HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Communications Laboratory

Valter Rönnholm

Push-to-Talk over Bluetooth

This master's thesis has been submitted for official examination for the degree of Master of Science

Supervisor	**Prof. Sven-Gustav Häggman**
Instructor	**M.Sc. Matias Järnefelt**

Espoo 18.1.2005

HELSINKI UNIVERSITY OF TECHNOLOGY

Abstract of Master's Thesis

Author:	Valter Rönnholm		
Title of thesis:	Push-to-Talk over Bluetooth		
Date:	January 18, 2005	**Number of pages:**	94
Department:	Department of Electrical and Communications Engineering		
Professorship:	Communications Laboratory		
Supervisor:	Sven-Gustav Häggman, Prof.		
Instructor:	Matias Järnefelt, M.Sc.		

Push-to-Talk over Cellular (PoC) is an emerging technology enabling a walkie-talkie-like service over GPRS. At the time of writing, an open standard for PoC is being specified by the Open Mobile Alliance (OMA). As specified by the OMA standard drafts, PoC is based on an IP/UDP/RTP protocol stack and a client-server based architecture. The systems exploits the SIP signalling capabilities of the the IP Multimedia Subsystem (IMS). Group management, floor control etc are administered by the network elements of PoC.

The research problem of this thesis is: "How can mobile phone users be provided with a free-of-charge PTT-feature with PoC-like user experience by means of Bluetooth technology?" The primary objective of the study is thus to propose an outline for developing a Push-to-Talk (PTT) feature that utilizes a Bluetooth scatternet and the PAN profile for data communications. A reasonable range can be obtained with Bluetooth class 1 devices, which provide a range of up to 100 m. A subsidiary objective is to provide a description of OMA PoC and the protocols it relies upon. The description serves both as a basis for pursuing the primary objective and as a tutorial, which is suitable for students or professionals desiring to acquaint themselves with OMA PoC.

The proposed outline for Push-to-Talk over Bluetooth (PoB) comprises e.g. methods for group formation, network formation, communication, and floor control. The network formation method, which can be utilized in other applications as well, is based on creating a scatternet among a predefined set of devices and on avoiding loops. This approach enables usage of a simple broadcasting based communication method, in which the devices bridging the piconets into a scatternet act as repeaters.

A method for combining PoB and PoC is also outlined. It is intended for enabling PTT-communication with both local and distant group members over Bluetooth and GPRS respectively.

Keywords: local communication, PTT, Bluetooth, network formation, scatternet, PAN, PoC, SIP, SDP, IMS

TEKNISKA HÖGSKOLAN

Sammanfattning av diplomarbete

Skribent:	Valter Rönnholm		
Rubrik:	Push-to-Talk över Bluetooth		
Datum:	18. Januari, 2005	**Sidoantal:**	94
Avdelning:	Avdelningen för elektro- och telekommunikationsteknik		
Professur:	Telekommunikationslaboratoriet		
Övervakare:	Sven-Gustav Häggman, Prof.		
Handledare:	Matias Järnefelt, Dipl.Ing.		

Push-to-Talk over Cellular (PoC) är en teknologi som möjliggör en radiotelefonlik service över GPRS vilken väckt ökande popularitet. I skrivande stund pågår specifiering av en öppen PoC-standard inom Open Mobile Alliance (OMA). OMA planerar att baser PoC på en IP/UDP/RTP protokollstack samt en server-client-arkitektur. Systemet utnyttjar även SIP-signaleringsegenskaperna hos IP Multimedia Subsystem (IMS). PoC-nätelement handhar bl.a. gruppförvaltning och taltursfördelning.

Forskningsproblemet för denna avhandling är: ”Hur kan en PoC-liknande service erbjudas gratis åt mobiltelefonsanvändare med hjälp av Bluetooth-teknologi?” Den primära målsättningen för detta arbete är därmed att skissa upp ett förslag för hur man kunde utveckla en Push-to-Talk (PTT)-funktion som utnyttjar ett Bluetooth scatternet-nät samt PAN-profilen för att överföra data. En måttlig räckvidd kan uppnås med hjälp av Bluetooth apparater av effektklass 1 vars räckvidd kan vara t.o.m. 100 m. En sekundär målsättning är att beskriva PoC samt de protokoll PoC utnyttjar (t.ex. SIP och SDP). Denna beskrivning utgör både en utgångspunkt för att uppnå den primära målsättningen och erbjuder även en introduktion till OMA PoC som lämpar sig för både studeranden och yrkesmän.

Det uppskissade förslaget för Push-to-Talk över Bluetooth (PoB) innefattar metoder för skapande av grupper och nät, dataöverföring samt taltursfördelning. Metoden för nätskapande (som kan vara användbar även för andra ändamål) baserar sig på att skapa ett scatternet emellan apparater som tillhör en på förhand specifierad grupp av apparater samt på att undvika slingor. Detta möjliggör enkel kommunikation genom att skicka data till alla apparater inom nätet, förutsatt att de apparater som sammanbinder piconet-näten till ett scatternet fungerar som repeterare.

Ytterligare uppskissas en metod för att kombinera PoB och PoC. Avsikten med detta är att möjliggöra PTT-kommunikation med både lokalt och avlägset belägna gruppmedlemmar med hjälp av Bluetooth respektive GPRS.

Nyckelord: local communication, PTT, Bluetooth, network formation, scatternet, PAN, PoC, SIP, SDP, IMS

Acknowledgements

I wish to present my gratitude to the following people:

Professor **Sven-Gustav Häggman** for his advice on structuring and writing this thesis in his role as a supervisor. I am particularly grateful for his exceptionally serious attitude towards his duty as a supervisor and for the time he devoted for discussions in monthly meetings. This kind of devotion appears to be regrettably rare among his colleagues.

My instructor and superior, **Matias Järnefelt**, deserves my thanks for giving me the opportunity to tackle an interesting and challenging problem, whose solution may prove to have a significant impact, very independently. I am also grateful for the fact that I was allowed to focus fully on my thesis work, unlike many of my peer students.

The people who arranged the funding for my work and thus entrusted me with this project: **Eric Cronstedt**, **Janne Jormalainen**, and **Teemu Suila** (”in order of appearance” so to say). I also truly appreciate the opportunity they gave me to pursue my original initiative, which unfortunately turned out to require further research in a field clearly outside the core competence domain of both me and my employer.

Niko Eiden, **Janne Tarsa**, **Per Persson** and **Younghee Jung** were all involved in the process of generating and defining the subject for this thesis. In addition, I am thankful for the opportunity they gave me together with **Jarkko Kuntanen**, **Riitta Nieminen-Sundell** and **Jim Lundin** to get acquainted with the product development process of very exciting products. Even though we did not actually work together and I actually worked on a different site, I felt that these people supported me.

As a new technology and service, PoC is not yet very well known. PoB, on the other hand, is a totally new concept itself, which is based on Bluetooth features that are currently not commonly utilized. The number of people with whom to discuss the arising issues was thus quite limited and their time likewise. I would therefore like present special thanks to the following experts for dialogues and advice: **Javier**

Lafuente, **Jukka Alakontiola**, **Keijo Lähetkangas**, **Markku Vimpari**, **Jari Keskimaula**, **Timo Lind**, and **Kai Mustonen**.

I am very grateful to my parents, **Liisa** and **Gustaf**, for their emotional, intellectual, practical and financial support during the past 28 years. I am very grateful for bringing me up to what I have become.

Last and most I would like to thank my fiancée **Minna**, for her love.

Salo, July 3, 2004

Valter Rönnholm

P.S. I would also like to thank the people who reviewed the first complete draft of this thesis: **Thomas Müller** (who pointed out a serious flaw in the loop avoidance algorithm), **Jürgen Schnitzler** and Javier and Keijo, who are already mentioned above. I would also like to thank the Intellectual Property Rights department of my employer for a thorough but time-consuming evaluation of the subject matter disclosed herein.

