcher is “idle” or “suspended” (e.g.: app in background, screen switched off, …). In such case the Presence Server will keep the “status” of the Watcher subscription, but it will not deliver notifications to the Watcher. Once the Watcher becomes active again (e.g.: the user picks up the phone and switches the screen on) the Presence Server may deliver the updated information. Note that in this way –if subscriptions are suspended but not terminated– the Presence Server will have to deliver only information that has changed since the suspension, not having to restart all subscriptions again.

Note that this optimization is mostly applicable when RLS is used. Otherwise the individual suspension traffic sent to each individual subscription may end up generating a traffic overload, which is precisely the opposite of the goal of the proposed mechanism. Alternatively, another future extension could be the development of a future mechanism based on individual subscriptions (which prove to be more efficient in many circumstances) but place them in the context of a general SIP dialog to enable unified management of such subscriptions. In this case, a single SUBSCRIBE message to put all subscriptions "on hold" (while the device is idle) could be used, thus leveraging the best combination of RLS (single subscription to manage several events) with individual subscriptions (notifications only delivered in relation to each contact, thus avoiding the RLMI overhead). This could represent future research work.

The baseline concept is depicted in the figure below.

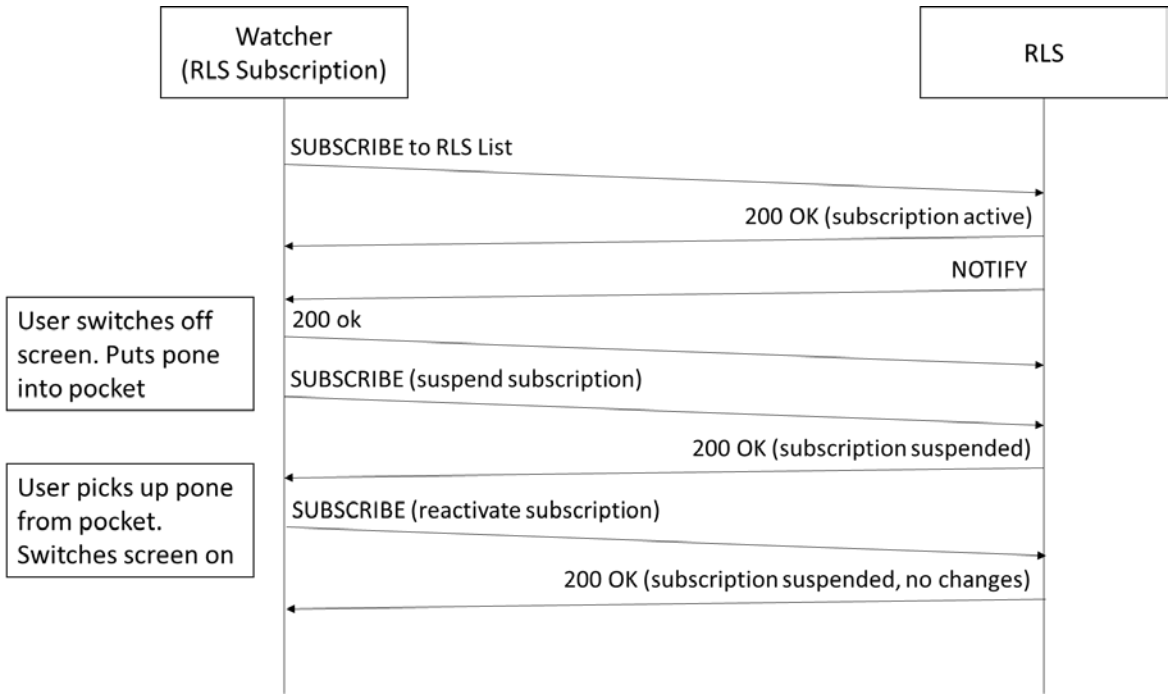

Figure 45. Watcher "idle" subscription concept.

The concept of suspended subscription due to "idle" Watcher is described in US patent US20090319655 [19].

5.5 Contributions into OMA PoC and 3GPP Mission Critical PTT services

5.5.1 Overview of OMA PoC and 3GPP MCPTT intro

The SIMPLE Presence service is largely based on the SIP protocol as used in 3GPP IMS. SIP also supports a number of other real-time communication services as mentioned above, such as VoLTE. In this context, part of our research work also focused on other SIP-based services, with particular interest in Group

Communications. Our approach comes, as usual, with an interest about architecture, signaling protocols and optimizations of standard procedures.

Group Communications over 3GPP networks are sometimes referred, as a technology, in general, as “PoC” (“Push-to-Talk over Cellular”). Indeed, PoC technologies bring the “push-to-talk” (PTT) concept into the mobile domain to networks based on cellular structure (e.g.: 3G, 4G, 5G, …). PTT services have been in use over the last decades, and are the foundation for instant, real-time group communications among teams of users that need to remain coordinated. Traditional non-cellular walkie-talkie technologies have been delivering the “PTT experience” since the first units were created approximately 80 years ago (e.g.: the C-58 “Handy-Talkie”). Walkie-talkie PTT communications are fast, convenient, group-oriented and very integrated into everyday operational duties of teams in multiple fields, from military to air operations, from search&rescue to Oil&Gas, from railway communications to fire fighting.

Over the last 10 years, the interest to run PTT services over cellular networks has grown rapidly. Effectively, as the capacity, coverage, reliability and flexibility of 4G networks has evolved, it has become obvious that cellular systems (which serve thousands of millions of users today) have dramatically better economies of scale than traditional radio systems, with tens of millions of users worldwide [108].

With SIP and RTP being the core protocols of the IMS platform, and the foundations to support VoLTE services, it is no surprise that the Industry selected SIP, RTP and the IMS architecture as the baseline to define a future group oriented communication service to emulate traditional walkie-talkie / PMR (Private Mobile Radio) services.

In the period 2004 – 2011 it was initially the Open Mobile Alliance who took the lead and defined an architecture for PoC / PTT services over cellular networks, standardizing the OMA Push-to-Talk over Cellular enabler, in its 1.x and 2.x releases [79] [80]. The picture below depicts the architecture of the OMA PoC service, when deployed over an IMS network and in combination with other OMA enablers such as Presence or XDMS.

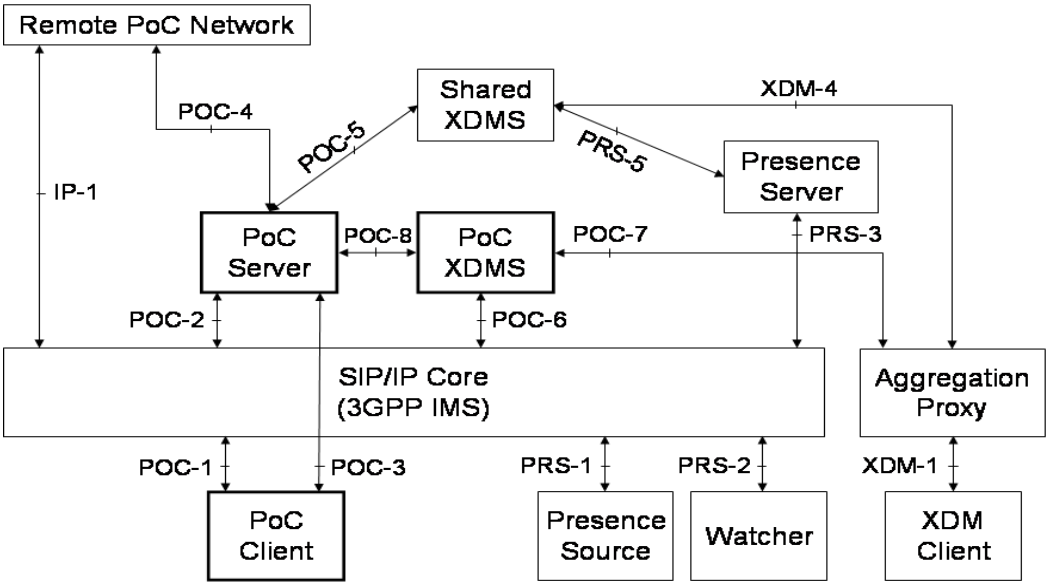

Figure 46. OMA PoC, Presence, XDM, Presence architecture over 3GPP IMS.

While there were a few deployments of OMA PoC 1 and OMA PoC 2 solutions worldwide, lack of fit for purpose devices in the key industries, as well as lack of consistent infrastructure support (first OMA PoC launches date back to 2004, when not even 3G was still widely available) did not enable instant success of the service. However, the architecture, protocol suite and key concepts proved invaluable for future developments.

As a consequence of author's involvement and research in this field crystalized in the form of the publication of the first and single book reference that covers OMA Push-to-Talk, Presence and XDMS services in deep detail, together with Nokia's Andrew Rebeiro-Hargrave [109].

As mentioned, as several industry trends converged, the importance of the OMA PoC foundation increased steadily. Effectively, massive deployments of 4G LTE networks[12], appearance of open Operating Systems, which in turn boosted creativity and availability of new types of devices both for consumers as well as for vertical market segments, and the slow adoption of IMS and VoLTE, coupled with traditional PMR technologies slowly reaching end of life, represented a renewed interest in PoC / PTT over cellular technologies.

[12] In particular, with the more recent allocation of LTE-enabled bands in the range 450MHz – 800MH such as bands 14, 20 or 31, with particularly good coverage characteristics in rural areas, not that far away from the coverage that can be reached by PMR systems in the UHF band.

In particular, with several Public Safety agencies having an urgency to enhance or renew their traditional PMR networks, 3GPP embarked in the creation of a new IMS-based standard to support Mission Critical Push-to-Talk communications over 4G and future NG networks. Due to the importance of this activity from a Public Safety perspective and the strong involvement by national agencies in several 3GPP working groups, an exception was made and this new enabler would be led by 3GPP, instead of OMA. However, both from an architectural, protocol and design perspective, the 3GPP MCPTT activity strongly inherited the developments by OMA in PoC 1 and –particularly– PoC 2 standards.

With all these ideas in mind, and after completing [109] and [19], we engaged into further investigating the possibilities and architectures enabling 3GPP Mission Critical Communications over LTE/4G. Such journey materialized in the paper entitled "A Distributed Man-Machine Dispatching Architecture for Emergency Operations based on 3GPP Mission Critical services". In the following sections we will provide a high level overview of 3GPP MCPTT, together with related research work by other authors, as well as a brief description of our main contribution to the area [17].

5.6 A distributed "bot" Dispatching Architecture for Emergency Operations based on 3GPP Mission Critical Communication Services

5.6.1 Introduction and related work

The final author's contribution in the scope of this Thesis Work was IEEE Access' published paper "A Distributed Man Machine Dispatching Architecture for Emergency Operations based on 3GPP Mission Critical Services" [17]. In this paper we leveraged the guiding principles shared across the rest of the work ("do more with less", "keep it simple", "keep interoperability principles") and applied them to the Mission Critical Communications service MCPTT. In this case, rather than discussing detailed protocol implementation aspects, we focused our work on architectural deployment aspects. Anyhow, the architecture of the MCPTT service and the involved protocols (SIP, RTP) retain a close linkage to the rest of topics presented in previous chapters.

In the rest of section 5.6 we provide a high level overview of [17]. Some of the main contributions of the paper include:

- Presenting the Dispatching concepts in the context of 3GPP MCPTT / MCC services.
- Introducing the MCC "bot" concept, as a way to automate and enhance the decision-making process to support on-the-field operational efficiency.
- Outline how MCC "bots" can bring value to the traditional top-down hierarchical dispatching paradigm in mission critical communications.

- Describe different alternatives for architectural implementation of MC "bots" in function of complexity level and intended operational use, including usage of UAVs (e.g.: drones) as MC "bots".

In this context, we provide first a quick overview of related work by other authors in the context of 3GPP Mission Critical communications, as well as research work around Critical Communications dispatching.

Firstly, in the field of LTE for Mission Critical / Public Safety operations, several authors have already described some of the gaps and required evolution for 3GPP networks to support Mission Critical communications. Kumbhar et.al. [110] discusses the validity of LMR and LTE coexistence splitting voice and data delivery over separated complementary systems. Simic [111] also claims that while LTE is the current natural choice for Public Safety broadband data communications, existing LMR systems may still carry Public Safety voice services for a relevant time until 3GPP networks deliver the required features in terms of preemption, prioritization and QoS. Other authors highlight that hybrid public / dedicated LTE networks may be deployed to fully cope with voice and data services for Public Safety, as described in [112]. If we assume that PTT over LTE will eventually replace traditional LME systems, several authors have evaluated the performance of such PTT services over LTE [113] [114].

In relation to the "dispatching" concept, there are research contributions in the areas of Control Rooms for MC operations [115] [116], drone-assisted MC operations [117] [118] [119] and involvement of autonomous robots in MC operations [120] [121], respectively.

When it comes to research around Control Rooms operations specifically, we acknowledge that significant work has been devoted into areas such as the optimization of the decision-making process in traditional top-down Command and Control systems for Public Safety and Fire Fighting [115] or the optimization of the design of the Command and Control infrastructure itself [116], where the community actually acknowledges that excess of information processing by Control Room operators may degrade operational efficiency [122]. However, little discussion exists about the possibility to mitigate the issue through process automation or distributing the decision making process.

In relation to drone-assisted MC operations, the community has invested significant effort into defining the network-layer interaction, both from a device-to-device (D2D) [123] and Isolated E-UTRAN Operation for Public Safety (IOPS) [124] perspective where such drones mostly operate as coverage "extenders" [117] or local connectivity providers [118] [119] among First Responders. However, little contribution has been made considering such drones capable of interacting with First Responders from an operational perspective.

Finally, [120] [121] cover the involvement of mostly automated or remotely managed robots in Mission Critical operations with little interaction or coordination with the rest of human First Responders in the field.

Actually, it has been proven that even in cases where UAVs (Unmanned Air Vehicles) are deployed to capture on-the-field information, significant improvements are required in the decision-making process to reduce the time required to spread information and to perform decisions based on such information [125].

5.6.2 Introduction to 3GPP Mission Critical Communication Services Architecture

Mission Critical (MC) operations in the civil domain comprise activities handled by so-called "First Responders" such as police officers, fire fighters, Search and Rescue, Medical Emergency support personnel, Civil Protection, … Not all activities carried out by such professionals imply Mission Critical nature, but under certain circumstances these professionals participate in activities where either citizens' life or emergency personnel's life is at stake.

MC operations heavily relay on Mission Critical Communications (MCC) systems. While public cellular mobile networks are intended to serve massive voice and data services with carrier-grade reliability, traditional MCC systems typically consist on highly redundant dedicated network infrastructure leveraging spectrum frequency bands specifically allocated for MCC. Importantly, some of the key requirements by First Responders for their communication systems include high reliability, instant delay (typical mouth-to-ear delay must not exceed 300ms) [126], group communications (required for emergency teams coordination) and strong support for Dispatch-managed operations. As an example, traditional MCC systems offer a set of interfaces to connect external Control Room or Dispatch Centers.

Over the last five years, 3GPP has been working in defining a new service enabler, strongly leveraging OMA PoC [79] [80] [109] standards. The new so-called 3GPP MCPTT solution is expected to enable the transition from traditional radio solutions currently in use by First Responders, such as TETRA [127] or P25 [128].

After the approval of MCPTT, such work is being expanded with new MCC work items devoted to enhance MCPTT and the definition of the Mission Critical Data (MCData) [129] and Mission Critical Video (MCVideo) enablers, standardized in Release-14 and Release-15 [130].

The following diagram provides a high level overview of the 3GPP MCC services architecture with focus on the MCPTT service.

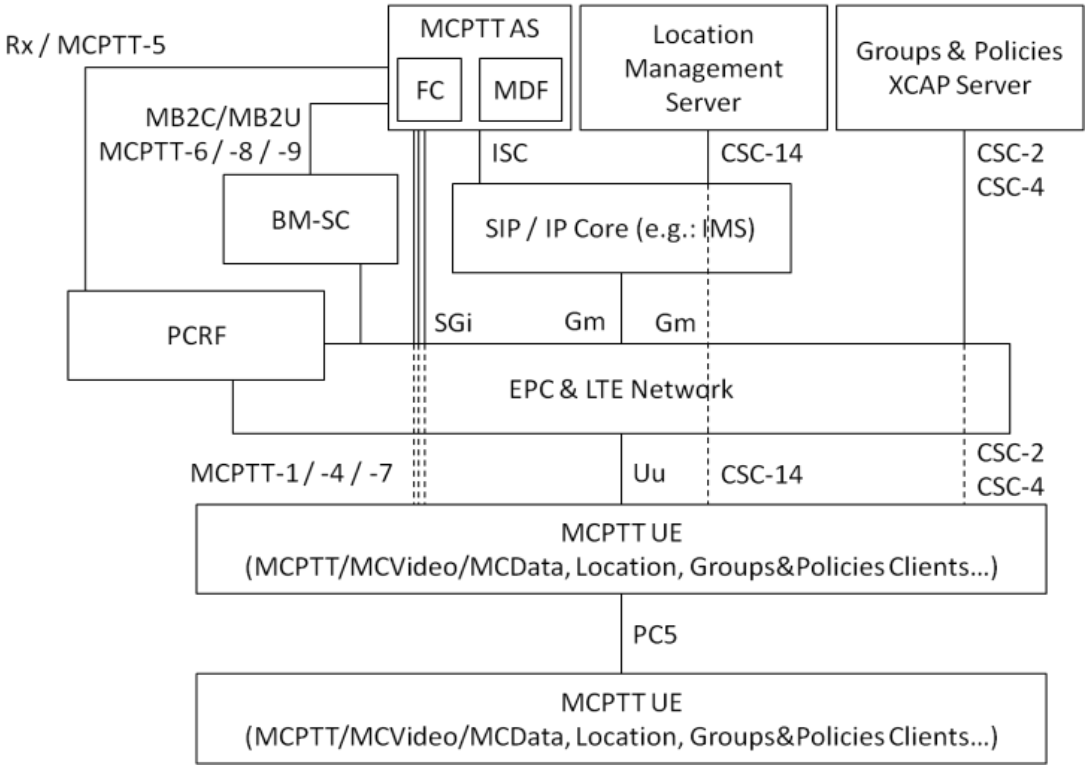

Figure 47. High Level 3GPP MCC architecture.

Note that MCPTT comprises a set of architectural concepts from core Long-Term Evolution (LTE) / Evolved Packet Core (EPC) [61], MCPTT [131], 3GPP Proximity-based services (ProSe) [132] as well as the Common Services Core (CSC) for MCC services [133]. We briefly outline some of the key concepts from the above figure:

- The MCPTT Application Server (MCPTT AS) is in charge of delivering a walkie-talkie-like half duplex group voice communication service, and comprises two important functions: Floor Control and Media Distribution Function. In addition, it contains all the service logic and acts as a Back-to-Back User Agent (B2BUA) from the signaling perspective.

- The MCPTT AS will interface the Policy and Charging Rules Function (PCRF) through the DIAMETER-based Rx interface. This interface lets the MCPTT AS push policies and inform the network about bandwidth/delay/loss tolerance and emergency status of a given media stream. The PCRF will use this information to determine flow priority in the EPC/LTE network.

- The EPC and LTE comprise the 4G core and Radio Access Network (RAN) respectively.

- The Broadcast/Multicast Service Center (BM-SC) allows the MCPTT AS to stream media through multicast bearers over the LTE RAN. This mechanism represents the combination of the

MCPTT and the evolved Multimedia Broadcast and Multicast Service (eMBMS), which can help greatly enhance service performance and avoid congestion.

- The Location Management Server (LMS) and the Groups & Policies Server provide service to all MCC services (PTT, Video, Data). The LMS tracks location of all users while the Groups & Policies server stores XML Configuration Access Protocol (XCAP) documents which are used to store group and policies information for MCC services. Other common services such as Identity Management Server or Key Management Server are not displayed for the sake of simplicity.

In general the above architecture can be generalized to Mission Critical Data and Mission Critical Video by replacing the MCPTT AS with the corresponding MCVideo/MCData AS [129] [130] respectively.

Finally, and importantly for the sake of our discussion, MCPTT, MCData and MCVideo also incorporate a D2D interface (PC5) that allows User Equipment (UE) to establish direct UE-to-UE connectivity without the need of LTE RAN coverage. This type of interface is inherited as a requirement from legacy MCC systems such as TETRA or P25, and intensively used by certain types of First Responders [123].

For the sake of our discussion, an MCPTT Dispatch "bot" or the MCPTT Dispatcher operating from a control room are a special type of MCPTT UE.

5.6.3 Dispatching concepts in the context of Mission Critical Operations

A Control Room environment typically consists of one or more Dispatch operators who may receive information from a number of different sources (e.g,: emergency communications, 911/112 incoming calls, location devices, video feeds from surveillance cameras, …). In turn, Dispatch operators make decisions based on available information, operational procedures and experience. Such decisions are communicated in the form of voice commands toward First Responders instructing, advising and providing information to coordinate how individuals and teams work together to handle an emergency situation in the field.

In general, in order to achieve the best possible coordination of emergency teams, MCC typically follow a top-down hierarchical approach where the First Responder acts under the command of the Dispatch operator who has a general perspective of the operational situation.

Over the last 15-20 years two main architectures have been deployed worldwide to support MCC, namely the ETSI TErrestrial Trunked RAdio (TETRA) [127] system and the US APCO Project 25 (P25) system [128], under the broad family of Professional Mobile Radio (PMR) technologies.

The communication paradigm between the First Responder and the Dispatch Operator in traditional MCC systems is depicted at a high level in Figure 1. Essentially, the First Responder will Detect, Report and Inform, while the Dispatcher will Instruct, Command and Coordinate.

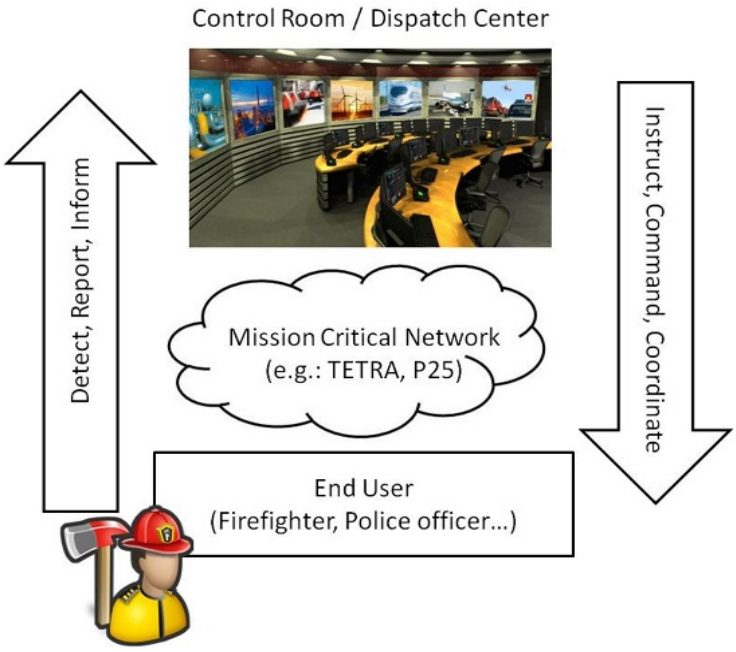

Figure 48. Top-down dispatching in Mission Critical operations.

The novelty of our approach comes by combining three elements (semi-automated "bots", distributed Dispatch functions and 3GPP MCC technologies) to deliver a new paradigm. Effectively, the MC Dispatch "bot" expands the "Control Room" concept among a set of human and robot entities that collaborate together to increase efficiency of MC operations and/or safety of First Responders. In such framework, such "bots" may not operate in a fully automated way but in a collaborative way with First Responders in-the-field. Furthermore, the "bot" concept that we propose does not restrict its area of activity as coverage extender/connectivity provider, rather in our proposal the "bots" ability to gather and process information enable them to deliver useful, timely information to First Responders through the most convenient and instant communication means available: voice.

In essence, in our approach the assumption is made that in the foreseeable future a significant fraction of MC operations will require on-the-field human intervention. We argue that First Responders may benefit from enriching traditional top-down Control Center coordination with MC-enabled "bots" in the form of drones or robots –that work in close coordination of humans– deployed to support and increase coordination, situational awareness and safety of First Responder crews.

In this paper we present a novel distributed man-machine Dispatching architecture for emergency operations based on the combination of 3GPP MCC services and "bots" capable of interacting with First Responders through man-machine communications.

5.6.4 Distributing the Control Room Dispatch Function in the 3GPP MCC Framework

Importantly, MCPTT not only will allow for the migration of the traditional voice-centric service delivered by P25 and TETRA systems (instant, team-focused, Dispatch-managed voice communications). Beyond the pure voice service, 3GPP MCC will enable a broad new range of services, which now become possible by leveraging IP technologies and mobile broadband capabilities that are not feasible in traditional narrowband PMR technologies.

Among other features, capabilities that will become available to First Responders include sending and receiving video feeds, remotely managing and exchanging information with LTE drones or "bots", remotely accessing Public Safety databases, triggering alerts, receiving situational awareness information from multiple sources and sharing, updating or displaying location information about users and devices.

In such framework the amount of relevant information available in real-time to multiple stakeholders (end users, support personnel, Dispatch users, supervisors…) will grow dramatically. Such information (e.g.: where are the team members, what are they seeing, how temperature is increasing around a firefighters' team, what is the heart rate of the police officer, …) may be used to greatly enhance the decision-making process in the scope of emergency situations.

On the other hand, such substantial increase in information availability will surely lead to a dramatic shift in how emergency information is processed, managed, and acted upon. Effectively, small data and big data processing capabilities will be key in enhancing the human decision-making process based on rich, reliable and pre-processed information.

In this environment the authors envisage also a shift into how the traditional command-response, hierarchical, human-to-human interaction between First Responders and Dispatchers will evolve. Effectively, the current paper proposes that the traditional Dispatcher role may be split among a number of human and non-human entities that will spread and collaborate seamlessly through the whole MCC system.

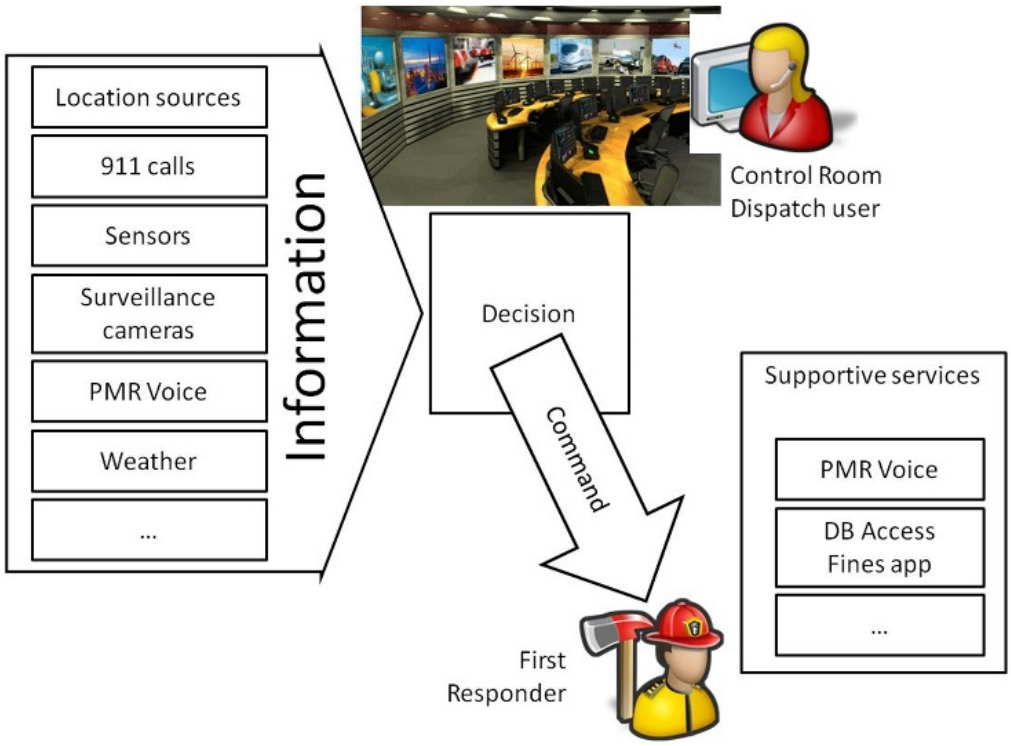

Figure 49. Decision-making process in hierarchical dispatching.

In the future, human Dispatch users will benefit from the collaboration with one or more non-human Dispatch "bots". Dispatch "bots" can act upon received and processed data. Furthermore, for certain types of interactions, "bots" can even proactively coordinate First Responders without the need of an explicit human decision by the Dispatcher in the Control Room environment. Of course, such non-human actions will only be issued in specific cases when the non-human Dispatch "bot" has been pre-programmed to perform automated decisions that are safe to be handled without direct human intervention.

The above figure presents the architecture that depicts the traditional Dispatch-managed paradigm in a greater degree of detail.

As depicted above, Traditional Dispatching has been heavily based on three main pillars, namely: different types of voice communications (including TETRA and P25 systems), some location capabilities displayed on a Geographic Information System (GIS) map and, potentially, some video integration capability with surveillance cameras. All this information is displayed in large control room environments with dozens of large displays and dozens of Dispatch operators who try to make the most, through empirically developed processes, of the available information in tutoring and helping personnel in the field.

In a traditional MC scenario all events and information must be received by the Dispatcher, understood and voiced back to first responder teams. This approach has several drawbacks:

- The central human Dispatch operator becomes a bottleneck of the operation, which may impact speed of reaction, safety and effectiveness of the whole team [122].

- Relative importance and relevance of each individual information item is subject to judgement by the central Dispatch operator.