Turku, December 12, 2004

Contents

Abbreviations

3GPP	The 3rd Generation Partnership Project
ACK	Acknowledgement
ACL	Asynchronous Connection-less
ALG	Application Level Gateway
AM_ADDR	Active Member Address
AMR	Adaptive Multi-Rate
AMR-FR	AMR-Full Rate
AMR-HF	AMR-Half Rate
AMR-WB	Wideband AMR
AR_ADDR	Access Request Address
ARP	Address Resolution Protocol
ARQ	Automatic Repeat Request
ASCII	American Standard Code for Information Interchange
BD_ADDR	Bluetooth Device Address
BER	Bit Error Rate
BNEP	Bluetooth Network Encapsulation Protocol
BSR	Bluetooth Scatternet Routing
BSS	Base Station System
CAC	Channel Access Codes
CDMA	Code Division Multiple Access
CMR	Codec Mode Request
CRC	Cyclic Redundancy Check
CS	Circuit Switched
CSCF	Call State Control Function
CSD	Circuit Switched Data
DAC	Device Access Codes
DH	Data - High Rate
DM	Data - Medium Rate
DNS	Domain Name Server
EDGE	Enhanced Data Rate for GSM Evolution
eSCO	Extended Synchronous Connection-Oriented (Link)
ESCO	Extended SCO
FO	Follow On
FQI	Frame Quality Indicator
FT	Frame Type
GERAN	GSM/EDGE Radio Access Network
GGSN	Gateway GPRS Support Node
GN	Group Ad-hoc Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GTP	GPRS Tunneling Protocol
HCI	Host Controller Interface
HLR	Home Location Register
HSCSD	High Speed Circuit Switched Data
HTTP	Hypertext Transfer Protocol

IANA	Internet Assigned Numbers Authority
I-CSCF	Interrogating CSCF
iDEN	Integrated Dispatch Enhanced Network
IEEE	The Institute of Electrical and Electronics Engineers, Inc
IETF	Internet Engineering Task Force
IF1	Interface Format 1 (AMR)
IF2	Interface Format 2 (AMR)
IM	IP Multimedia
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPv4	IP version 4
IPv6	IP version 6
IrDA	Infrared Data Association
ISC	IP Multimedia Subsystem Service Control Interface
ISM	Industrial, Scientific, and Medical (radio frequency band)
kbps	kilobits per second
L2CAP	Logical Link Control and Adaptation Protocol
LMP	Link Manager Protocol
ME	Mobile Equipment
ME	Management Entity
MGCF	Media Gateway Control Function
MGW	Media Gateway
MMD	(All-IP Core Network) Multimedia Domain
MMS	Multimedia Messaging Service
MOS	Mean Opinion Score
MRF	Multimedia Resource Function
MS	Mobile Station
NAP	Network access points
OMA	Open Mobile Alliance
PAN	Personal Area Network
PANU	PAN User
PC	Personal Computer
P-CSCF	Proxy CSCF
PDN	Packet Data Network
PLMN	Public Land Mobile Network
PM_ADDR	Parked Member Address
PoB	Push-to-Talk over Bluetooth
PoC	Push-to-Talk over Cellular
PS	Packet Switched
QoE	Quality of Experience
QoS	Quality of Service
RFC	Request for Comments
RTCP	RTP Control Protocol
RTP	Real-Time Transport Protocol
RTS	Request-to-Send
RTT	Radio Transmission Technology

SCO	Synchronous Connection Oriented
S-CSCF	Serving CSCF
SDeP[1]	Session Description Protocol
SDiP[1]	Service Discovery Protocol
SGSN	Serving GPRS Support Node
SIG	Special Interest Group
SIP	Session Initiation Protocol
SMS	Short Message Service
SNDCP	Sub-Network Dependent Convergence Protocol
TCH	Traffic Channel
TCP	Transfer Control Protocol
TE	Terminal Equipment
TS	Technical Specification
UART	Universal Asynchronous receiver Transmitter
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URI	Universal Resource Identifier
USB	Universal Serial Bus
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP
WAP	Wireless Application Protocol

Symbols

C	Required Capacity
c	Number of (connected) devices in a scatternet
f	RTP Packet Rate
S_P	RTP Payload Size
S_F	AMR IF2 Frame Size Within RTP
s	Number of slaves in a piconet
x	Number of AMR Frames per RTP Packet

[1] In general, the acronym SDP is used to denote both the Service Discovery Protocol of Bluetooth and the Session Description Protocol. However, since this thesis discusses both of these the acronyms SDeP and SDiP are used for clarity.

1 INTRODUCTION

1.1 Background

A rather new service called Push-to-talk over cellular (PoC) is gaining interest in the field of mobile cellular communications. Within mobile wireless telecommunications, the method in which one person's speech is conveyed to a group of people in a half duplex fashion is ancient. However, PoC rejuvenates this capability by integrating it into e.g. GSM (Global System for Mobile Communications) terminals, which traditionally provide dial-up speech communications. In addition to combining these two capabilities traditionally associated with separate devices and technologies, PoC could some day provide a virtually global walkie-talkie type communications service.

Cellular operator's push-to-talk (PTT) services based on Motorola's proprietary standard, Integrated Dispatch Enhanced Network (iDEN), have gathered a significant number of customers in the U.S.A and some Latin American countries. This success has naturally drawn the attention of other players in the field and several competing proprietary solutions are emerging. These solutions are generally based on VoIP (Voice over Internet Protocol) over GPRS (General Packet Radio Service) in GSM or over 1xRTT (Radio Transmission Technology) in CDMA2000 (Code Division Multiple Access). It has been estimated that the number of PTT subscribers in USA and Europe will increase from 11.8 million in 2003 to 115 million in 2009. However, interoperability between different operators and vendors has been predicted to be a critical success factor for PTT in general. [1] The Open Mobile Alliance (OMA) is currently drafting an open PoC standard with the aim to ensure interoperability over a variety of mobile access networks [2]. If PoC becomes a success and the OMA PoC standard becomes the prevailing standard worldwide, millions of users will be accustomed to the user interface and user experience of OMA PoC. Therefore, this thesis assumes the requirements document of the OMA PoC draft standard as a baseline for user experience.

At the time of writing many of the OMA PoC technical specification documents are still being drafted and many of the details related to the drafting process are unpublished. However, public material seems to suggest a VoIP-based approach using a protocol stack incorporating the Internet Protocol (IP), User Datagram Protocol (UDP), and Real-time Transfer Protocol (RTP).

In the somewhat different field of short-range wireless communications, Bluetooth is becoming the preferred local connectivity technology in mobile phones, laptops and other devices. The sales of mobile phones with Bluetooth connectivity has increased from 4.1 million in 2001 to 27 million in 2003 and according to an estimate the number will increase to 490 million in 2009 [3]. The standard is being continuously developed further by Bluetooth SIG, which adopted the Personal Area Networking (PAN) profile in February 2003. From the perspective of superior protocol layers and applications, the PAN profile gives Bluetooth connections the appearance of an Ethernet-network. IP-based applications can thus easily exploit Bluetooth connectivity without remarkable adaptations or modifications.

Among the public, Bluetooth connectivity is currently associated with a range of approx. 10 meters. The reason for this is that the majority of commercially available end-user devices use the middlemost power class of the three classes defined by the Bluetooth core specification. However, Bluetooth devices and modules with power class 1, which provides a range of approx. 100 meters, are currently commercially available as well.

The above developments lead to the notion of providing a free-of-charge, short range PTT-functionality over Bluetooth, which is the subject of this thesis.

1.2 Related Work

Kargl et al [4] discuss conveying point-to-point voice over Bluetooth. Particularly, a routing protocol called Bluetooth Scatternet Routing (BSR) is proposed for routing voice data over multiple hops on Synchronous Connection Oriented (SCO) links. The proposed protocol is similar to Ad-Hoc On-Demand Distance Vector Routing (AODV) [5] and Dynamic Source Routing (DSR) [6] and sets up a scatternet on demand by flooding Path Request messages over Asynchronous Connection-less (ACL) links to devices in the vicinity. If a request message reaches the desired recipient, it sends a Path Found message over the discovered path to the originator. In contrast to AODV and DSR, BSR does not necessarily find the shortest path but settles for the first discovered path, as it probably is one of the shortest. As soon as the first path is found, the flooding algorithm is aborted with Path-End messages. The reason for this approach is that the delay induced by the setting up of the ACL connections, increases the overall delay for finding the path. The algorithm is thus expedited by reducing the number of ACL links needed. Once a path is found, SCO links may be set up for voice traffic.

In their evaluation of BSR, Kargl et al find that the end-to-end connection setup times are long. This is bound to produce problems in the case of a broken link whereby an alternative path is needed. Some optimization are outlined for remedying this problem. In addition, Kargl et al state that the end-to-end connection setup times become quite large when the number of nodes is increased. As an example, the average delay for finding a path in an environment with 20 or 55 nodes is 5 s and 10 s respectively. Kargl et al also state, that a number of changes to the Bluetooth 1.1 specifications are needed to make BSR efficient.