- Having to cope with specific items affecting just one or a very few users involved in an operation may unnecessarily occupy resources that might be useful for the coordination of the whole team or operation.

It must be noted that the above conclusions should be weighted with the fact that in large operations and complex environments, the "command chain" concept already provides a certain degree of distribution and shared responsibility among the different managers involved in the operation. Effectively, tactical control centers, on-the-field commanders, team leaders, supervisors and observers may help in implementing a relatively coordinated decision-making process where not all actions depend on one and only one Dispatch operator. However, the process is generally strictly hierarchical, and it has already been proven that as complexity increases in terms of the amount of information that a human has to process, effectiveness of the coordination activity decreases [122] and the convenience to automate certain tasks increases [134].

When the MC Dispatch "bot" concept is brought into the picture, the above situation can be enhanced significantly. Effectively, the Dispatch "bot" could work in close coordination with the isolated first responder it is providing support to. The "bot" can be pre-programmed to gather, consume and process certain types of information and trigger events based on such information.

Importantly, it is quite common that first responders are involved in tasks that require full attention, hands free operation and physical activity. This means that, as opposed to many everyday situations by other types of users, audio communications become fundamental to support first responders' activity. Effectively, a fire fighter, a police or paramedical officer may not be able to look at a smartphone screen or computer monitor in the middle of the heat of a real emergency operation. This is when good old, reliable, instant voice, walkie-talkie-like really can become the difference between success and a dangerous situation.

In such environment, being able to distill the "core" of useful and valuable information "pills", based on real facts, to a first responder can help save lives, speed resolution times and improve overall efficiency. In this framework, MC Dispatch "bots" come into play.

Figure below depicts the Dispatch "bot" concept, where the Dispatch function becomes distributed between human and non-human actors, which may provide assistance, instructions, informational awareness and commands.

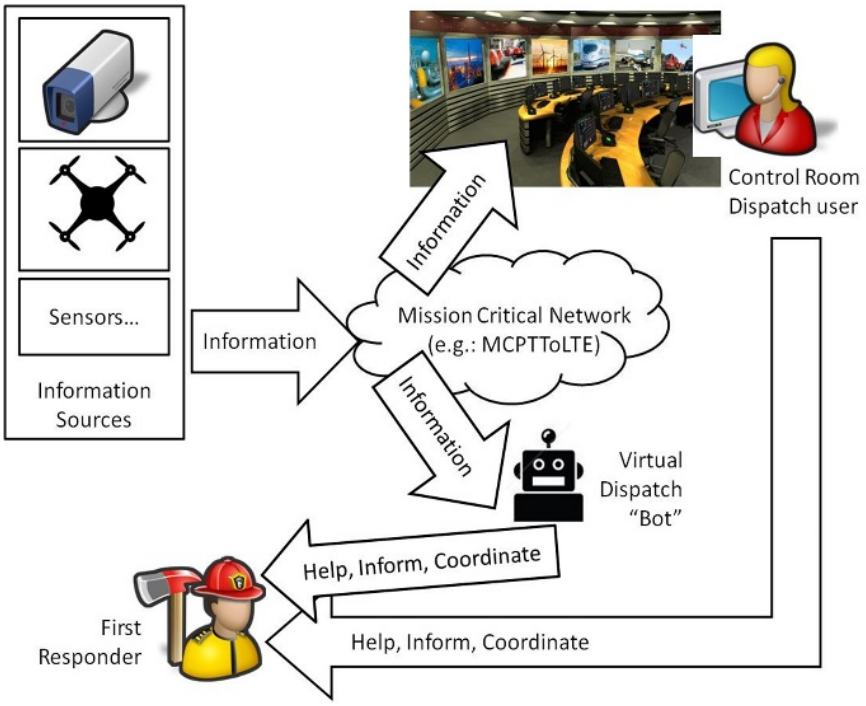

Figure 50. The MC Dispatch "bot" concept.

The virtual Dispatch "bot" represents the fact that it is today possible to gather significant amounts of information, process it and deliver to human recipients in a human-understandable format that prevents the end user from having to grasp raw data or invest significant time in simply understanding such amount of non-processed, disperse, unconnected data from different sources.

The virtual Dispatch "bot" is meant to work in coordination with the end user (the First Responder) as well as the human central Dispatch user. With such approach the Dispatch concept becomes "distributed", always under the command and supervision of the central human Dispatch user.

The benefits of this approach include:

- Reduce decision times, particularly in cases when certain decisions are obvious from an operational perspective but the First Responder involved may not have the perspective to take it and the Control Room Dispatch operator may not have the information required to make it.

- Ensure that all key information required to work effectively is available and presented in the right format at the right time to First Responders.

- Increase real as well as perceived safety by First Responders in the field.

- Ensure that critical information is delivered continuously toward the Critical Control Room Dispatcher.

The MC Dispatch "bot" concept is developed forward in [17], where the following areas are explored:

a) A sample high level architecture of the MC Dispatch "bot" concept and the data collection functionality is described,

b) Discussion about the type of information collected and processed by the Dispatch "bot" is presented,

c) In what format the Dispatch "bot" may deliver the information and be helpful to the First Responder, and

d) How the Dispatch "bot" concept may fit into the MCPTT architecture.

We will focus on d) above and we discuss a sample application scenario. The reader is referred to [17] for further details.

5.6.5 Deploying Dispatch "bots" in a 3GPP MCC Architecture

In this section we review the architecture of the Dispatch "bot" concept from the 3GPP MCC architectural perspective. We have split the evaluation into four layers, as described below.

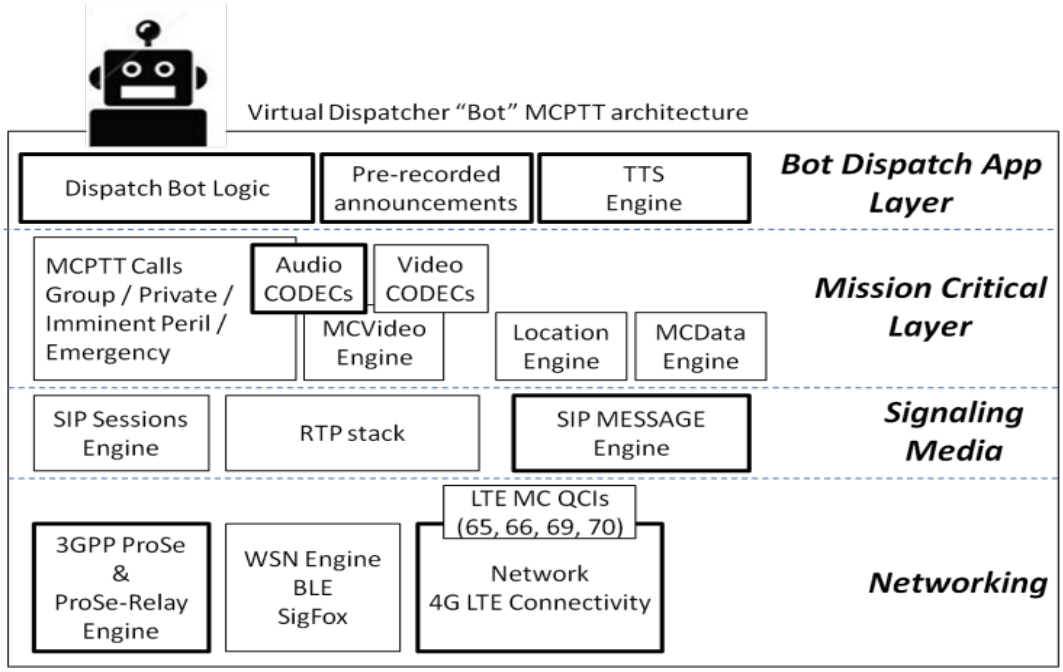

Figure 51. The MC "bot" in the context of the 3GPP MCC architectural framework.

The application layer contains all the necessary logic as well as the elements required for the man-machine interaction, which may include a set of pre-recorded announcements or a text-to-speech engine that can be plugged into MC communications with the end user or the team the "bot" is providing support to.

On the MC layer three main functions may be present:

a) An MCData engine can be used to exchange text, charts or pictures with an MCData application at the UE of one or more First Responders. As an example, data retrieved (e.g.: sensed) from the environment can be pushed to the relevant UEs interested in receiving such information. This delivery mechanism can be used mostly for non-intrusive non-urgent interaction, by providing information that complements the core operation of the team.

b) An MCPTT engine. This can be used to deliver instant, real-time information to a user or a team. It may be based on triggers (time, location, sensed information) and it may be comprised of a combination of pre-recorded announcements and/or text-to-speech composed audio. MCPTT media can be played in the speaker of the user's device. Importantly, when audio information is delivered over MCPTT the listening user can freely use his hands without the need to pick up the device (as opposed to most MCVideo or MCData interactions).

c) An MCVideo engine. Additionally, the "bot" may decide to stream video to a group or to the Dispatch Center in order to provide enhanced situational awareness (e.g.: a flying drone may stream video to a team of firefighters).

The signaling/media layer generally comprises the Real-time Transport Protocol (RTP) used to carry encoded audio or video related to MCPTT and MCVideo sessions. The Session Initiation Protocol (SIP) is used as the main signaling protocol by 3GPP Mission Critical services.

Note that, depending on the purpose, budget and capabilities, a given "bot" implementation may comprise one or more MC services, namely MCPTT, MCData or MCVideo, two, or all of them.

As a specific case of the MCData scenario the "bot" may include a location module capable of reporting location to one or more users as well as to the central Control Room.

Interestingly, many MCData and Location use cases can be implemented on top of the simple SIP MESSAGE transaction. Effectively, delivery of text messages, location coordinates (encoded in XML payload), file sharing URLs or status reports can easily be carried over the atomic SIP MESSAGE transaction based on MCData. This opens up the opportunity for developing a simplified "bot" without audio or video capabilities and running a trimmed down SIP stack, but able to report significant information both from Control Center to First Responders and reverse. Some of the functions required to implement a "simplified" bot are depicted in bold boxes in the figure above.

Note that the "bot" may have different types of form factors. From bodyworn equipment that connects to the UE to a specific SW or HW module attached to the device. It is of particular interest the case of Unmanned

Aerial Vehicles (UAV) that may have simultaneous visibility of the end user(s) they are supporting to as well as connectivity to the core LTE network at the same time. When considering the MC Dispatch bot concept from a 3GPP MCC perspective we can present different example configurations as depicted below.

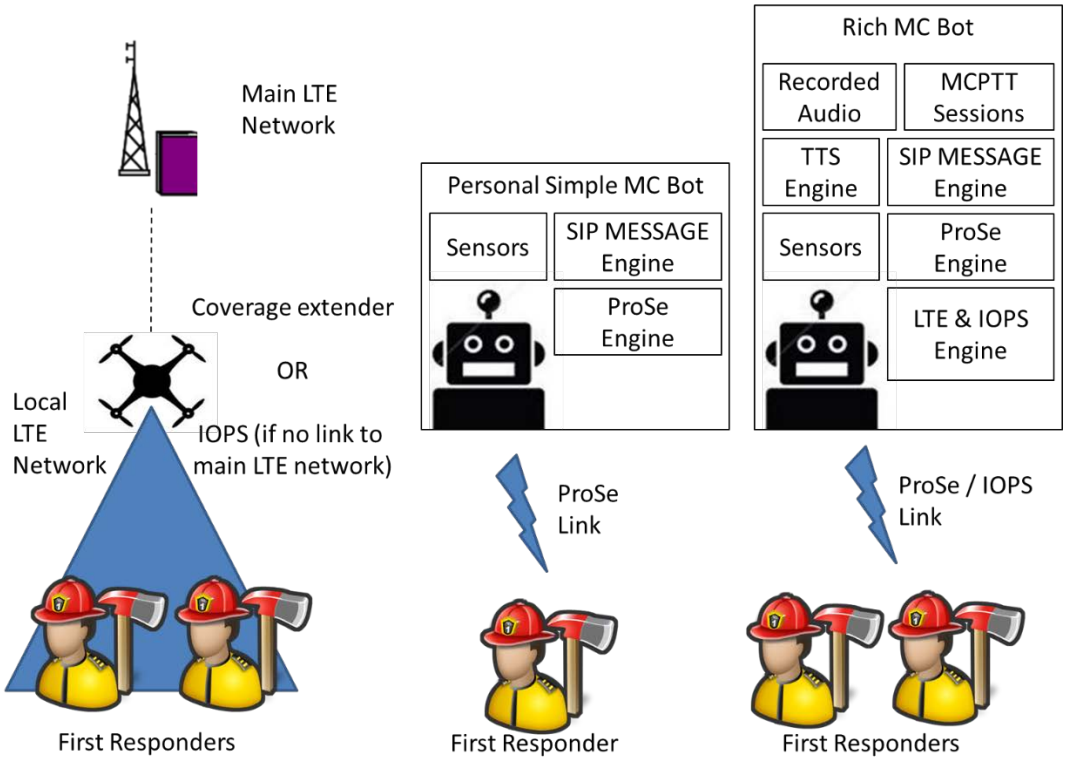

Figure 52. Example MC Dispatch "bot" configurations rom 3GPP MCC perspective.

a) A first configuration covered elsewhere is usage of UAVs as LTE network coverage extenders or as providers of group communications to isolated teams [124] [117] [118] [119]. In this regard it is of particular interest the case in which the "bot" can offer a hotspot of LTE coverage in a remote area, thus potentially serving a group of MCPTT users and allowing them to communicate efficiently regardless of the connectivity to the central network. 3GPP has standardized this scenario under the so-called Isolated Operations for Public Safety (IOPS), in which essentially a node (e.g.: the "bot" itself) can implement the whole RAN+EPC function and deliver one or more MC services in a local area [124]. We do not consider these as MC Dispatch "bots" as described in this paper because there is no application level logic and no reacting based on sensor information, but it is a powerful baseline scenario on top of which additional MC Dispatch "bot" functionality can be added.

b) Informative MC Dispatch "bot". This could consist on a simple "bot" with three main capabilities a) Sensors, b) A MCData SIP MESSAGE engine, and c) Connectivity to the end user UE (e.g.: Bluetooth, 3GPP ProSe, …). This bot provides supportive information that is sent to the UE and displayed to the user. It does not provide coverage extension and it does not send TTS audio messages

(even though information sent over MCData could be played back locally by the UE through its own TTS engine).

c) Rich MC Dispatch "bot". In this case, in addition to the coverage / relay services mentioned in a), the bot may provide information to one or more First Responders by using its TTS engine connected to an MCPTT group communication. This group communication can be delivered to a team of First Responders in the field through a ProSe bearer.

Note that there may be valid use cases in which even if a UAV acting as a MC Dispatch "bot" keeps a ProSe link toward First Responders, it keeps an independent LTE link to the core Control Room, without acting as a relay. With this approach the "bot" may provide to the Control Room some critical information (e.g.: location of the team to keep them safe) while avoiding congestion on the LTE uplink (if it would relay all ProSe MCPTT communications) and saving its own limited battery resources.

5.6.6 Example application scenarios

The high level concepts described above can be better visualized with the presentation of their application into a real life Mission Critical scenario. We will focus on a firefighting environment, while [17] presents further material.

In this case the scenario is as follows: a fire fighter is operating in the forest. He carries a belt that senses a number of human parameters such as blood pressure, skin temperature, heart rate, … As an isolated fire fighter he is supported by a drone that measures air temperature in the vicinity of the fire, concentration of dangerous gases (e.g.: NO, CO, CO2, ...). The advanced Command Center has pointed a Global Positioning System (GPS) coordinates where the next water delivery by a water tanker plane will be performed. The Estimated Time of Arrival (ETA) of the water tanker plane is 9 minutes from now. The fire fighter is 300 meters from the delivery point and 400 meters from the closest and safest team.

We also assume that a supportive drone carries a small "bot" application with LTE connectivity that links him both to the fire fighter as well as the central Command Center. The drone detects that ambient temperature and gas concentration has reached high levels. Fire fighter skin temperature and heart rate indicate that the user is tired and quite close to the area where the next air water drop will happen. The Dispatch "bot" app sends a voice warning message to the fire fighter "*Next water drop is expected in 9 minutes. You are in an unsafe area. The closest support team is 400 meters to the East. You should depart now*". This message is shared instantly through the radio loudspeaker of the fire fighter as well as in a monitoring room back in the central Control Room. Support personnel supervise this message to confirm that this is a safe decision. The fire fighter confirms this order and safely joins his colleagues in time to watch how the air water drop significantly lowers the virulence of the forest fire.

Figure below highlights a more detailed scenario of the fire fighter MCPTT "bot" flow.

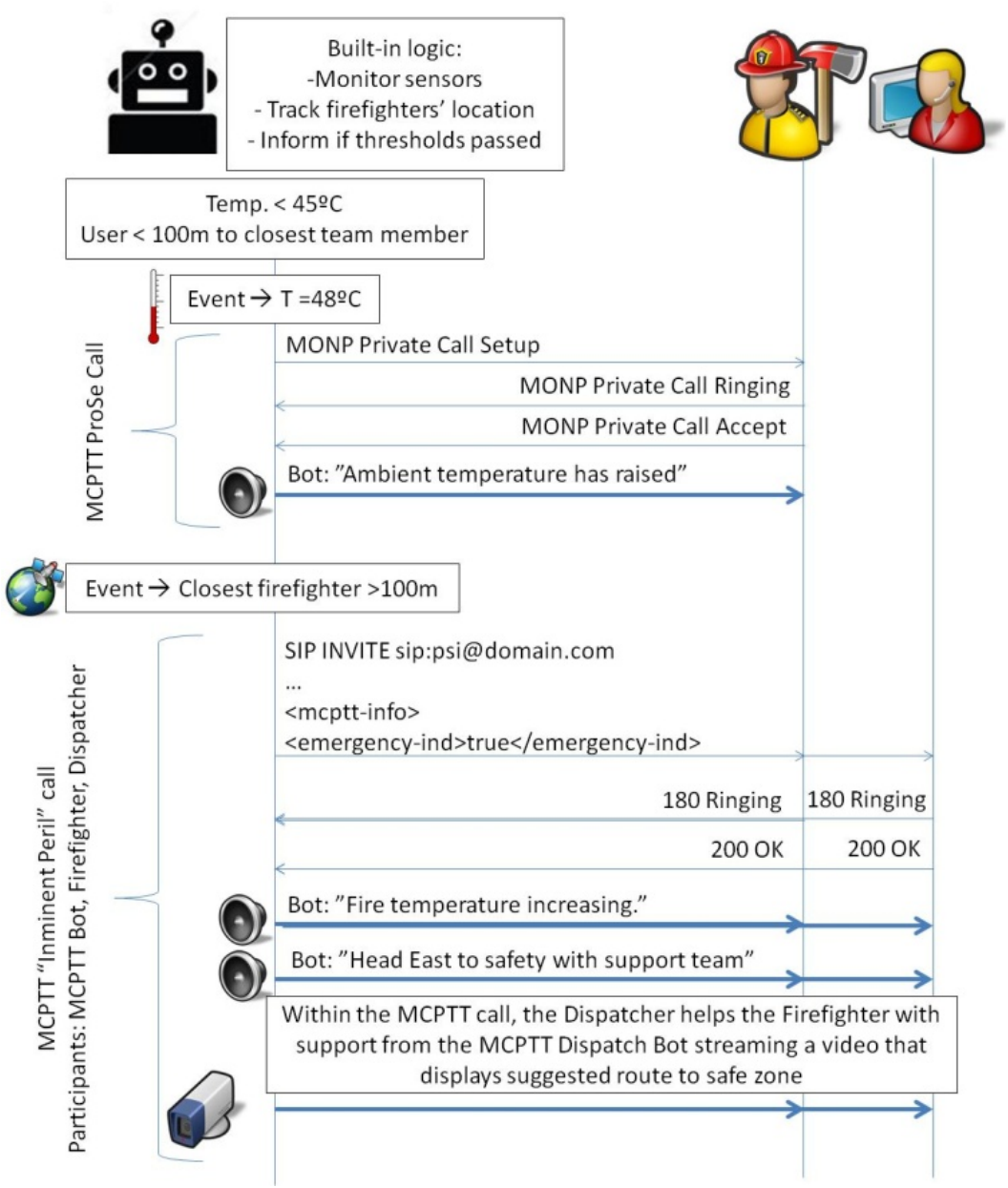

Figure 53. Example Dispatch "bot" firefighter scenario.

Note some of the features highlighted in the diagram:

- The "bot" may participate in different types of MCPTT sessions with different priority levels depending on the trigger point.

- The first flow depicts a direct mode call when the bot warns the firefighter that ambient temperature is rising. In this case the MCPTT Off-Network Protocol (MONP) [131] is used to setup a direct call over the 3GPP Proximity Services (ProSe) bearer [132]. This allows the "bot" and the user

being supported to stay permanently in touch without occupying network and channel resources for the rest of the brigade.

- Upon crossing certain threshold, the "bot" decides to start a new MCPTT "imminent peril" call involving the Dispatch Control Room operator. This is a group call run over the LTE network (since Dispatch-firefighter-bot communication is required). In particular, as highlighted in the flow, the "bot" sets the XML <emergency-indicator> element to "true" in the body of the SIP INVITE message to highlight that potential death or injury is possible.

  This will naturally increase the level of alert as these calls are managed with special priority by the network and the MCPTT system, as well as displayed with additional visual and acoustic alerts in the control room. Imminent Peril calls are defined by 3GPP MCPTT as those were an emergency has not declared but it is likely and immediate unless urgent action is carried out [126].

- Eventually, the combination of the "bot" capabilities and the implementation of bot-assisted operational procedures may help bring the firefighter to the safe zone in a quicker and safer way than if pure human hierarchical communication would have been in place.

In a nutshell, the scenario depicted above, as well as others described in [17] showcase how the MC Dispatcher "bots" can improve safety of First Responders and automate some on-the-field decisions to improve efficiency of MC operations. In particular, we can list some of the features and benefits of this approach:

- "bots" can be pre-loaded with programs adapted to the type of operations (e.g.: fire brigades, police operation, covert ops, counter terrorist, natural disaster, …).

- By leveraging 3GPP ProSe, Mission Critical "bots" would almost never lose connectivity to the end user they are providing support to. Hence, they could provide supportive assistance, information and safety even in cases when the end user has lost connectivity to the main site. Additionally, in certain circumstances (e.g.: UAVs used as "bots") the "bot" itself can have connectivity to the central site, thus enhancing the safety of the end user.

- "Bots" could be programmed with waypoints, trigger points, pre-recorded messages and actions that can easily be converted into audio messages voiced to end users -in some cases, using Text-to-speech (TTS) technology. Hence, the "bot" would communicate to the end user through the most convenient, least intrusive, most instant and most natural interaction mechanism, expected and widely used by emergency users.

- All “bot” communications and actions could be monitored from a back office environment in order to track the appropriateness of “bot” decisions, re-program or disable them in real-time (e.g.: in case that a wrong decision has been made) and develop a self-trained, continuous improvement procedure to enhance the behavior in future operations.

- A set of priorities could be defined, so that high priority or emergency communications triggered by human users or from the central Dispatch user, may take precedence over the “bot” communications, when needed. This would ensure that the first responder never loses communication with his human “counterpart”, the human Control Room Dispatch operator. This will ensure that “bots” mechanisms and communications are only used in a supportive, complementary way, so that the traditional “command chain” is respected, thus ensuring full human control in all times.

In addition to the example above, the paper [17] also outlines other possibilities in the Mission Critical framework. Finally, [17] also includes multimedia material to showcase how the concept could run in real life. The live scenario described in [17] is based on the Genaker MCPTT solution [135] and can be watched in associated URL [136].

One of the important conclusions of the use cases presented in [17] is that several decisions and actions happen without requiring intervention by the Control Room Dispatch operator, but by leveraging the MC Dispatch “bot” concept, which can automate certain decisions, support First Responders and assist the Control Room Dispatch users by making a set of pre-defined automated decisions based on available information at any time.

5.6.7 Conclusions and impact analysis

In section 5.6 we have described the concept of Mission Critical Dispatch “bots” and summarized the contributions of our paper published in IEEE Access [17]. Mission Critical Dispatch “bots” are entities capable of gathering environmental/situational information and triggering certain automated actions without the need of human intervention. [17] proves that in certain circumstances these “bots” can help to handle emergency situations in a more efficient way by distributing the traditionally centralized Dispatch role. Also, the fit of MC Dispatch “bots” into the 3GPP architecture for Mission Critical services has been presented, considering different architectural approaches and complexity levels. Importantly, because First Responders must operate hands-free most of the time it is of particular interest to convert “bot” interactions into audio information exchanged over MCPTT communication services, be it through the LTE network or leveraging the 3GPP device-to-device capability. By delivering information, advice and commands via voice, First Responders can keep their hands free for operational use. Hence, we will keep investigating how “bots” can enhance First Responders’ operations in the minimally possible intrusive way.

Potential evolution areas of these concepts include: a) Implementation of the proposed concepts into real usage, and b) Incorporation of other use cases by leveraging MCData and MCVideo capabilities developed by 3GPP, with particular attention to the use cases and requirements that triggered the definition of such new enablers [137] [138].

5.7 Conclusions

In the rest of chapter 5 we have provided an overview of some of the main SIP-based services that have been defined and standardized by OMA and 3GPP respectively. We have explained that after the definition of IMS as the Next Generation network architecture for 3GPP networks, and SIP as the signaling as the core signaling protocol, SIP has become a key element of future mobile services, starting with VoLTE and, more recently, with the standardization of MCPTT, MCData and MCVideo.

In this context, we have explained how SIP and RTSP share a number of commonalities, and explained how, after completion of a number of contributions related to the Packet-switched Streaming Service and the Real-time Streaming Protocol, our research focus expanded into IMS and SIP-based services, with particular focus on SIMPLE Presence and Group Communication services respectively.

We have explained how initially OMA defined the PoC Group Communication enabler, and how such work was later inherited and expanded further by 3GPP when it standardized the MCPTT standard, that will eventually replace many Public Safety legacy radio communication systems.

From the perspective of author's contributions, we can split them into two parts, the first one mostly related to OMA Presence SIMPLE and the second one related to Group Communication services [109].

In relation to Presence optimization, section 5.3 contains a detailed overview and analytical analysis of the Presence subscription / notification. This detailed study helped better understand how the Presence service works, which proved an invaluable background to the Presence related chapters of author's contribution to Multimedia Group Communications

With this background the author was also able to invent US Patent 9,143,574 [19], which has been widely cited by other patent applications in the field Presence optimizations.

As a side node, also in relation to the Presence area, the author also informally provided valuable input to the development of the Conditional Events Notifications framework as it was led by Aki Niemi [139]. Effectively, in the early stages of the work that would –years later– become IETF RFC 5839, the original intention by the authors was to avoid sending redundant notifications to Watchers. In the basic SUBSCRIBE / NOTIFY framework, every time a Watcher refreshes a subscription (e.g.: sends a re-SUBSCRIBE request) the Presence Server must issue a full notification transaction including the PIDF /

RPID document with full information about the Presentity. This mechanism is very inefficient, since in many cases the information being sent is already available at the Watcher. However, the baseline SUBSCRIBE / NOTIFY mechanism works in such inefficient way.

When [139] was originally being developed, the intention was to use an `ETag` header in the (re-)SUBSCRIBE message to inform the Presence Server about last received PIDF/RPID document. If the `ETag` value would match the latest known status at the Presence Server, there would be no need to send the redundant NOTIFY message.

At the early stages of the spec, however, the spec defined that if the Client `ETag` and Server `ETag` matched, no NOTIFY transaction would be initiated by the Presence Server. However, the original direction that the Internet-Draft was heading to, it was impossible to differentiate whether the Presence Server had decided not to send a NOTIFY message (to avoid sending redundant PIDF/RPID info) or if the NOTIFY transaction would be lost en route toward the Watcher UE. This situation is depicted as follows.

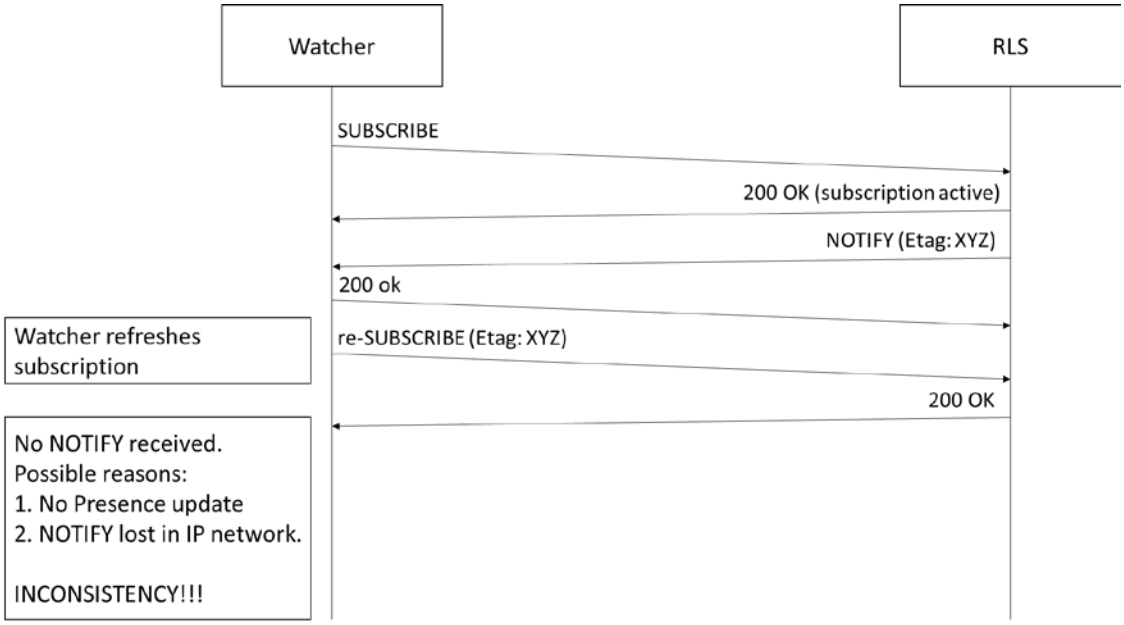

Figure 54. Inconsistency issue with Subnot Etags [139] if no 204 response code is used.