Even though the Bluetooth 1.2 specification halves both the Inquiry and Paging times [7], the delay for finding a path with BSR seems to be quite long, particularly in an environment with a dense penetration of devices supporting BSR. Even though it might be acceptable to have a rather long delay for the initial connection setup, the procedure for replacing broken links with alternative paths is unacceptably slow. In addition, if BSR would be used to find paths to several destinations for a one-to-many PTT message, the complexity of the procedure would be significantly increased and thus also the requirements for memory and processing power.

1.3 Research Problem

When an enabling technology gains popularity and reaches significant market penetration, an obvious consequence is the aspiration to develop innovative features and services based on the technology in question. In many cases, it is more lucrative to enhance existing and proven features and services than to develop and nurture totally new ones. Enhancing PTT by providing it for free over Bluetooth seems to be a promising alternative for exploiting the opportunity emerging by the rising popularity of Bluetooth. The range of Bluetooth obviously constrains the PTT functionality to a significantly smaller area than e.g. PoC. However, taking into account that most oral communication takes place in local context, this may not be an as drastic disadvantage as first perceived. Two or more people located at a distance of say 50 to 100 meters from each other may easily communicate without any aids, if the environment is noiseless and no bystanders need to be taken into account. However, if the environment is noisy or crowded, discussing at this distance may not be very convenient. For example, if there is background noise such as loud discussions, music, traffic noise or wind, discussing at these distances may be difficult or even impossible. On the other hand, a loud "long-distance" discussion in a somewhat quieter environment, such as a department store, crowded outdoor area or an open-plan office, may be considered as indiscrete or rude, or may attract unpleasant attention.

The general research problem of this thesis is:

- How can mobile phone users be provided with a free-of-charge PTT-feature with PoC-like user experience by means of Bluetooth technology?

1.4 Focus and Objectives

The objective of this thesis is twofold. The first objective is to thoroughly study the current publicly available draft version of the OMA PoC standard and to describe it to a sufficient extent to allow a reader to obtain a comprehensive general understanding of PoC. Particular emphasis is placed on the matters affecting the user experience but the underlying technical architecture is thoroughly described as well, in order to provide a comprehensive view of the capabilities and restraints. Therefore other relevant standards such as the IP Multimedia Subsystem (IMS) as defined by 3GPP, Session Initiation Protocol (SIP) and Session Description Protocol (SDeP)[1] are also studied and described to the relevant extent. However, details that

[1] In general, the acronym SDP is used to denote both the Service Discovery Protocol of Bluetooth and the Session Description Protocol. However, since this thesis discusses both of these the acronyms SDeP and SDiP are used for clarity.

are irrelevant to the research problem of this thesis, such as billing and its realization, are omitted. The motivation for this first objective is that fulfilling the second objective of this thesis requires a thorough understanding of PoC and its technical implementation. In addition, hardly any material discussing PoC in a technologically holistic manner is currently available. Actually, the first book focusing on IMS, which is an enabling framework for PoC, was published as late as June 2004.

The second objective of this thesis is to specify how Bluetooth can be used to enable a free of charge PoC-like feature. The fact that it is not possible to achieve all the functionalities and capabilities of PoC in a Bluetooth-based solution is appreciated. Therefore some of the PoC functionalities are curtailed and some drawbacks are accepted. The PAN profile of Bluetooth is chosen as a platform for the specification and possible implementation. The reason for this selection is that PoC is designed for an IP network, which seems to imply that adapting the PoC protocol stack onto the IP-enabling PAN profile should be relatively straightforward compared to other alternatives.

Security issues are important whenever telecommunication systems are designed and specified. However, the level of security is always a question of policy. Traditionally, basic walkie-talkies have been associated with no security whatsoever. The extreme alternative would thus be to accept the possibility of eavesdropping and to communicate this flaw to the user. In any case, the required security level depends on many product marketing related factors. In addition, means for providing different levels of security have been incorporated in Bluetooth and it thus seems inefficient to include security mechanisms above the Bluetooth protocol stack. Because of these reasons, outlining a security policy for PTT over Bluetooth (PoB) and considering implementation of security is outside the scope of this thesis. Nevertheless, security must be considered when implementing the system outlined in this thesis.

Speech coding also plays an important role since its characteristics, such as capacity requirements, error tolerance and processing requirements, set requirements on the underlying protocols and thus affect the system design significantly. In order to limit the scope of this thesis, the codec selection issue is not discussed in detail.

1.5 Methodology

The starting points of this thesis are the current publicly available draft version of the OMA PoC standard and the Bluetooth specification, particularly the PAN profile. The OMA PoC standard drafts set the user experience and user interface targets for PoB. However, when the PTT functionalities defined by OMA PoC are adapted to the Bluetooth PAN environment, conflicts occur between the restrictions of Bluetooth and the requirements of PoC. In this thesis the problems arising from these contradictions are identified and possible solutions are outlined and evaluated. On the basis of these evaluations, the solutions perceived as the preferred alternatives are selected. When discussing the alternatives and the selected solution to a problem, the rejected alternatives, their pros and cons, and the reasoning behind the decision are discussed as well. Altogether, the selected alternatives constitute a proposal for implementing a PTT-feature according to the research problem and objective of this thesis.

1.6 Thesis Outline

This thesis is divided into two parts, a theoretical review and a proposed outline for implementing PoB. The theoretical review constitutes chapters 2–6 and the proposal chapters 7 and 8.

The second chapter of this thesis briefly discusses IP in GPRS networks at a general level, because it is the foundation PoC is built upon. In addition, IMS is also described because the OMA PoC standard assumes it as a framework for signalling.

Chapter 3 gives an introduction to the Session Initiation Protocol (SIP), because PoC signalling is based on SIP. SDeP is also presented at a general level, since its usage is essential in SIP.

Chapter 4 of this thesis gives a thorough overview of the current version of the OMA PoC draft standard and the related standards to the extent they are relevant to the research problem of this thesis.

Chapter 5 describes Bluetooth and particularly the PAN profile thereof. The focus is on the PAN profile and the core specification is only discussed to the extent necessary for understanding the functioning of the PAN profile.

In Chapter 6, the high level requirements and use cases for PTT over Bluetooth are set out. These are based on the corresponding use cases and requirements specified in the requirements document of the OMA PoC standard draft.

In Chapter 7, the general level conflicts and problems arising from deploying a PoC-like PTT-feature over Bluetooth PAN are identified and confronted. The chapter outlines a proposal for the implementation of PoB. The problems arising on different levels are analyzed and possible solutions are outlined and evaluated. On the basis of the evaluations, one solution alternative for each of the identified problems is selected as the preferred solution and the reasoning behind the selection is rationalized.

Chapter 8 discusses an optional enhancement of PoB, namely combining the feature described in Chapter 7 with conventional PoC. With a hybrid solution of PoC and PoB, a feature with both the free-of-charge benefit of Bluetooth and the range benefit of GPRS can be reached.

Finally, Chapter 9 concludes this thesis. The thesis is summarized, conclusions are made on the outcome of this thesis and possible subjects for further study are outlined.

2 IP IN GPRS NETWORKS

This section describes the well known GPRS system of GSM and the IP Multimedia Subsystem (IMS), since they provide the basis that OMA PoC relies upon. The purpose of this chapter is to give a general understanding of these systems to readers who are less acquainted with them. Readers with a good knowledge of GPRS and IMS can readily skip this chapter. Those with a good understanding of GPRS but with less knowledge of IMS are encouraged to read Section 2.3.

2.1 HSCSD and GPRS

PLMN networks have evolved from a circuit switched telephony oriented architecture towards an IP based packet switched architecture. Mobile networks where originally designed for conveying speech but the sustainable trend has been an increasing share of data traffic. As the amount of data traffic is taking over the amount of speech traffic, the tendency is to move from a circuit switched network that enables packet switched services towards a packet switched network that provides circuit switched services.