After our suggestion, the draft evolved into defining a new SIP response code (204 No Notification). By using this new response code, any possibility of inconsistency was removed:

- If a `200 OK` response is received to a SUBSCRIBE request with `Etag` value, the Watcher should expect a subsequent NOTIFY transaction, since the state has changed. The new NOTIFY transaction should carry a new `ETag` value.

- If a `204 No Notification` response is received, the Watcher understands that the Presentity has not changed its Presence status and latest `ETag` value remains valid.

The following picture depicts how the 204 No Notification response works, after it was introduced in [139].

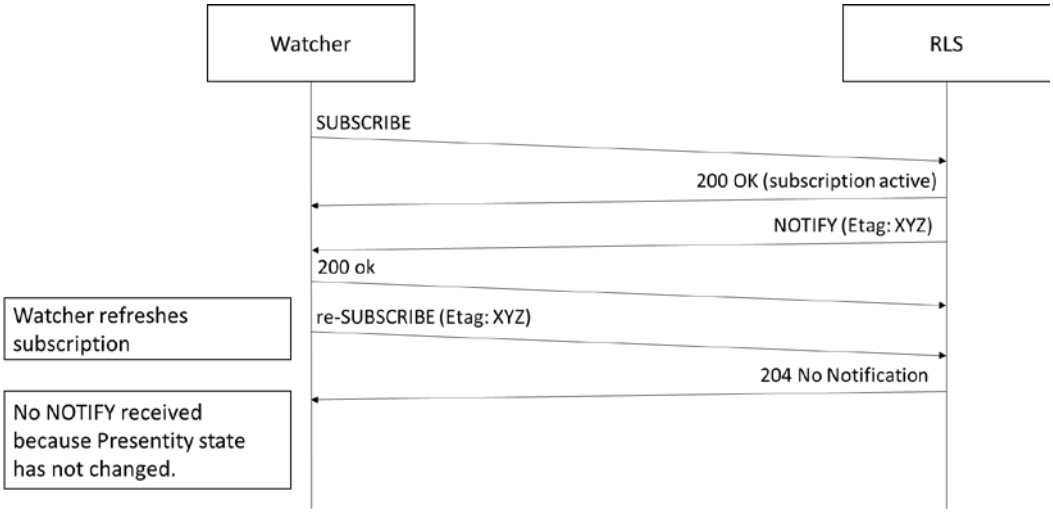

Figure 55. Resolving inconsistency issues in Subnot-Etags [139] with 204 response code.

In relation to Group Communications services, our main contributions are, on the one hand, co-authorship the only reference related to the OMA Push-to-Talk over Cellular service in the form of Wiley's book "Multimedia Group Communications" [109] together with Andrew Rebeiro-Hargrave, which has been significantly cited as one of the key references describing Push-to-Talk over Cellular networks.

Finally, the most recent contribution in this field [17] provides an overview of the up-to-date standardization of Group Communications for Public Safety and Mission Critical applications. Our contribution introduces innovative concepts such as non-human MC users, Dispatch "bots", distributed dispatching and associated architectural proposals. With the evolution toward 5G and Mobile Edge Computing (MEC) we are certain that these concepts will spread to the future of Mission Critical Communications.

6 Conclusions and Discussion

Our relationship with the Internet Multimedia Framework that comprises protocols such as RTP, RTSP, SIP, … has been an intense one over the last 20 years, both from an academic, research, professional and curiosity perspective. When we started our activities in this domain, we were far from understanding where the journey would lead to.

If we look back at all the contributions of this thesis work, we can see a relevant amount of commonality in all the developed work, as well as a good share of diversity in touching different-but-related areas with a common context: Internet Multimedia Protocols (Signaling Protocols, in particular) and applicability to the mobile domain.

Under such framework, if we think about those commonalities, one of the guiding principles of the whole workload, and probably an unnoticed, unintended "motto" all the way through, has been "Do more with less". I will try to explain why.

Since the early start of our thesis, teaching and professional activities, one of our obsessions has been around using standard protocols. Probably because through our research and academic journey we have been able to be intensively exposed to the elegance and beauty of standards, be it IETF RFCs, OMA Enabler Releases, 3GPP or ETSI Technical Specifications, or similar specifications by other SDO's, we have been working with them over the last 20 years.

Standards are not necessarily a good thing "per se". Sometimes they are complex, intricate, cumbersome, stupid, sub-optimal, outdated or incorporate too much bias to certain Industry influence (e.g.: mostly due to IPR and patent issues). However, part of their beauty comes from the fact that they simply "make things work". Standards are at the core of many beautiful technology solutions that are helping us today to have a much better life than 100, 200, or 1.000 years ago… if we –as a Civilization– wish to use such Standards for the good and for the benefit of all our human peers. Effectively, we can easily relate the fact that today we can have billions of devices connected to the Internet, to the fact that a few researchers and geeks, fifty years ago, decided to plant the seeds of the IP and TCP protocols… and they did so with a great share of altruism and generosity. Similarly, some aspects that today we are taking from granted, such as watching TV, listening to radio, connecting our computer to a WiFi network, or engaging into mobile phone calls ubiquitously (even with free roaming calls within Europe!), we can relate them to the fact that standards exit in these fields, that they are adopted, managed and enforcement by governments, SDOs and regulators.

How does it all come together in the context of this Thesis work? As I said above, a guiding principle of most of our activity has been around "doing more with less". Effectively, by studying the relevant standards in detail such as RTSP (RFC 2326 and 7826), 3GPP Packet-switched Streaming (e.g.: 3GPP TS 26.233,

TS 26.234, TS 26.247, …), SIP and RTP (RFC 3261, 3550), all SIP/SIMPLE Presence related specs (RFC 3265, RFC 6665, RFC 5839, RFC 3863, RFC 4662), all OMA related enablers (OMA SIMPLE Presence, OMA Push-to-Talk over Cellular, OMA XDMS) plus the innumerable 3GPP specs related to IMS and MCPTT (e.g.: 3GPP TS 22.179, 23.379, 24.280, …) our underlying intention has been to understand those standards and architectures, identify gaps or limitations and enhance them with the minimal possible disruption to the standard itself.

Because standards are a key enabler of interoperability, in particular cross-vendor interoperability, our approach has not been disruptive in the sense of creating something completely now from scratch. Rather, we have always intended to ensure alignment to standards or –when deviations are unavoidable– a way to ensure backwards compatibility with the core standard. In a nutshell, our intention has been to maximize the benefits of enhancing or improving existing standards, while minimizing the impact in key concepts such as interoperability and backwards compatibility.

In this framework, our focus has generally been investigating multimedia signaling protocols and architectures performance over 3GPP networks. Both SIP and RTSP are signaling protocols. While RTSP was initially conceived to support live and on-demand content delivery from media servers and SIP was initially conceived to support multiparty multimedia conferences, the scope of SIP –particularly since it was selected as the multimedia signaling protocol in IMS and NGN– has consistently expanded over the last 20 years, while RTSP has remain relatively stable over time. Hence, while our initial thesis work focused on RTSP architecture and enhancements, our focus gradually shifted –always with multimedia signaling in mind– toward SIP-based services, with particular emphasis on SIMPLE Presence, OMA PoC and 3GPP MCPTT.

Thesis work is summarized in the Annex IV and comprises publication of one co-authored book, three International Journal papers (all of them indexed in Journal Citation Reports (JCR)), one International Conference papers, one National Conference paper and one granted International Patent. The scope of our contributions is mostly described in chapters 3 (which mainly summarizes preliminary work in collaboration with other authors) as well as the chapters that contain the bulk of author-led contributions: 4 (which contains the description of the work related to PSS and RTSP optimization) and 5 (which describes the contribution in the area of SIP/SIMPLE Presence and OMA PoC / 3GPP MCPTT contributions).

While all these contributions are described in the relevant chapters, we will provide a quick overview in the rest of this section, as a conclusion of the Thesis document.

In relation to streaming related research activities, our first contribution represents a good example of the “doing more with less” paradigm. Effectively, in the “Session-information Based Admission Control Strategy for All-IP networks” paper [16] we simply proposed a way in which both the Application layer

and the Network layer could collaborate to improve the admission control policy in 3GPP networks. We did not propose any modification of the RTSP protocol, but simply outlined how –by reusing information already available in RTSP/SDP signaling messages– it is possible to estimate future session behavior and apply such estimation to enhance network admission control strategies. Interestingly, when the paper was developed an interface between the Application Function and the Network had recently been defined at 3GPP (the Gq interface, between the AF and the PCF). Such interface and the definition of network nodes evolved over time (e.g.: Rx interface and PCRF, in 4G), however the principle that the application layer and the network layer may collaborate to mutual benefit remains in 3GPP architecture across generations (e.g.: N5 interface between the AF and the PCF in 5G).

Our two main other contributions when it comes to RTSP / PSS relate to slightly modifying the RTSP protocol to enable fast session setup, message pipelining, fast triggering of Secondary PDP-Context activation (in 3GPP networks with QoS control and –eventually– fast channel switching in a broader IPTV context). These contributions are summarized in [46] [18]. This work also kept the principle of achieving more with less. In particular, a few modifications of the baseline RTSP protocol were introduced with the care of ensuring backwards compatibility through the `Supported` RTSP header. Essentially, by sharing a session-id as early as possible in a slightly modified RTSP session setup procedure it is possible to avoid several RTT's and establish streaming sessions faster than with regular RTSP / PSS. The other main contribution also aimed at reducing the amount of SDP information shared over the air interface by splitting the SDP message into a fixed and variable part respectively. The fixed part could be cached by the client and reused from presentation to presentation (without need to downloading it every time a new session is to be started) while only the dynamic part would be sent during the DESCRIBE transaction, thus consuming less radio bandwidth and further reducing session setup time.

After completion of such work [46] [18] it was interesting to see how 3GPP and IETF implemented a feature that, formally, resembles the RTSP / PSS Early Setup concept proposed by the author. This is described in further detail in section 4.7, while Annex II contains an excerpt from the relevant 3GPP spec that described the Pipelined-Requests concept.

In addition to the Pipelined-Requests header that fulfils a goal very similar to the Early Setup concept, it is interesting to note that, while 3GPP has moved away from RTSP-based streaming in recent releases (with the incorporation of DASH and SAND over DANE) some of the underlying design principles of our work remain valid still today (e.g.: caching, fast content setup, fast channel switching, network-application collaboration, …).

After the completion of our thesis work related to RTSP and PSS work, in chapter 5 we have explained that we shifted our area of activity into services based on a protocol that shared a lot of commonalities with RTSP (text-based, same author, multimedia signaling, extensible, SDP body, …).

After the definition of IMS as the Next Generation network architecture for 3GPP networks, and SIP as the signaling as the core signaling protocol, SIP became a key element of future mobile services, starting with VoLTE and, more recently, with the standardization of MCPTT, MCData and MCVideo. Hence, and since RTSP-based streaming progressively evolved into HTTP-based streaming where the signaling layer would become less relevant, so our research focus expanded into IMS and SIP-based services, with particular focus on SIMPLE Presence and Group Communication services respectively.

Our initial area of work focused on SIP/SIMPLE Presence. We started by investigating the impact of RLS vs. individual subscriptions in an IMS vs. IETF context, which is described in section 5.3. While this work remained unpublished, it became key for us to be able to consistently describe SIP SIMPLE Presence in the Multimedia Group Communication book [109] and allowed us to invent a new patent to let a Watcher signal an “idle” state to the Presence Server and avoid unnecessary redundant Presence traffic over the air link [19], which has become widely cited by new patents and patent reviews in the area.

In the context of SIP / SIMPLE Presence, and as a curiosity, we have also explained in section 5.7 that our comments and recommendations shared with author of RFC 5839 represented a modest contributor that – over time– would lead to the creation of the SIP 204 response code.

In the area of Group Communications over 3GPP networks based on the PoC standard and the later evolution into 3GPP MCPTT, the author has co-authored the first and only book about the OMA PoC service, which has become the only reference in the field, cited by several authors. Last but not least, our latest contribution, in a high impact factor Journal at the end of 2017 [17], allowed us to present an overview of the up-to-date standardization of Group Communications for Public Safety and Mission Critical applications. Our contribution introduces innovative concepts such as non-human MC users, Dispatch “bots”, distributed dispatching and associated architectural proposals. Again, our aim with this contribution has been to follow the “doing more with less” and “keep it simple” principles, showcasing how the standard can be implemented in a way that it enriches the services delivered, without necessarily breaking the standard itself.

In a nutshell, the author contributions described above are the ones that build up this Thesis work. They are summarized in annex C and, for the sake of convenience, an overview is also provided here:

- (Book) A. Rebeiro-Hargrave, D. Viamonte. Multimedia Group Communication. Push-to-Talk over Cellular, Presence and List Management Concepts and Applications. John Wiley & Sons publishing (ISBN: 978-0470058534). February 2008 [83].
- (International Journal) D. Viamonte, A. Calveras. A Distributed Man-Machine Dispatching Architecture for Emergency Operations Based on 3GPP Mission Critical Services. IEEE Access (indexed in JCR). December 2017 [19].
- (International Journal) D. Viamonte, A. Calveras, J. Paradells, C. Gómez Montenegro. Evaluation and optimization of Session Setup Delay for Streaming Services over 3G Networks with Quality-of-Service Support. Wiley Wireless Communications and Mobile Computing (WCMC'08) (indexed in JCR). May 2008 (first published online, September 2006) [20].
- (International Journal) C. Gómez, M. Catalán, D. Viamonte, J. Paradells, A. Calveras. Web browsing optimization over 2.5G and 3G: end-to-end mechanisms Vs usage of performance enhancing proxies. Wiley Wireless Communications and Mobile Computing (WCMC'07) (indexed in JCR). June 2007 [24].
- (International Conference) E. Garcia, R. Vidal, D. Viamonte, J. Paradells. Achievable Bandwidth Estimation for Stations in Multi Rate IEEE 802.11 WLAN Cells. IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM'2007). June 2007 [25].
- (International Conference) D. Viamonte, A. Calveras, J. Paradells, C. Gómez Montenegro. Requirements for the Optimization of Session Setup Delay and Service Interactivity in Streaming Services over 3G. Proceedings of the European Wireless Conference (EW2006). April 2006 [48].
- (International Conference) C. Gómez, M. Catalán, D. Viamonte, J. Paradells, A. Calveras. Internet traffic analysis and optimization over a precommercial live UMTS network. IEEE Vehicular Technology Conference Spring (VTC Spring'05). April 2005 [23].
- (International Conference) M. Catalán, C. Gómez, D. Viamonte, J. Paradells, A. Calveras, F. Barceló. TCP/IP analysis and optimization over a precommercial live UMTS network. IEEE Wireless Communications and Networking Conference (WCNC'05). March 2005 [22].
- (International Conference) D. Viamonte, A. Calveras. Session Based Admission Control Strategy for Streaming Services over All-IP 3G Networks. Proceedings of the 15th IEEE International Workshop on Personal, Indoor and Mobile Radio Communications (PIMRC'04). September 2004 [18].
- (National Workshop) D. Viamonte. Streaming over 3GPP wireless networks. Challenges and Issues. Proceedings of the Workshop on Internet Usage over 2.5G and 3G. IST-2001-92125. March 2003 [140].

- (International Patent) D. Viamonte, Presence System and a Method for providing a Presence System, US Patent US20090319655 A1, June 2008 (Granted: 2015) [21].

After the completion of this work, our future research will focus on the evaluation and enhancement of SIP-based multimedia architectures for Mission Critical services over 5G. Effectively, with the introduction of a new 5G architecture, which can be split into core vs. MEC in several different ways based on Network Function Virtualization (NFV), our next areas of activity will spread into MCPTT over 5G as well as MCVideo, MCData and the impact of 5G into other relevant areas, from an MCPTT perspective, such as eMBMS or ProSe.

The future of Mission Critical Communications, as well as the evolution of SIP and IMS-based services is today in an exciting moment. 5G promises to merge a number of Industry trends that will completely transform the industry. From connected vehicle, to ultra-reliable communications, from Industry 4.0 and IoT to the future of railway communications. 5G, coupled with NFV will represent a new wave of challenges for system architects of the next decade. Effectively, while 5G will support new exciting services, there will be a fundamentally new layer of complexity, since many alternative deployment strategies and business models will be possible, based on Network Function Virtualization (NFV) and slicing of access, core and application layer VNFs. We shall actively monitor, research and –hopefully– influence how the Mission Critical community will leverage such revolution for the benefit of the Society as a whole.

This has been a long, exciting journey where we have never stopped learning and exploring. We have felt pressure, progress, writer's block, loneliness, excitement, stress, joy… life.

References

[1] U. Black, ISDN and SS7: Architectures for Digital Signaling Networks, New Jersey: Prentice-Hall, May 1997.

[2] I. T. U. (ITU-T), "ISDN user-network interface layer 3 specification for basic call control (Q.931)," May 1998.

[3] IETF, "Number of RFCs published per year," [Online]. Available: https://www.rfc-editor.org/rfcs-per-year/.

[4] IETF, «Number of RFC's Published per Year,» [En línea]. Available: https://www.rfc-editor.org/rfcs-per-year/.

[5] J. Postel, *Internet Protocol, Darpa Internet Program Protocol Specification, RFC 791,* September 1981.

[6] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications,* May 1974.

[7] J. Postel, "User Datagram Protocol," *Internet Standard, RFC 791,* August 1980.

[8] J. Rosenberg, H. Schulzrinne, G. Camarillo and A. Johnston, *Session Initiation Protocol (SIP), IETF RFC 3261,* June 2002.

[9] M. Handley, V. Jacobson and C. Perkins, SDP: Session Description Protocol, IETF RFC 4566, July 2006.

[10] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, *Real-time Transport Protocol (RTP), IETF 3550,* July 2003.

[11] H. Schulzrinne, A. Rao and R. Lanphier, *Real Time Streaming Protocol (RTSP), IETF RFC 2326,* April 1998.

[12] H. Schulzrinne, M. Westerlund, A. Rao, R. Lanphier and M. Stiemerling, «Real-time Streaming Protocol Version 2.0; IETF RFC 7826,» December 2016.

[13] *3GPP 23.002v13.3.0, 3GPP Network Architecture (Release-13),* March 2016.

[14] D. Astély, E. Dahlman et. al.., *LTE: The evolution of Mobile Broadband; IEEE Communications Magazine,* 2009.

[15] *3GPP 22.179v13.3.0 Mission Critical (MCPTT) over LTE; Stage 1,* December 2015.

[16] D. Viamonte and A. Calveras, *Session Based Admission Control Strategy for Streaming Services over All-IP 3G Networks; 15th IEEE International Workshop on Personal, Indoor and Mobile Radio Communications (PIMRC'04),* September 2004.

[17] D. Viamonte and A. Calveras, «A Distributed Man-Machine Dispatching Architecture for Emergency Operations Based on 3GPP Mission Critical Services; IEEE Access,» de *IEEE Access Special Topic on Mission Critical Public-Safety Communications: Architectures, Enabling Technologies and Future Applications; Volume 6; Pages 11614 - 11623,* December 2017.

[18] D. Viamonte, A. Calveras, J. Paradells and C. G. Montenegro, «Evaluation and optimization of Session Setup Delay for Streaming Services over 3G Networks with Quality-of-Service Support. Wiley Wireless Communications and Mobile Computing (WCMC),» *Wiley Wireless Communications and Mobile Computing (WCMC),* May 2008 (first published online, September 2006).

[19] D. Viamonte, «Presence System and a Method for Providing a Presence Service; US patent US20090319655A1,» December 2009, September 2015 (Granted).

[20] M. Catalán, C. Gómez, D. Viamonte, J. Paradells, A. Calveras and F. Barceló, «TCP/IP analysis and optimization over a precommercial live UMTS network; IEEE Wireless Communications and Networking Conference (WCNC'05),» March 2005.

[21] C. Gómez, M. Catalán, D. Viamonte, J. Paradells and A. Calveras, «Internet traffic analysis and optimization over a precommercial live UMTS network; IEEE Vehicular Technology Conference Spring (VTC Spring'05),» April 2005.

[22] C. Gómez, M. Catalán, D. Viamonte, J. Paradells and A. Calveras, «Web browsing optimization over 2.5G and 3G: end-to-end mechanisms Vs usage of performance enhancing proxies; Wiley Wireless Communications and Mobile Computing (WCMC'07),» June 2007.

[23] E. Garcia, R. Vidal, D. Viamonte and J. Paradells, «Achievable Bandwidth Estimation for Stations in Multi Rate IEEE 802.11 WLAN Cells. IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM'2007),» June 2007.

[24] J. Huang, F. Xian, A. Gerber, Z. M. Mao, S. Sen and O. Spatscheck, «A Close Examination of Performance and Power Management of 4G LTE Networks,» *Proceedings of the 10th international conference on Mobile systems, applications, and services (MOBISYS'12)*, pp. 225-238, June 2012.

[25] *Overview of 3GPP Release-4 Features*, July 2004.

[26] *3GPP TS 26.233v4.2.0, Packet-switched Streaming Service (PSS). General Overview (Release-4)*, June 2002.

[27] *3GPP TS 22.233v6.3.0, Transparent end-to-end packet-switched streaming; Stage 1 (Release-6)*, March 2009.

[28] *3GPP TS 26.233v6.0.0, Transparent end-to-end packet-switched streaming (PSS); General description (Release-6)*, September 2004.

[29] *3GPP TS 26.234v4.5.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs*, December 2002.

[30] *3GPP TS 26.234v6.14.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release-6)*, March 2009.

[31] *3GPP TS 22.140v4.3.0, Multimedia Messaging Service. Stage 1 (Release-4)*, December 2002.

[32] I. Elsen, F. Hartung et. al.., *Streaming Technology in 3G Mobile Communication Systems; IEEE Computer Magazine*, September 2001.

[33] H. Montes, G. Gómez and R. Cuny, *Deployment of IP Multimedia Streaming Services in Third-Generation Mobile Networks; IEEE Wireless Communications*, October 2002.

[34] D. Durham, J. Boyle et. al.., «The COPS (Common Open Policy Server) Protocol; RFC 2748,» January 2000.

[35] V. Fajardo, J. Arkko, J. Loughney and G. Zorn, «Diameter Base Protocol; RFC 6733,» October 2012.

[36] *3GPP TS 26.234v5.7.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release-5),* April 2005.

[37] *3GPP TS 23.207v6.6.0 End-to-end Quality-of-Service (QoS) concept and architecture (Release 6),* October 2005.

[38] *3GPP TR 23.917v1.2.0, Dynamic Policy control enhancements for end-to-end QoS (Release-6) (spec withdrawn),* January 2004.

[39] P. Frödj, U. Horn et. al.., *Adaptive Streaming within the 3GPP Packet-switched Streaming Service; IEEE Network,* April 2006.

[40] *3GGP TS 26.244v6.7.0,3GPP Packet-switched Streaming Service (PSS). 3GP File Format (Release-6),* June 2007.

[41] *3GPP TR 26.937v8.0.0, Transparent end-to-end Packet-switched Streaming Service (PSS). RTP Usage Model (Release-8),* December 2008.

[42] *3GPP TS 26.247, Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (Release-10),* December 2014.

[43] O. Oyman and S. Singh, *Quality of Experience (QoE) for HTTP Adaptive Streaming Services; IEEE Communications Magazine,* April 2012.

[44] A. Hernández, A. Valdovinos and F. Casadevall, «Capacity Analisys and Performance Evaluation of Call AC for Multimedia Packet Transmission in UMTS WCDMA System,» *Proceedings of Wireless Communications and Networking (WCNC),* March 2003.

[45] S.-E. Elayoubi, T. Chahed and G. Hêbuterne, «Measurement-based Admission Control in UMTS: Multiple Cell Case,» *Proceedings of the International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC),* March 2003.

[46] D. Viamonte, A. Calveras, J. Paradells and C. GómezMontenegro, *Requirements for the Optimization of Session Setup Delay and Service Interactivity in Streaming Services over 3G; XIIth European Wireless Conference (EW2006),* April 2006.

[47] *3GPP TS 23.203v15.4.0, Policy and Charging Control Architecture (Release-15),* September 2018.

[48] E. Burger, «A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages; RFC 4483,» May 2006.

[49] J. Peisa and M. Meyer, «Analytical Model for TCP File Transfer over UMTS, P roceedings of the IEEE 3G Wireless Conference,» June 2001.

[50] A. Simonsson, J. Peisa and J. Pettersson, «Analytic study of TCP performance over a soft rate switching WCDMA bearer. The16th Annual IEEE International Symposium on Personal Indoorand Mobile Radio Communications (PIMRC 2005),» September2005.

[51] «3GPP TR 25.993v15.0.0; Typical Examples of Radio Access Bearers (RABs) and Radio Bearers (RBs) supported by Universal Terrestrial Radio Access (UTRA) (Release-15),» June 2018.

[52] «3GPP TS 25.322v.15.0.0: Radio Link Control (TLC) protocol specification (Release-15),» June 2018.

[53] G. Cheung, T. Wai-tian and T. Yoshimura, «Rate-distortion optimised application-level retransmission using streaming agent for videostreaming over 3G wireless network; Proceedings of the IEEE International Conference on Image Processing,» September 2002.

[54] A. Atayero, M. Luka, M. Orya and J. Iruemi, «3GPP Long Term Evolution: Architecture, Protocols and Interfaces; International Journal of Information and Communication Technology Research (IJICT),» November 2011.

[55] G. Abed, M. Ismail and K. Jumary, «The Evolution to 4G Cellular Systems: Architecture and Key Features of LTE-Advanced Networks; IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC),» 2012.

[56] T.-T. Tran, Y. Shin and O. Shin, «Overview of enabling technologies for 3GPP LTE-advanced; EURASIP Journal on Wireless Communications and Networking,» December 2012.

[57] «3GPP TS 29.214v15.5.0 Policy and Charging Control over Rx reference point (Release-15),» December 2018.

[58] «3GPP TS 26.233v15.0.0 Transparent end-to-end Packet-switched Streaming service (PSS); General description (Release-15),» June 2017.

[59] «3GPP TS 26.234v15.1.0 Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release-15),» September 2018.

[60] «3GPP TS 26.247v16.1.0 Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Adaptive Streaming over HTTP (3GP-DASH) (Release-16),» December 2018.

[61] «3GPP 23.002v15.0.0 3GPP Network Architecture (Release-15),» March 2018.

[62] «3GPP TS 23.501v15.4.0 System Architecture for the 5G System, Stage 2 (Release-15),» December 2018.

[63] «3GPP TS 26.234v8.4.0 Packet-switched Streaming (PSS); Protocols and Codecs (Release-8),» September 2009.

[64] «3GPP TS 26.237v11.3.0 IP Multimedia Subsystem (IMS) based Packet Switch Streaming (PSS) and Multimedia Broadcast/Multicast Service (MBMS) User Service; Protocols (Release-11),» December 2013.

[65] «3GPP TR 26.953v14.0.0 Interactivity support for 3GPP-based streaming and download services (Release-14),» March 2017.

[66] «3GPP TR 22.816v14.1.0 3GPP Enhancements for TV Service (Release-14),» March 2016.

[67] «3GPP TR 22.833v16.0.0 Study on Enhancement of LTE for Efficient delivery of Streaming Service (Release-16),» May 2018.

[68] «3GPP TS 26.501 5G Media Streaming (5GMS); General description and architecture (Release-16) (work-in-progress)».

[69] M. Zimmermann and G. Seilheimer, «Reviewing HTTP and RTSP Work in Two Actual Commercial Media Delivery Platforms for Multimedia Services and Mobile Devices,» de

Multimedia Services and Streaming for Mobile Devices: Challenges and Innovations, IGI Global, September 2011, pp. 91-100.

[70] V. Vasanthi and M. Chidambram, «A Study on Video Streaming in Cloud Environment; International Journal of Emerging Technology and Advanced Engineering (IJETAE),» March 2015.

[71] H. Schulzrinne, A. Rao and R. Lanphier, «IETF Interned Draft Real Time Streaming Protocol (RTSP) 2.0 (standards draft),» February 2002.

[72] J. Rosenberg, J. Weinberger, C. Huitema and R. Mahy, «STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs); RFC 3489,» March 2003.

[73] R. Mahy, P. Matthews and J. Rosenberg, «Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN); RFC 5766,» April 2010.

[74] J. Rosenberg, «Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols; RFC 5245,» April 2010.

[75] H. Schulzrinne and J. Rosenberg, «The IETF Internet Telephony Architecture and Protocols, IEEE Network,» June 1999.

[76] K. Rosenbrock, R. Sanmugam, S. Bradner and J. Klensin, *IETF RFC 3113, 3GPP-IETF Standardization Collaboration*, June 2001.

[77] *3GPP TS 22.173v14.1.0, IMS Multimedia Telephony and Supplementary Services. Stage-1 (Release-14),* March 2016.

[78] *3GPP 23.228v13.5.0, IP Multimedia Subsystem (IMS); Stage 2 (Releas-13),* March 2016.

[79] *Open Mobile Alliance; OMA Push to Talk over Cellular (PoC) v1.0.4*, December 2009.

[80] *Open Mobile Alliance; OMA Push-to-Talk over Cellular (PoC) v2.0 v2.1*, August 2011.

[81] «Open Mobile Alliance; OMA XML Document Management (XDM) v1.1,» June 2008.