The original GSM Phase 1 standard provided a circuit switched data (CSD) service utilizing at a data rate of 9.6 kbps, which was later increased to 14.4 kbps by modifying the channel coding and protocols. CSD utilizes one GSM traffic channel (TCH) and, as the name implies, reserves this channel for the entire duration of the circuit switched connection regardless of whether data is transported or not. [8]

As the need for mobile data communications increased, the GSM release 1996 standard introduced the high-speed circuit switched data (HSCSD) service at a maximum data rate of 64 kbps. HSCSD uses up to eight TCH channels and provides a maximum data rate of 64 kbps. GSM Release 1997 introduced a packet switched data service for GSM, i.e. GPRS. GPRS uses 1-8 TCH channels dynamically depending on the instantaneous data transfer capacity requirement. The theoretical maximum capacity of GPRS is 160 kbps, if no error correction is used. One objective of GPRS is to provide mobile stations with access to IP networks. [9]

An overview of the GPRS logical architecture is presented in Figure 1. In the figure,

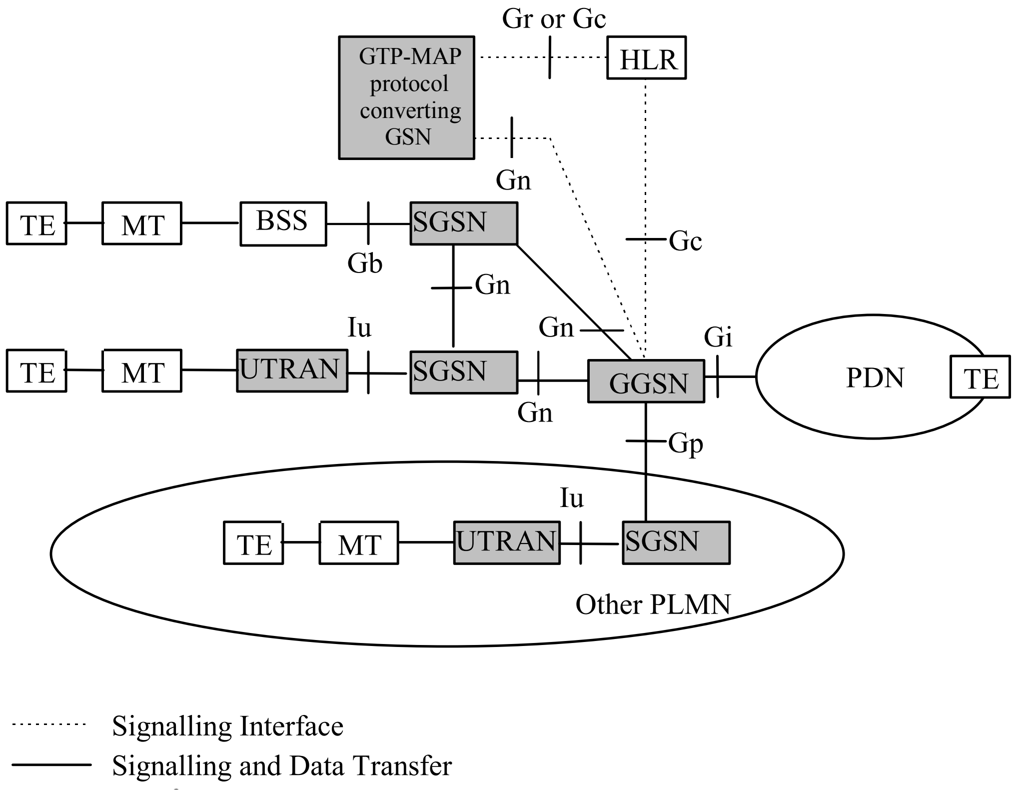

Figure 1: Overview of the GPRS logical architecture [10]

a mobile station (MS) is presented as its subelements terminal equipment (TE) and mobile termination (MT) as specified in [11]. The MS is connected to a base station system (BSS), which comprises base station controllers (BSC) and base transceiver stations (BTS) (neither depicted). The BSS is further connected to a serving GPRS support node (SGSN). SGSN:s perform delivery and routing of GPRS packets within a PLMN. The SGSN is further connected to another SGSN and a gateway GPRS support node (GGSN). A GGSN handles internetworking with other PLMN:s and packet data networks (PDN) such as IP networks or other packet data protocol (PDP) networks. As described below, the GGSN basically tunnels e.g. IP packets from the PDN to the correct SGSN, which relays it further to the correct MS. In order to route packets to the correct MS, the MS mobility is handled at two levels. The current BSS of an MS is stored in a SGSN and the current SGSN of an MS is stored in a GGSN and a home location register (HLR). As seen from the packet data network (PDN) e.g. IP network, the GGSN behaves like a router and hides thus the structure and function of the GPRS network. [12]

The HLR also stores user profiles comprising e.g. information regarding connections between an MS and a PDN. Such a connection is referred to as a PDP context, which is explained in more detail below. In addition, the figure depicts a UMTS terrestrial radio access network (UTRAN) in parallel with the GSM BSS and another PLMN with a UTRAN. [10]

The protocols used for transmission of IP packets between an MS and a GGSN is presented in Figure 2. As presented in the figure, IP packets are tunneled with the GPRS Tunneling Protocol (GTP) between the GGSN and the SGSN. The IP transmission between the SGSN and the MS is performed with the Sub-Network Dependent Convergence Protocol (SNDCP). SNDCP provides transparent transmission over the underlying protocol layers, which may constitute of varying subnetworks and protocols. Within the SNDCP packets, the IP header and payload data may be compressed. [14]

In order to convey data between a PDN and an MS, a PDP context is created for the MS. The PDP context is basically a binding between the PLMN address and a PDN address, which has been assigned to the MS. In addition, other characteristics of the connection, such as QoS information, can be stored as well. [14] This information is stored in a PDP context information element in the MS and in the relevant SGSN:s and GGSN:s [10].

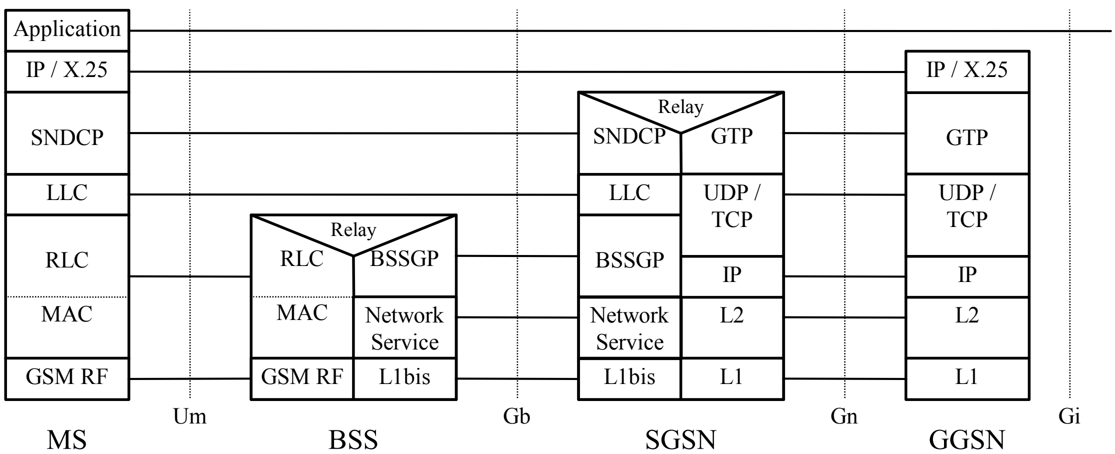

Figure 2: GPRS Transmission Plane Protocols [13]

2.2 EDGE

The performance of GPRS and HSCSD was improved in GSM Release 1999 by introducing an alternative modulation scheme, which provides higher throughput if interference is relatively low i.e. if the MS in question is located relatively close to the BS. The modifications of higher layers and protocols where minor. EDGE (Enhanced Data Rates for GSM Evolution) enhances GPRS and HSCSD to EGPRS (Enhanced GPRS) and ESCSD (Enhanced CSD). The maximum user plane bit rate of EGPRS is 473 kbps. [9]

2.3 IP Multimedia Subsystem

The 3GPP Release 5 introduces IMS, which defines an extension to the core network of the packet switched domain of a UMTS network. IMS enables SIP (Session Initiation Protocol) Signaling over IP for establishing, modifying and closing IP connections. In addition, QoS (Quality of Service) mechanisms are introduced for configuring end-to-end QoS services e.g. between a PLMN terminal and a PDN terminal [15, 16]. The relationship between the IMS and other PLMN domains is presented in Figure 3. IMS introduces three new elements: the call state control function (CSCF), the media gateway control function (MGCF) and the multimedia resource function (MRF) (not depicted). A CSCF acts basically as a SIP proxy but may also have additional functions such as QoS policy control [15]. Depending on the functionalities, CSCF:s are further categorized as proxy, serving- or interrogating CSCF:s. The MGCF controls internetworking with CS systems and the functioning of MGW:s. [17] The MRF controls and processes multimedia conferences such as multiparty calls and audio-video conferences over IP [18, 19].