[82] «Open Mobile Alliance; OMA XML Document Management (XDM) v2.2,» *May 2016.*

[83] *Open Mobile Alliance; OMA Presence SIMPLE v1.1*, February 2010.

[84] *Open Mobile Alliance; OMA Presence SIMPLE v2.0,* July 2012.

[85] «Open Mobile Alliance; OMA SIMPLE Instant Messaging (IM) v1.0,» *August 2012.*

[86] «Open Mobile Alliance; OMA SIMPLE Instant Messaging (IM) v2.0,» *March 2015.*

[87] «3GPP TS 23.179v13.1.0, Functional architecture and information flows to support mission critical communication services; Stage 2 (Releas-13),» *March 2016.*

[88] C. Chi et. al.., *IMS Presence Server: Traffic Analysis and Performance Modelling; IEEE International Conference on Network Protocols (ICNP'08),* October 2008.

[89] e. a. Z. Cao, *User Behavior Modeling and Traffic Analysis of IMS Presence Servers; IEEE Global Telecommunications Conference (GLOBECOM'08),* December 2008.

[90] J. Rosenberg, «SIMPLE made Simple: An Overview of the IETF Specificacions for Instant Messaging and Presence Using the Session Initiation Protocol (SIP); RFC 6914,» April 2013.

[91] A. B. Roach, «SIP-Specific Event Notification; RFC 3265,» June 2002..

[92] J. Rosenberg, «A Presence Event Package for SIP; RFC 3856,» October 2004..

[93] F. Rui et. al., «Evaluation and Analysis of SIP and VoIP Performance with Presence Traffic over HSPA, 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07),» September 2007.

[94] C. Urrutia-Valdés, «Presence and Availability with IMS: Applications, Architecture, Traffic Analysis and Capacity Impacts; Bell Labs Technical Journal,» 2006.

[95] F. McKeon, «A study of SIP based Instant Messaging Focusing on the Effects of Network Traffic Generated due to Presence; IEEE International Symposium on Consumer Electronics 2008,» April 2008.

[96] F. Wegscheider, «Minimizing Unnecessary Notification Traffic in the IMS Presence System; 1st International Symposium on Wireless Pervasive Computing 2006,» 2006.

[97] A. Roach, B. Campbell and J. Rosenberg, «A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists; RFC 4662,» August 2006.

[98] H. Sugano et. al.., «Presence Information Data Format; RFC 3863,» August 2004.

[99] H. Schulzrinne, V. Gurbani, P. Kyzivat and J. Rosenberg, «RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF); RFC 4480,» July 2006.

[100] A. Houri et. al., «Presence Interdomain Scaling Analysis for SIP/SIMPLE; Internet-Draft,» Aufust 2009.

[101] C. Chi et. al.., «IMS Presence Server: Traffic Analysis and Performance Modelling; IEEE International Conference on Network Protocols (ICNP'08),» October 2008.

[102] V. Beltran and J. Paradells, «SIP/SIMPLE Resource List Server: Optimization or Burden for Presence Systems; Proceedings of the 4th IEEE Conference on Context Awareness for Proactive Systems,» May 2011.

[103] «3GPP TR 24.141v12.1.0 Presence Service using the IP Multimedia (IM) Core Network (CN) Subsystem; Stage 3 (Release-12),» December 2012.

[104] M. Lonnfors et. al.., «Session Initiation Protocol (SIP) Extension for Partial Notification of Presence Information; RFC 5263,» September 2008..

[105] M. García-Martín, «The Presence-Specific Static Dictionary for Signaling Compression; RFC 5112,» January 2008..

[106] A. Niemi, «An extension to Session Initiation Protocol (SIP) Events for Conditional Event Notification; RFC 5839,» May 2010.

[107] A. Niemi et. al.., «Session Initiation Protocol (SIP) Event Notification Extension for Notification Rate Control; RFC 6446,» January 2012.

[108] M. Mijatovic, «Critical Communications State of the Play; The International Critical Control Rooms Congress 2017,» December 2017.

[109] A. Rebeiro-Hargrave and D. Viamonte, *Multimedia Group Communication*, John Wiley & Sons, February 2008.

[110] A. Kumbhar, F. Koohifar, I. Güvenç and B. Mueller, «A Survey on Legacy and Emerging Technologies for Public Safety Communications,» *IEEE Communications Surveys and Tutorials (submitted)*, Sept. 2015.

[111] M. Simic, «Feasibility of Long Term Evolution (LTE) as a technology for Public Safety; 20th Telecommunication Forum (TELFOR),» November 2012.

[112] R. Fantacci, F. Gei, D. Marabissi and L. Micciullo, «Public safety networks evolution toward broadband: sharing infrastructures and spectrum with commercial systems; IEEE Communications Magazine,» April 2016.

[113] A. Kuwadekar and K. Al-Begain, «A real world evaluation of Push to Talk service over IMS and LTE for public safety systems; IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),» October 2014.

[114] A. Ali, M. Alshamrani, A. Kuwadekar and K. Al-Begain, «Evaluating SIP Signaling Performance for VoIP over LTE Based Mission-Critical Communication Systems; 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies,» Sept. 2015.

[115] M. Zambrano, I. Pérez, F. Carvajal, M. Esteve and C. Palau, «Command and Control Information Systems applied to Large Forest Fires Response; IEEE Latin America Transations, Volume 15, Issue 9, Pages 1735 – 1741,» August 2017.

[116] D. Wang et. al., «Optimal design of command and control organizational communication network based on task; IEEE IAEAC 2017; Chongqing, China,» October 2017..

[117] A. Merwaday and I. Guvenc, «UAV assisted heterogeneous networks for public safety communications; IEEE WCNCW 2015; New Orleans, USA,» June 2015..

[118] A. Alnoman and A. Alagan, «On D2D communications for public safety applications; IEEE IHTC 2017; Toronto, Canada,» July 2017.

[119] A. Orsino et. al., «Effects of Heterogeneous Mobility on D2D- and Drone-Assisted Mission-Critical MTC in 5G; IEEE Communications Magazine, Volume 55, Issue 2, Pages 79 – 87,» February 2017..

[120] J. Undung et. al., «Fire Locator, Detector and Extinguisher Robot with SMS Capability; IEEE HNICEM 2016; Cebu City, Philippines,» January 2016..

[121] S. V. P. Kumar Maddukuri et. al.., «A low cost sensor based autonomous and semi-autonomous fire-fighting squad robot; IEEE ISED 2016; Patna, India,» July 2017..

[122] L. Marusich et. al., «Effects of Information Availability on Command-and-Control Decision Making. SAGE Human Factors 2016, Volume 58, Issue 2, Pages 301-321,» 2016.

[123] S.-H. Lien et. al., «Enhanced LTE Device-to-Device Proximity Services; IEEE Communications Magazine, , Volume 54, Issue 12, Pages 174-182,» December 2016.

[124] J. Oueis et. al., «Overview of LTE Isolated E-UTRAN Operation for Public Safety; IEEE Communications Standards Magazine, Volume 1, Issue 2, Page 98 – 105, 2017,» July 2017.

[125] B. Duncan and R. Murphy, «Field study identifying barriers and delays in data-to-decision with small unmanned aerial systems; IEEE International Conference on Technologies in Homeland Security (IEEE HST 2013), Waltham, USA,» January 2014..

[126] «3GPP TS 22.179v16.4.0 Mission Critical PTT (MCPTT) over LTE – Stage 1 (Release-16),» December 2018.

[127] P. Stavroulakis, «TErrestrial Trunked RAdio - TETRA: A Global Security Tool (Signals and Communication Technology),» Springer-Verlag Berlin Heidelberg, September 2011.

[128] «TIA TSB-102; APCO Project 25 System and Standards Definition,» 2006.

[129] «3GPP TS 23.282v15.4.0 Common functional architecture to support Mission Critical Data (MCData); Stage 2 (Release-15),» June 2018.

[130] «3GPP TS 23.281v15.5.0 Common functional architecture to support Mission Critical Video (MCVideo) - Stage 2 (Release-15),» June 2018.

[131] «3GPP TS 23.379v15.5.0 Functional architecture and information flows to support Mission Critical Push To Talk (MCPTT) - Stage 2,» September 2018.

[132] «3GPP TS 23.303v15.1.0 Proximity-based Services (ProSe) – Stage 2 (Release-15),» June 2017.

[133] «3GPP TS 23.280v15.5.0 Common functional architecture to support mission critical services; Stage 2 (Release-15),» December 2018.

[134] G. Steinbauer and A. Kleiner, «Towards CSP-based mission dispatching in C2/C4i systems; IEEE International Symposium on Safety, Security and Rescue Robotics (IEEE SSRR 2012), College Station, Texas,» November 2012.

[135] «First ETSI LTE Mission-Critical Push to Talk interoperability tests achieve 85% success rate.,» de *http://www.etsi.org/news-events/news/1201-2017-06-news-first-etsi-lte-mission-critical-push-to-talk-interoperability-tests-achieve-85-success-rate*, June 2017.

[136] D. Viamonte, «Multimedia Material. MCPTT Demo associated to MCPTT paper in IEEE Access,» [En línea]. Available: https://ieeexplore.ieee.org/document/8222966/media#media.

[137] «3GPP TS 22.281v15.1.0 Mission Critical (MC) Video over LTE (Release-15),» January 2018.

[138] «3GPP TS 22.282v16.4.0 Mission Critical (MC) Data over LTE,» December 2018.

[139] A. Niemi and D. Willis, «An Extension to SIP Events for Conditional Event Notification; IETF RFC 5839,» May 2010.

[140] D. Viamonte, «Streaming over 3GPP wireless networks. Challenges and Issues; Workshop on Internet Usage over 2.5G and 3G. IST-2001-92125,» March 2003.

[141] R. Preiskel and N. Higham, "Liberalization of telecommunication infrastructure and cable television networks," *The European Commission's Green Paper. Telecommunications Policy*, 1995.

[142] J. Rosenberg, H. Schulzrinne et. al., «Session Initiation Protocol (SIP); RFC 3261,» June 2002.

Annex A. Example RTSP flows and messages related to Early Setup and SDP Template usage

In this section we provide a more detailed overview of the signaling flows and message structure of regular PSS RTSP/SDP-based session setup, as well as the alternative setup signaling and message structure if the optimizations proposed by the the author in section 4.6 are implemented. The theoretical enhancements that can be achieved by implementing these two procedures are described in section 4.6.6. Note that the RTSP and SDP messages and sizes used in section 4.6.6 and in [46] [18] were based on a real RTSP/Streaming client implementation, while the messages described in this annex are theoretical and based on [12].

The standard reference session setup based on RTSP and PSS without any further optimization is depicted in the figure below. Note that the reference scenario assumes that a Secondary PDP-Context is enabled for media delivery (3GPP network with QoS support) and that the multimedia session consists of one audio and one video stream. In such scenario the session setup procedure consists on the following items:

- TCP three-way handshake.
- An RTSP DESCRIBE / 200 OK transaction (steps 1, 2).
- An RTSP SETUP / 200 OK transaction for the video stream (steps 3, 4).
- An RTSP SETUP / 200 OK transaction for the audio stream (steps 5, 6).
- The setup of the Secondary PDP-Context to carry audio and video.
- The PLAY / 200 OK transaction to trigger media streaming delivery over RTP.

The scenario is depicted below. Note that we will use signaling examples from [12] in the rest of this section.

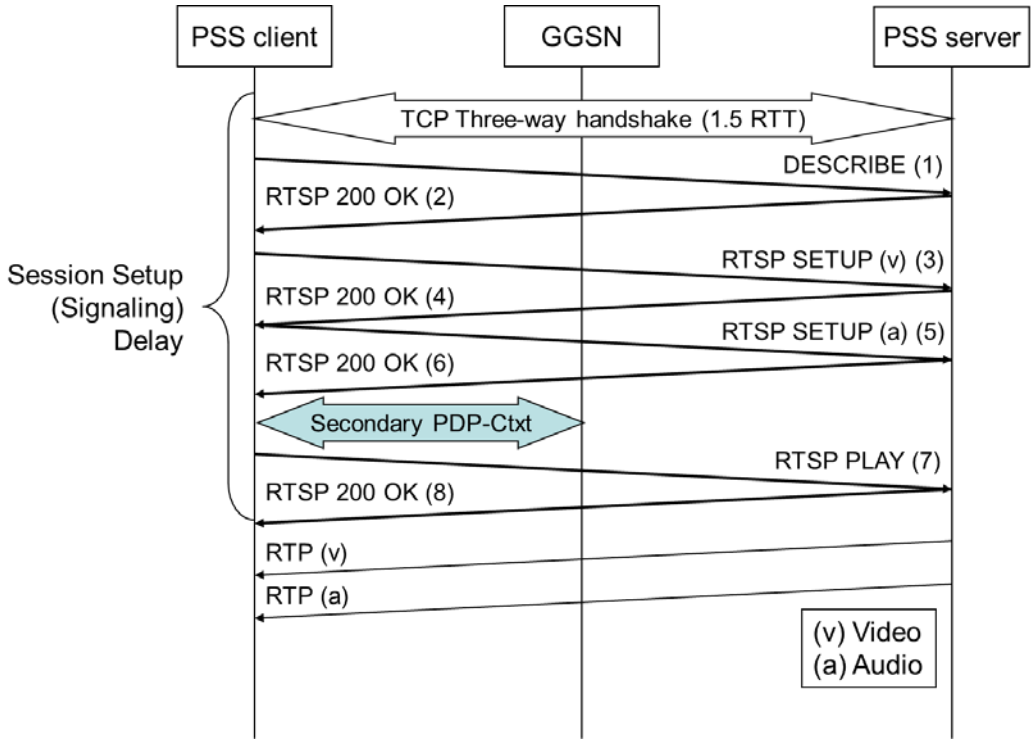

Figure 56. Baseline RTSP/PSS session setup without optimizations.

Message structure in the above case follows.

1. RTSP DESCRIBE request.

```
DESCRIBE rtsp://example.com/twister.3gp RTSP/2.0
CSeq: 1
User-Agent: PhonyClient/1.2
```

2. RTSP 200 OK response.

```
RTSP/2.0 200 OK
CSeq: 1
Server: PhonyServer/1.0
Date: Fri, 20 Dec 2013 10:20:32 +0000
Content-Type: application/sdp
Content-Length: 271
Content-Base: rtsp://example.com/twister.3gp/
Expires: Fri, 20 Dec 2013 12:20:32 +0000

v=0
o=- 2890844256 2890842807 IN IP4 198.51.100.5
s=RTSP Session
i=An Example of RTSP Session Usage
e=adm@example.com
c=IN IP4 0.0.0.0
a=control: *
a=range:npt=00:00:00-00:10:34.10
t=0 0
m=video 0 RTP/AVP 26
```

```
a=control: trackID=1
m=audio 0 RTP/AVP 0
a=control: trackID=4
```

3. RTSP SETUP request (video).

```
SETUP rtsp://example.com/twister.3gp/trackID=1 RTSP/2.0
CSeq: 2
User-Agent: PhonyClient/1.2
Require: play.basic
Transport: RTP/AVP;unicast;dest_addr=":8000"/":8001"
Accept-Ranges: npt, smpte, clock
```

4. RTSP 200 OK response

```
RTSP/2.0 200 OK
CSeq: 2
Server: PhonyServer/1.0
Transport: RTP/AVP;unicast; ssrc=93CB001E;
           dest_addr="192.0.2.53:8000"/"192.0.2.53:8001";
           src_addr="198.51.100.5:9000"/"198.51.100.5:9001"
Session: OccldOFFq23KwjYpAnBbUr
Expires: Fri, 20 Dec 2013 12:20:33 +0000
Date: Fri, 20 Dec 2013 10:20:33 +0000
Accept-Ranges: npt
Media-Properties: Random-Access=0.02, Immutable, Unlimited
```

5. RTSP SETUP request (audio).

```
SETUP rtsp://example.com/twister.3gp/trackID=4 RTSP/2.0
CSeq: 3
User-Agent: PhonyClient/1.2
Require: play.basic
Transport: RTP/AVP;unicast;dest_addr=":8002"/":8003"
Session: OccldOFFq23KwjYpAnBbUr
Accept-Ranges: npt, smpte, clock
```

6. RTSP 200 OK response.

```
RTSP/2.0 200 OK
CSeq: 3
Server: PhonyServer/1.0
Transport: RTP/AVP;unicast; ssrc=A813FC13;
           dest_addr="192.0.2.53:8002"/"192.0.2.53:8003";
           src_addr="198.51.100.5:9002"/"198.51.100.5:9003";
Session: OccldOFFq23KwjYpAnBbUr
Expires: Fri, 20 Dec 2013 12:20:33 +0000
Date: Fri, 20 Dec 2013 10:20:33 +0000
Accept-Range: NPT
Media-Properties: Random-Access=0.8, Immutable, Unlimited
```

7. RTSP PLAY request.

```
PLAY rtsp://example.com/twister.3gp/ RTSP/2.0
CSeq: 4
User-Agent: PhonyClient/1.2
```

```
Range: npt=30-
Seek-Style: RAP
Session: OccldOFFq23KwjYpAnBbUr
```

8. RTSP 200 OK response.

```
RTSP/2.0 200 OK
CSeq: 4
Server: PhonyServer/1.0
Date: Fri, 20 Dec 2013 10:20:34 +0000
Session: OccldOFFq23KwjYpAnBbUr
Range: npt=30-634.10
Seek-Style: RAP
RTP-Info: url="rtsp://example.com/twister.3gp/trackID=4"
  ssrc=0D12F123:seq=12345;rtptime=3450012,
    url="rtsp://example.com/twister.3gp/trackID=1"
  ssrc=4F312DD8:seq=54321;rtptime=2876889
```

Now we will compare the above message sequence and contents with the case when Early SETUP, RTSP Pipelining and SDP Templates are used.

First of all, we will assume that the following SDP template is stored at the client (i.e.: these fields are stable across all encoded media content by a given service provider).

```
v=0
o=
s=RTSP Session
i=An Example of RTSP Session Usage
e=adm@example.com
c=
a=control: *
a=
t=0 0
m=
a=control: trackID=1
a=
m=
a=control: trackID=4
a=
```

In summary, the above SDP file indicates that several informative SDP fields are not modified across different media sessions. Furthermore, this content provider seems to always use trackID 1 for video delivery and trackID 4 for audio delivery.

When all the optimizations we have proposed in chapter 4 are implemented, the resulting session setup procedure looks is depicted below (compare the below diagram with Figure 56). Note that this scenario is the one described in section 4.6.7.

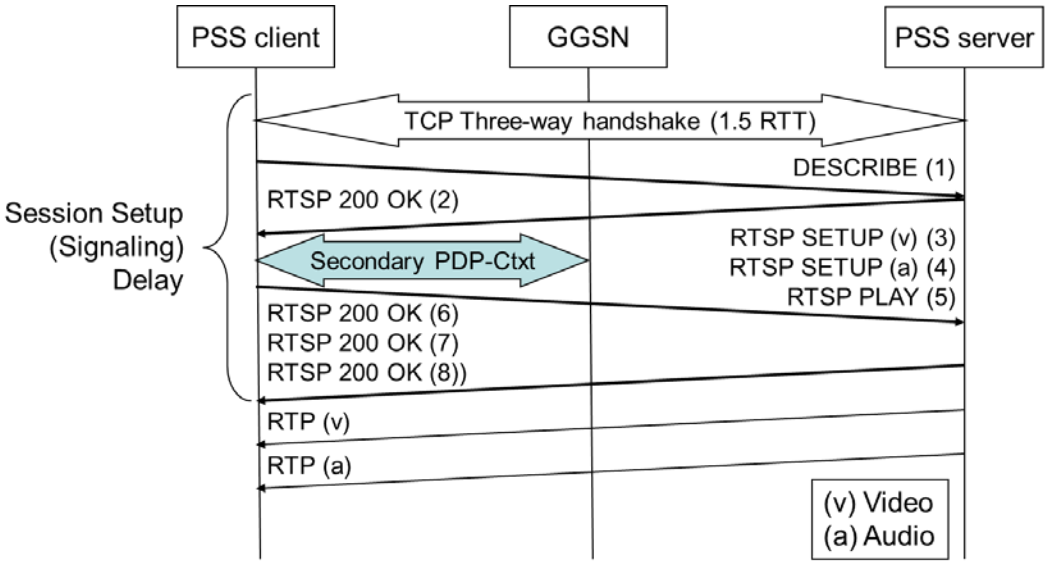

Figure 57. RTSP / PSS session setup when Early Setup, RTSP Pipelining and SDP Templates are used.

In this case the message structure is presented below (fields marked in bold are the ones that relate to the optimizations proposed by the author):

1. RTSP DESCRIBE request.

```
DESCRIBE rtsp://example.com/twister.3gp RTSP/2.0
CSeq: 1
User-Agent: PhonyClient/1.2
Supported: early-setup; sdp-template
```

2. RTSP 200 OK response.

```
RTSP/2.0 200 OK
CSeq: 1
Server: PhonyServer/1.0
Supported: early-setup; sdp-template
Session: OccldOFFq23KwjYpAnBbUr
Date: Fri, 20 Dec 2013 10:20:32 +0000
Content-Type: application/sdp
Content-Length: 271
Content-Base: rtsp://example.com/twister.3gp/
Expires: Fri, 20 Dec 2013 12:20:32 +0000

v=
o=- 2890844256 2890842807 IN IP4 198.51.100.5
s=
i=
e=
c=IN IP4 0.0.0.0
a=
a=range:npt=00:00:00-00:10:34.10
t=
m=video 9000 RTP/AVP 26
a=ssrc : A813FC13
```

```
a=
m=audio 9002 RTP/AVP 0
a=ssrc : A813FC13
a=
```

3. RTSP SETUP request (video).

```
SETUP rtsp://example.com/twister.3gp/trackID=1 RTSP/2.0
CSeq: 2
User-Agent: PhonyClient/1.2
Supported: early-setup; sdp-template
Session: OccldOFFq23KwjYpAnBbUr
Require: play.basic
Transport: RTP/AVP;unicast;dest_addr=":8000"/":8001"
Accept-Ranges: npt, smpte, clock
```

4. RTSP SETUP request (audio).

```
SETUP rtsp://example.com/twister.3gp/trackID=4 RTSP/2.0
CSeq: 3
User-Agent: PhonyClient/1.2
Supported: early-setup; sdp-template
Session: OccldOFFq23KwjYpAnBbUr
Require: play.basic
Transport: RTP/AVP;unicast;dest_addr=":8002"/":8003"
Session: OccldOFFq23KwjYpAnBbUr
Accept-Ranges: npt, smpte, clock
```

5. RTSP PLAY request.

```
PLAY rtsp://example.com/twister.3gp/ RTSP/2.0
CSeq: 4
User-Agent: PhonyClient/1.2
Supported: early-setup; sdp-template
Session: OccldOFFq23KwjYpAnBbUr
Range: npt=30-
Seek-Style: RAP
```

6. RTSP 200 OK response to the SETUP (v) request.

```
RTSP/2.0 200 OK
CSeq: 2
Server: PhonyServer/1.0
Transport: RTP/AVP;unicast; ssrc=A813FC13;
           dest_addr="192.0.2.53:8002"/"192.0.2.53:8003";
           src_addr="198.51.100.5:9002"/"198.51.100.5:9003";
Session: OccldOFFq23KwjYpAnBbUr
Expires: Fri, 20 Dec 2013 12:20:33 +0000
Date: Fri, 20 Dec 2013 10:20:33 +0000
Accept-Ranges: npt
Media-Properties: Random-Access=0.02, Immutable, Unlimited
```

7. RTSP 200 OK response to the SETUP (a) request.

```
RTSP/2.0 200 OK
CSeq: 3
Server: PhonyServer/1.0
Transport: RTP/AVP;unicast; ssrc=A813FC13;
           dest_addr="192.0.2.53:8002"/"192.0.2.53:8003";
           src_addr="198.51.100.5:9002"/"198.51.100.5:9003";
Session: OccldOFFq23KwjYpAnBbUr
Expires: Fri, 20 Dec 2013 12:20:33 +0000
Date: Fri, 20 Dec 2013 10:20:33 +0000
Accept-Range: NPT
Media-Properties: Random-Access=0.8, Immutable, Unlimited
```

8. RTSP 200 OK response to the PLAY request.

```
RTSP/2.0 200 OK
CSeq: 4
Server: PhonyServer/1.0
Date: Fri, 20 Dec 2013 10:20:34 +0000
Session: OccldOFFq23KwjYpAnBbUr
Range: npt=30-634.10
Seek-Style: RAP
RTP-Info: url="rtsp://example.com/twister.3gp/trackID=4"
  ssrc=0D12F123:seq=12345;rtptime=3450012,
    url="rtsp://example.com/twister.3gp/trackID=1"
  ssrc=4F312DD8:seq=54321;rtptime=2876889
```

Note that while server UDP ports are provided in the 200 OK response to the DESCRIBE request (when applying Early Setup) nothing prevents the Client and the Server to still populate full RTSP `Transport` headers in the SETUP transactions as well. This can be done for backwards compatibility and particularly when intermediate firewalls or ALGs are used.

Observe also how in the above example we have expanded the amount of provided SDP information in the response to the DESCRIBE request in order to incorporate the RTP SSRC field that can be carried in the `Transport` header when using standard SETUP-based stream configuration.

Annex B. 3GPP Pipelined-Requests references from 3GPP TS 26.234 (Release-7)

Excerpt from section 5.5.3 from 3GPP TS 26.234:

5.5.3 Start-up

In order to improve start-up times, a client may pipeline all necessary SETUP requests and the PLAY request. This allows streaming to begin with a single RTSP round trip if the client already has the SDP (or other adequate content description), or two round trips if it needs to first perform a DESCRIBE in order to receive the necessary information.

If the client intends to send upstream packets to ensure correctly open firewalls (also called port punching packets), then the client should not send a PLAY request until all SETUP responses are received. Pipelining of SETUP requests is still possible in this case.

If the client uses RTSP DESCRIBE to fetch the SDP from the server, then the client shall probe the server capabilities as described in clause 5.5.2.2 using the feature-tag value "3gpp-pipelined".

The client shall add the RTSP "Require" header to all but the first pipelined RTSP SETUP request with the value "3gpp-pipelined". Note that the first RTSP SETUP request shall not use a "Require" header. This will allow the PSS client to interoperate with minimal impact with older servers that do not support this feature.

Since the session does not yet exist when these pipelined messages are sent, a request header is defined which allows the client to inform the server that these messages are to be carried on the same session once it is created. Clients wishing to use pipelined start-up must implement the "Pipelined-Requests" header in order to signal the session grouping to the server.

The syntax of the "Pipelined-Requests" header is defined in ABNF [53] as follows:

Pipe-Hdr = "Pipelined-Requests" COLON startup-id

startup-id = 1*8DIGIT

The client should monitor whether the server behaves as declared.

A client unique "startup-id" is required until the client receives the session ID. The "startup-id" is unique for a particular TCP connection. Pipelined requests using this header must be sent on the same TCP connection. The method through which this ID is generated is to be decided by the client.

Annex C. Summary of Published Work

C.1 Books

- A. Rebeiro-Hargrave, D. Viamonte. Multimedia Group Communication. Push-to-Talk over Cellular, Presence and List Management Concepts and Applications. John Wiley & Sons publishing (ISBN: 978-0470058534). February 2008 [109].

C. 2 International Journals

- D. Viamonte, A. Calveras. A Distributed Man-Machine Dispatching Architecture for Emergency Operations Based on 3GPP Mission Critical Services. IEEE Access (indexed in JCR). December 2017 [17].
- D. Viamonte, A. Calveras, J. Paradells, C. Gómez Montenegro. Evaluation and optimization of Session Setup Delay for Streaming Services over 3G Networks with Quality-of-Service Support. Wiley Wireless Communications and Mobile Computing (WCMC'08) (indexed in JCR). May 2008 (first published online, September 2006) [18].
- C. Gómez, M. Catalán, D. Viamonte, J. Paradells, A. Calveras. Web browsing optimization over 2.5G and 3G: end-to-end mechanisms Vs usage of performance enhancing proxies. Wiley Wireless Communications and Mobile Computing (WCMC'07) (indexed in JCR). June 2007 [22].

C. 3 International Conferences

- E. Garcia, R. Vidal, D. Viamonte, J. Paradells. Achievable Bandwidth Estimation for Stations in Multi Rate IEEE 802.11 WLAN Cells. IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM'2007). June 2007 [23].
- D. Viamonte, A. Calveras, J. Paradells, C. Gómez Montenegro. Requirements for the Optimization of Session Setup Delay and Service Interactivity in Streaming Services over 3G. Proceedings of the European Wireless Conference (EW2006). April 2006 [46].
- C. Gómez, M. Catalán, D. Viamonte, J. Paradells, A. Calveras. Internet traffic analysis and optimization over a precommercial live UMTS network. IEEE Vehicular Technology Conference Spring (VTC Spring'05). April 2005 [21].
- M. Catalán, C. Gómez, D. Viamonte, J. Paradells, A. Calveras, F. Barceló. TCP/IP analysis and optimization over a precommercial live UMTS network. IEEE Wireless Communications and Networking Conference (WCNC'05). March 2005 [20].

- D. Viamonte, A. Calveras. Session Based Admission Control Strategy for Streaming Services over All-IP 3G Networks. Proceedings of the 15th IEEE International Workshop on Personal, Indoor and Mobile Radio Communications (PIMRC'04). September 2004 [16].

C. 4 Workshops

- D. Viamonte. Streaming over 3GPP wireless networks. Challenges and Issues. Proceedings of the Workshop on Internet Usage over 2.5G and 3G. IST-2001-92125. March 2003 [140].

C. 5 Patents

- D. Viamonte, Presence System and a Method for providing a Presence System, US Patent US20090319655 A1, June 2008 (Granted: 2015) [19].