The IMS also includes additional features. According to Wong et al [20], the IMS architecture enables mobile operators to gain the advantage of offering basic IP connectivity without the drawbacks traditionally associated with providing merely a bearer service. In addition to the basic IP connectivity service, IMS enables mobile operator's to offer additional services to both end users and third party service providers. From the end user perspective, the IMS architecture offers the

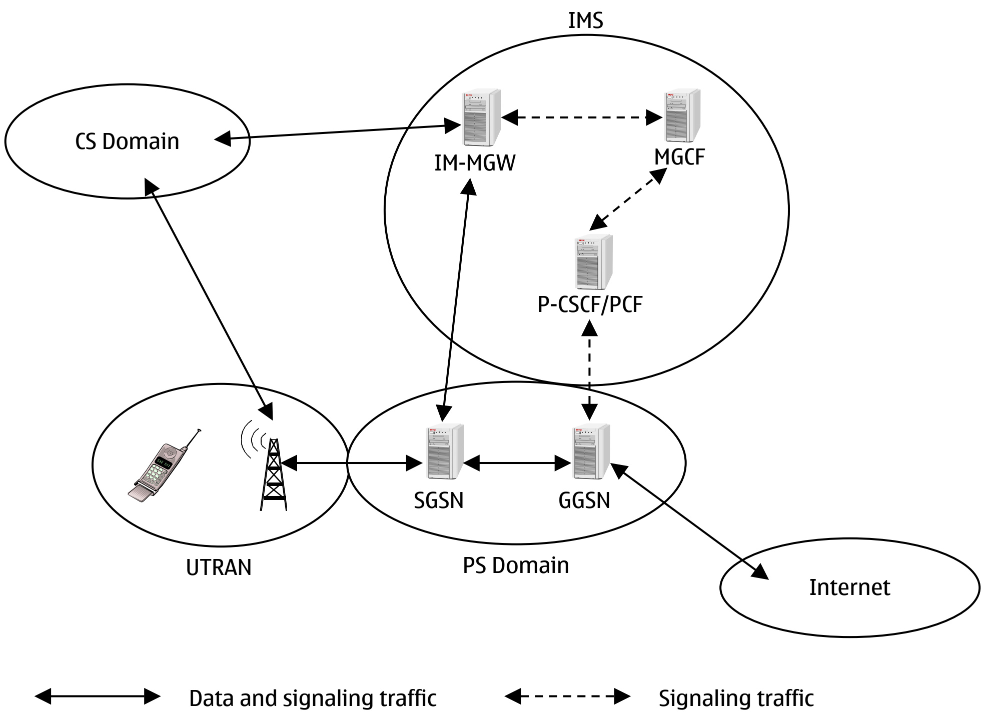

Figure 3: IP Multimedia Subsystem [17] (adapted)

alternative of accessing a basic package of IP based services. The benefit of third party service providers is that the IMS enables them to access capabilities of the network below the IP-layer, such as location and presence information.

2.4 Chapter Summary

The GPRS networks of GSM and UMTS provide a medium for transferring data over IP connections between both mobile terminals and other packet data networks as well as for peer-to-peer communication between mobile terminals. IMS adds a SIP based signalling framework for handling personal mobility i.e. allows users to be accessible at any suitable device the user may select. These two capabilities - IP data communication and SIP signalling - create a framework that provides a basis for implementing e.g. PTT services.

3 SESSION INITIATION PROTOCOL

This section is based on RFC 3261 [21] and [22], which specify and explain SIP respectively. The intention of this section is to describe the Session Initiation Protocol and the Session Description Protocol to the extent required in order to understand how PoC functions.

SIP is a protocol defined to run on top of transport level protocols such as TCP or UDP. It is intended for setting up, modifying and tearing down multimedia sessions over the Internet. Several different protocols (e.g. real-time transport protocol, RTP) have been specified for enabling various forms of real-time media transfer such as voice or video. SIP is intended to work in combination with these protocols by enabling user agents (i.e. hosts such as PC:s or mobile terminals) to discover each other and define the characteristics of a session.

An example of how a multimedia session can be initiated with SIP is presented in Figure 4. In the example, a session is established between Alice's phone and Bob's phone with the aid of SIP proxy servers of Alice's and Bob's SIP service providers. The SIP messages are chronologically numbered in the figure for clarity (the numbers in parenthesis):

1. Alice's phone sends an INVITE-message with Bob's universal resource identifier (URI) as the recipient address. The message is sent to a SIP proxy serving Alice's domain, i.e. atlanta.com. An example of an INVITE-message header is given in Figure 5, which is explained in section 3.3. The INVITE-message contains an offer according to the Session Description Protocol (SDeP). An example of an SDeP message is given in Figure 6, which is explained in section 3.4. The SDeP offer is basically a list of suggested parameter alternatives for the connection.

2. The SIP proxy of the atlanta.com domain forwards the invite message to the SIP proxy of biloxi.com, whose IP-address may e.g. be obtained by a domain name server (DNS) lookup. An additional via-field corresponding to the atlanta.com SIP proxy is added to the message header. (The newer via-field is above the older via-field.)

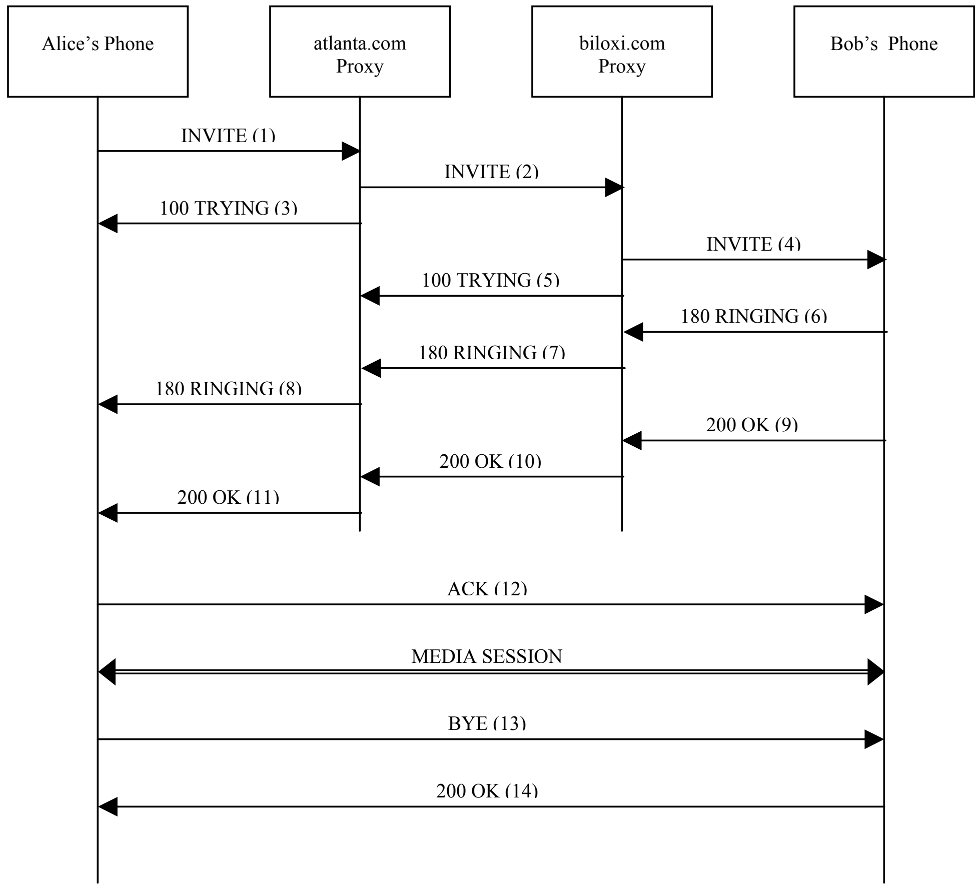

Figure 4: SIP session setup example [21] (adapted)

3. The atlanta.com SIP proxy sends a 100 TRYING-response to Alice's phone to inform that the INVITE-message was received and is being routed further.

4. The SIP proxy of the biloxi.com domain receives the INVITE-message from the atlanta.com SIP proxy, adds an additional via-field, and forwards the message to Bob's phone. The current IP-address of Bob's phone is acquired by querying a SIP registrar, which is a database for storing IP-addresses associated with users' URI-addresses. The operation of a SIP registrar is explained in section 3.1 and URI:s are discussed in section 3.2.

5. The biloxi.com SIP proxy sends a 100 TRYING-response to the atlanta.com SIP proxy to indicate that the INVITE-message was received and is being processed.

6. Bob's phone receives the INVITE-message and alerts the user e.g. with a ringing sound. A 180 RINGING-response is sent to the biloxi.com SIP proxy to indicate that the INVITE-message was received and the user is being alerted.

7. The biloxi.com SIP proxy receives the 180 RINGING-response, removes the uppermost via-field i.e. the field corresponding to itself, and forwards the response to the atlanta.com SIP proxy.

8. The atlanta.com SIP proxy receives the response, removes the uppermost via-field, and forwards the response to Alice's phone.

9.-11. When Bob has accepted the call e.g. by pressing a button, his phone sends a 200 OK-response to the biloxi.com SIP proxy. The response is forwarded to Alice's phone in a similar way as in steps 6-8. The 200 OK-response includes an SDeP message as an answer to the SDeP-message in the INVITE-message. The SDeP message indicates which of the suggested parameter sets are rejected and accepted as explained in section 3.4.

12. When Alice's phone receives the 200 OK-response, it sends an ACK message directly to Bob's phone bypassing the SIP proxies. Alice's and Bob's phones are now aware of each others IP-addresses and they have agreed on how to communicate. A media session has thus been initiated between the two phones and data can be conveyed.

13. Either party can request to disconnect the session by sending a BYE-message.

14. The BYE-message is acknowledged with a 200 OK-message, which terminates the SIP session.

3.1 SIP Registrars and Registration

A SIP registrar keeps track of at which IP addresses a user is currently reachable. This information can then be accessed by e.g. a SIP proxy when another user invites the user in question to a session. A SIP registrar thus maintains associations between URI:s and IP-addresses. Associations are created by sending REGISTER-messages to a SIP registrar. A user may register several different devices in order to define several alternative contact points for reaching the user. Similarly, several users may register the same address whereby they all are accessible at that particular device. In order to avoid e.g. call hijacking, the originator of a REGISTER-message is authenticated.

3.2 Uniform Resource Indicators

SIP specifies SIP servers that provide user agents with services for locating desired user agents and initiating sessions with them. SIP uses Uniform Resource Indicators (URI) for identifying users. The general URI format is of the following type, the fields of which are explained in Table 1:

sip:user:password@host:port;uri-parameters?headers

3.3 SIP Message Structure

SIP is based on a request-response transaction model similar to the one of HTTP. Client-server communication is initiated by a request that is replied to with a response. In the example of Figure 4, the signaling is initiated by an INVITE-message sent by Alice's phone. The INVITE-message may be e.g. as presented in Figure 5, the fields are explained in Table 2.

```
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 5: An example of a SIP INVITE-message header [21]

Table 1: SIP URI field explanations

Field	Explanation
sip	Specifies that the URI in question is a SIP URI. (Other URI:s used by SIP are e.g. "sips" for Secure SIP and "tel" for Telephone URI. The format of the former is analogous to the SIP URI while the latter differs significantly.)
user	Identifies the user in question e.g. "valter.ronnholm" or "jack". If supported by the host in question, global or local telephone numbers may also be used e.g. "3589465077". The "user:password@"-part is called the userinfo-part of the URI. This part is optional and can be omitted in which case the URI refers to the host specified in the host field in question (see below).
password	A password associated with the user. This field is optional and the SIP specification recommends that this field is not used since it causes a security risk.
host	The host providing the SIP resource to the user. Identified by a domain name such as "hut.fi" or an IPv4 or IPv6 address such as "130.233.220.31".
port	Specifies the transport level port number to which a request is to be sent e.g. "8080". This field is optional and if it is omitted, the default value "5060" is used.
uri-parameters	The URI-parameters are optional. A parameter is always preceded by a semicolon and several URI-parameters may be given by separating them with semicolons. SIP specifies six different parameters: transport, maddr, ttl, user, method and lr. For example, a "transport=udp"-parameter can be used to specify the transport protocol. The parameter mechanisms is extensible i.e. additional parameters may be freely specified and used. These will, however, be ignored by SIP elements that do not recognize them.
headers	The headers are optional. The first header is preceded by a question mark. Several headers may be included by separating them with the "&"-symbol. Headers can be used to convey additional information or settings.

Table 2: Explanations of some SIP message headers

Field	Explanation
INVITE	This is the request line, which indicates the request type (e.g. "INVITE"), the recipient's URI, and the SIP version.
Max-forwards:	This field indicates how many times the message is allowed to be forwarded.
Via:	This field indicates the entities via which the message is routed. For each hop, an additional via-field is added by e.g. the SIP proxy forwarding the message. Once the message reaches its recipient, the Via-fields indicate the route via which a response is to be conveyed.
To:	Indicates the recipient of the message.
From:	This field indicates the source of the message. The tag-parameter is used to uniquely identify the dialog, see the explanation for the Call-ID-field.
Call-ID:	A unique identification for the dialog. The Call-ID and the tags in the To- and From-fields uniquely identify the dialog. (As an exception, no tag is included in the To-field of an invite but it is assigned in the corresponding response.)
CSeq:	A sequence number for indicating the order of transactions in a dialog and for uniquely identifying the transactions. This field is also used to differentiate between new requests and retransmitted requests.
Contact:	In an INVITE-message, this field indicates the absolute URI (i.e. the direct address) of the originator. By using this URI, a response can be conveyed directly to the user in question without routing it through the proxies indicated in the via-field(s). This is useful e.g. when a session has been successfully established and direct communication between the peers begins (e.g. the Media Session-arrow in Figure 4.) The meaning of the URI in a Contact-field depends on the type of request or response it is in.
Content Type:	This field indicates the type of the content in the SIP message body. This field is used in accordance with section 3.7 of RFC 2616 (i.e. HTTP). In the example above, this field indicates that the message body is of the SDeP media type, which is a subtype of the media type "application". (SDeP is explained in section 3.4.)
Content Length:	This field indicates the length of the message body in bytes.

3.4 Session Description Protocol

This section is based on RFC 2327 [23].

The body of a SIP INVITE-message includes the characteristics of the upcoming connection. This information is typically conveyed in accordance with the Session Description Protocol (SDeP), which is specified in RFC 2327. The use of SDeP in SIP is specified in RFC 3264, which modifies SDeP slightly.

An SDeP message includes a session description part and a time description part. Optional media description parts may also be included. The mandatory fields of the session description part are v, o, s, and c (see Tables 3 and 4 for explanations). The c-field is, however, optional in the session description part, if the c-field is included in all media description parts. Only the t-field is mandatory in the time description part. An example of an SDeP description is presented in Figure 6 and the fields are explained in Tables 3 and 4.

A SIP INVITE-message typically contains an SDeP description similar to the one presented in Figure 6. This description is an offer, which presents the media type alternatives suggested by the originator. A typical response to an INVITE-message is a 200 OK-message including an SDeP description. The structure of the SDeP message in the 200 OK-message is analogous to the structure of the SDeP message in the INVITE-message. The responding SDeP message must, however, have an equal number of media description parts as the original SDeP message. An offered media description part can be rejected by including a corresponding media description part in the reply with the port-parameter set to "0". Acceptance is indicated by including a corresponding media description part with valid parameters.

Session Descr. Part
```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.com
s=
c=IN IP4 atlanta.com
```
Time Descr. Part
```
t=0 0
```
Media Descr. Part 1
```
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```
Media Descr. Part 2
```
m=audio 51372 RTP/AVP 96
a=rtpmap:96 AMR/8000
```
Media Descr. Part 3
```
m=audio 53000 RTP/AVP 97
a=rtpmap:97 AMR-WB/16000
```

Figure 6: An example of an SDeP message header [24] (adapted)

Table 3: Explanations of some SDeP message fields

Field	Explanation
v	Indicates the protocol version. At the time of writing, only one protocol version existed.
o	Indicates the originator of the SDeP announcement. The syntax of this field is: o=<username><session id><version><network type><address type><address> <session id> identifies the session in question. <version> indicates the version of the SDeP announcement. If a user agent sends an SDeP announcement that updates a previously sent announcement, the version number is increased. <network type> indicates the network type, "IN" means internet. <address type> indicates the address type i.e. "IP4" or IP6". <address> Indicates the address of the session initiator in the domain name format, the dotted decimal representation format of IPv4, or the compressed textual representation of IPv6. It is recommended that the time stamp method of the Network Time Protocol (NTP) as specified in RFC 1305 is used for allocating the <session id> and <version> values.
s	This is the session name field. It is mandatory in the SDeP specification but it is not very useful in SIP, because SIP is primarily intended for unicast whereas the session name is generally intended for multicast sessions. It is therefore recommended that the s-field is set to either "-" or " " (space) in SIP usage. [24]
c	This field contains connection data. The syntax of the field is: c=<network type> <address type> <connection address> The <network type> and <address type> parameters are used in the same way as in the o-field. (RFC 2327 does not specify usage of IPv6 addresses but it is specified in RFC 3266, which updates SDeP.) <connection address> contains the address of the endpoint for the intended session. This address may be identical with the address in the o-field but may also differ. Typically the addresses differ when SDeP is used for establishing a multicast session in which case the <connection address> is a multicast address.

Table 4: Explanations of some SDeP message fields (continued)

Field	**Explanation**
t	This field contains the beginning and end times of the session in question. When SIP is used, exchanging this information in the initiation stage of a session is unnecessary since the session can be terminated by means of SIP signaling. Nevertheless, this field is mandatory in the SDeP specification and must thus be included. For SIP, it is recommended that this field is set to ”0 0”, which generally indicates an permanent session [24].
m	This is the media announcement field. The syntax of this field is: m=<media> <port> <transport> <fmt list> <media> indicates the media type. The specified types are ”audio”, ”video”, ”application”, ”data” and ”control” but the list is extensible. <port> indicates the transport level port associated with the <transport> field. <transport> indicates the transport protocol. At the time of writing, the following alternatives where defined by RFC 2327 and IANA: ”RTP/AVP”, ”udp”, ”vat”, ”rtp”, ”UDPTL”, and ”TCP”. In the example of Figure 6, ”RTP/AVP” indicates the Audio/Video profile of RTP. <fmt list> is the payload format(s) list, which includes one or more payload format indicators. If several entries are listed, the first entry is the default format but all formats may be used in the session. This parameter is used in accordance with the specification of the transport protocol indicated by the transport-parameter. For example, if the transport-parameter is ”RTP/AVP”, the payload format may be e.g. ”0” or ”3” indicating Pulse Code Modulation (μ-law, 8 bits) and GSM (residual pulse excitation/long term prediction) respectively. The payload formats from ”96” to ”127” are reserved for dynamic usage and they require attributes in the a-field to be identified precisely. [25], [26]
a	This is the attribute field, which may be included in the session description part and/or the media description parts. The number of a-fields is not limited. Two valid syntaxes are specified for the a-field: a=<flag> a= <attribute>:<value> For example, ”a=sendonly” indicates that the media type the attribute refers to is specified to only send data (e.g. in the case where different addresses are used as traffic destination and traffic source). Correspondingly, ”a=orient:landscape” would indicate that the video media in question uses the landscape orientation. A number of attributes are registered by IANA but unregistered attributes may also be used, if a ”X-”-prefixes is used. In the example of Figure 6, the a-field of media description 1 indicates that the clock rate is 8000Hz. The a-fields of media description 2 and 3 indicate that adaptive multi rate coding (AMR) at 8000Hz and AMR-wideband (AMR-WB) coding at 16000Hz are used respectively.

3.5 Chapter Summary

In this chapter a general example of how a session can be setup with SIP was first given. Then one message of the messages presented in the general example was inspected in more detail and its header fields and message body consisting of an SDeP message were explained. The intention is to give a general level understanding of SIP but also provide some insight on the lower level implementation. A more thorough description of SIP, its various messages and functionalities can be found in e.g. [22] or [16].

4 PUSH-TO-TALK OVER CELLULAR

In September 2003 Ericsson, Motorola, Nokia and Siemens mobile announced that they have developed an industry standard that specifies PoC based on the IMS framework. This standard was also submitted to OMA as a baseline document for PoC standardization within a standardization body with a broader interest group [27, 28, 29, 30, 31, 32, 33]. OMA is a standardization body, which was founded in 2002 and currently has about 300 member companies from all branches related to PLMN business. OMA provides user oriented open standards for enabling bearer agnostic mobile services with interoperability between service providers. In relation to other standardization bodies, OMA provides more holistic specifications across the entire value chain of mobile services. OMA also provides a body for agreeing and standardizing ways for consolidating existing standards of e.g. 3GPP and IETF in mobile services. [34, 35]

This section is based on the OMA PoC Requirements specification [36] and on the most current OMA PoC Architecture specification draft [37] that was publicly available at the time of writing. At the time of writing, the PoC standard of the Open Mobile Alliance (OMA) is work in progress and the latest public draft version of the OMA PoC Architecture specification is dated April 7, 2004 [37]. The current target is to publish a candidate enabler in December 2004 [38]. A target date for publishing an approved enabler has not been publicly announced. An "enabler" is defined as a set of specifications that combined together define a platform for providing a service [39]. The OMA PoC requirements document was approved on October 8th 2003 [36]. The requirement document specifies terminology, use cases, high-level functional requirements, and operational and overall system requirements.

As stated in the charter of the PoC working group of OMA: "PoC service is a half-duplex form of communications that allows users to engage in immediate communication with one or more receivers, similar to Walkie Talkie type operation, simply by pushing a button on their handsets." [2] Figure 7 exemplifies a one-to-many PoC session in which group member B speaks to the other members in the group in question.

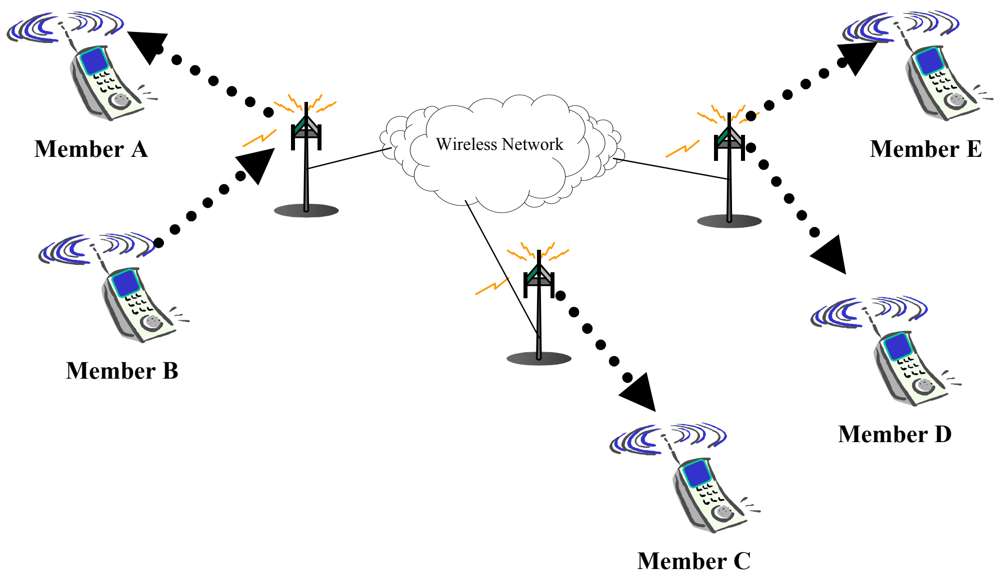

Figure 7: A one-to-many PoC session from member B to the other group members [37]

4.1 Architecture and Functional Entities

The general OMA PoC architecture is presented in Figure 8. The PoC client resides in the user equipment i.e. a mobile terminal. The access network and the SIP/IP Core, which is based on IMS, provide the PoC client access to PoC Servers and Group and List Management Servers (GLMS) both in local and remote PoC networks. (In the figure, the abbreviation MMD refers to the All-IP Core Network Multimedia Domain defined by 3GPP2. MMD includes an IMS corresponding to the 3GPP IMS. [40]) A PoC client registers itself to the SIP/IP core, after which SIP signalling between the PoC client and PoC server is enabled. The SIP/IP core also handles the authentication of PoC clients. The PoC servers take care of the handling of PoC and SIP sessions, floor control etc. As its name implies, the GLMS handles group and list management i.e. stores e.g. PoC group, contact and access information and provides methods for creating, modifying and retrieving this information. A presence server may be used to enable presence related PoC features, these are, however, outside the scope of this thesis.

The relation of the PoC elements to the IMS framework is presented in Figure 9. The PoC elements are encircled by the dashed line and the interfaces between the PoC and IMS elements are depicted. The abbreviation ISC denotes the IP multime-

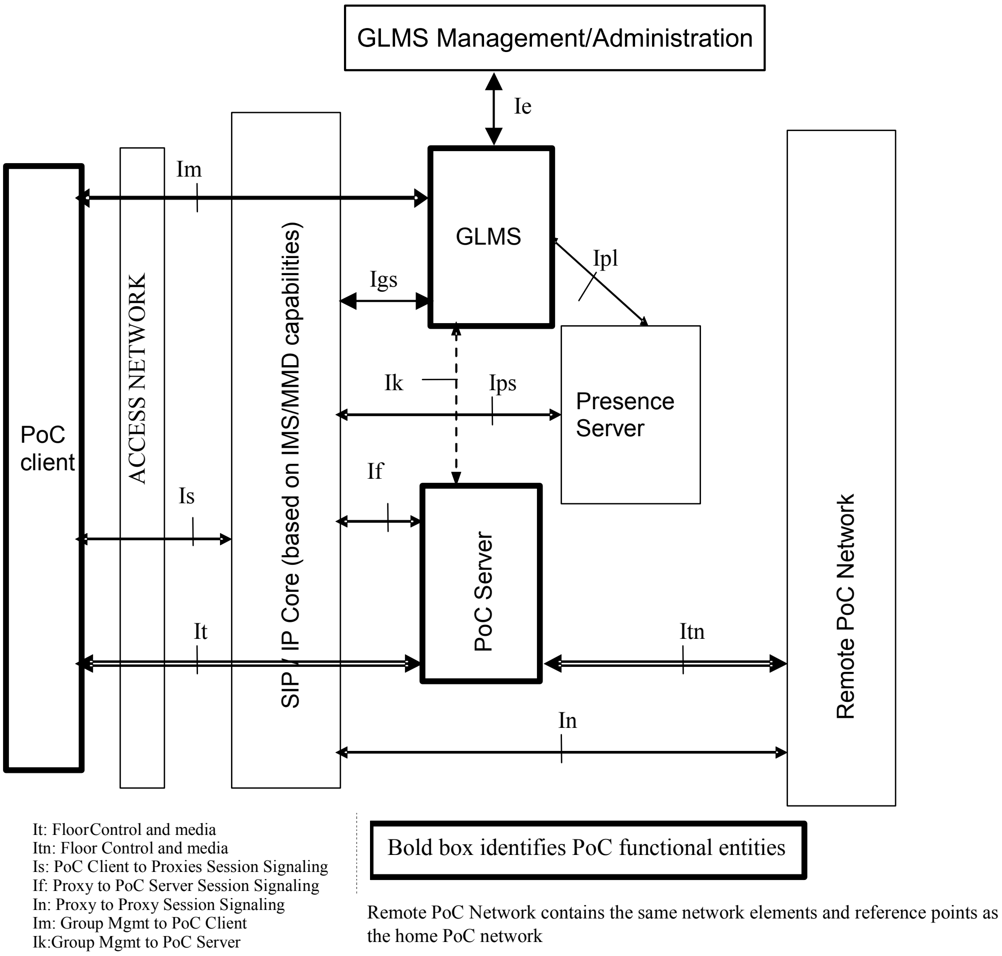

Figure 8: PoC architecture [37]

dia Subsystem Service Control Interface, which is SIP based and defined in 3GPP TS 23.228 [19]. The ISC-interface corresponds to the If-interface of the OMA specifications. From the IMS perspective, a PoC server is seen as an application server, which utilizes the SIP capabilities of IMS. Correspondingly the Ut-interface of IMS corresponds to the Im-interface of OMA PoC. [41]

4.2 Addressing

According to the PoC specification, each user has at least one SIP URI for addressing purposes and optionally a telephone URI. Telephone URI:s are specified in RFC 2806 [42], an example is "tel:+3589465077". SIP URI:s are also used for the addressing of groups. A group with a specific URI can be created and stored in a GLMS.

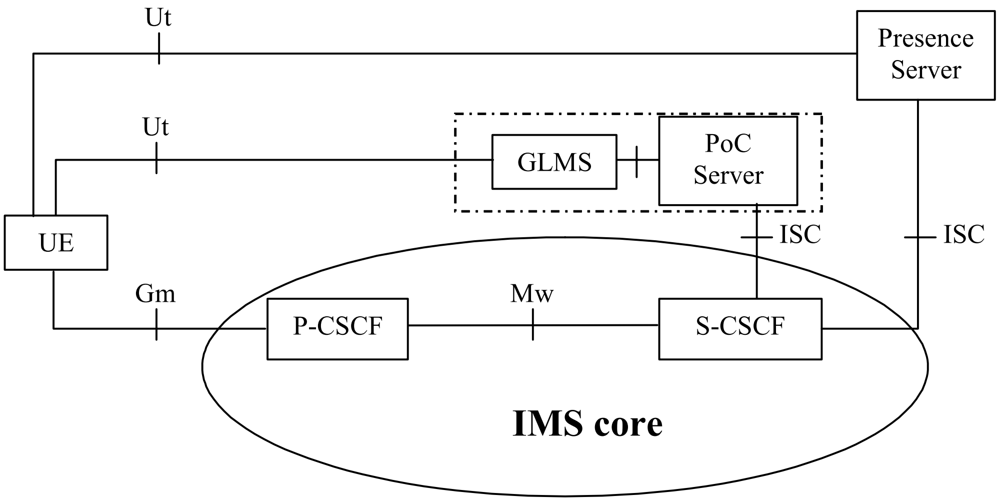

Figure 9: PoC architecture from IMS perspective [41]

4.3 Features

The OMA PoC requirements document outlines several different features, some of which are described in the following sections.

4.3.1 One-to-One Communication

The one-to-one communication feature is specified to provide PoC communication between two users. The initiating party sends an invitation to the desired recipient. The receiving device handles the invitation depending on whether it is set in automatic answer mode or manual answer mode. In the case of manual answer mode, the recipient must accept the invitation before the originator is given the right to speak. The SIP signalling flow is analogous to the one-to-many ad hoc group session case presented in Figure 10 and explained in Section 4.3.2. In case of the automatic answer mode the receiving device automatically accepts the invitation after which the initiating party may transmit speech to the recipient. The signalling flow is similar to Figure 10 but the RINGING-messages (8. - 14.) are missing and the OK-messages (15. - 21.) are replaced by UNCONFIRMED OK-messages.

In addition, a feature called Instant Personal Alert is specified for initiating one-to-one communication. This feature is basically a call back request, i.e. a notification that the sender wishes to speak to the recipient.

4.3.2 One-to-Many Communication

The OMA PoC Requirements specification defines three modes of one-to-many communication: pre-arranged, ad hoc and chat. Corresponding groups can be created and stored in GLMS by the users or administrators. Pre-arranged and chat groups are persistent whereas ad-hoc groups are temporary and their information is thus removed from the GLMS e.g. when the ad-hoc group session is terminated or after a certain period of time has elapsed without activity. Users can create and modify groups using their mobile terminal. The automatic and manual answer modes described in the previous section apply to one-to-many connections as well.

A *pre-arranged group* has a permanent URI by which all the group members can be contacted.

A *chat group* is a group, whose sessions users can join and leave individually in a similar manner as in e.g. www-based textual chat groups. Chat groups are thus similar to other PoC groups except for that inactive users are not invited to sessions. In other words, a user that is a member of a chat group but is currently not active in the group does not receive any talk bursts[2] or invitations regarding ongoing sessions.

An *ad-hoc group* can be created by a user by inviting a number of individuals to the group. It is also possible to define restricted groups with lists of the members allowed in the group.

Ad-hoc groups

An example of creating and initiating an ad hoc group session or a one-to-one session is presented in Figure 10, whose message flow is explained below. In the figure, only one invitee is depicted for clarity but other invitees would be invited in a corresponding manner. In the example, the invitees are assumed to be in manual answer mode.

1. - 7. PoC client A invites one or more PoC subscribers to a PoC session by sending an INVITE-message including the invited subscribers' URI:s and media

[2]The term "talk burst" is defined by OMA as "The media recording, transport and playback that occurs from the point the PoC Client has got the permission to send a talk burst until the permission is released." [37] In other words, a speech message conveyed from one user to others.

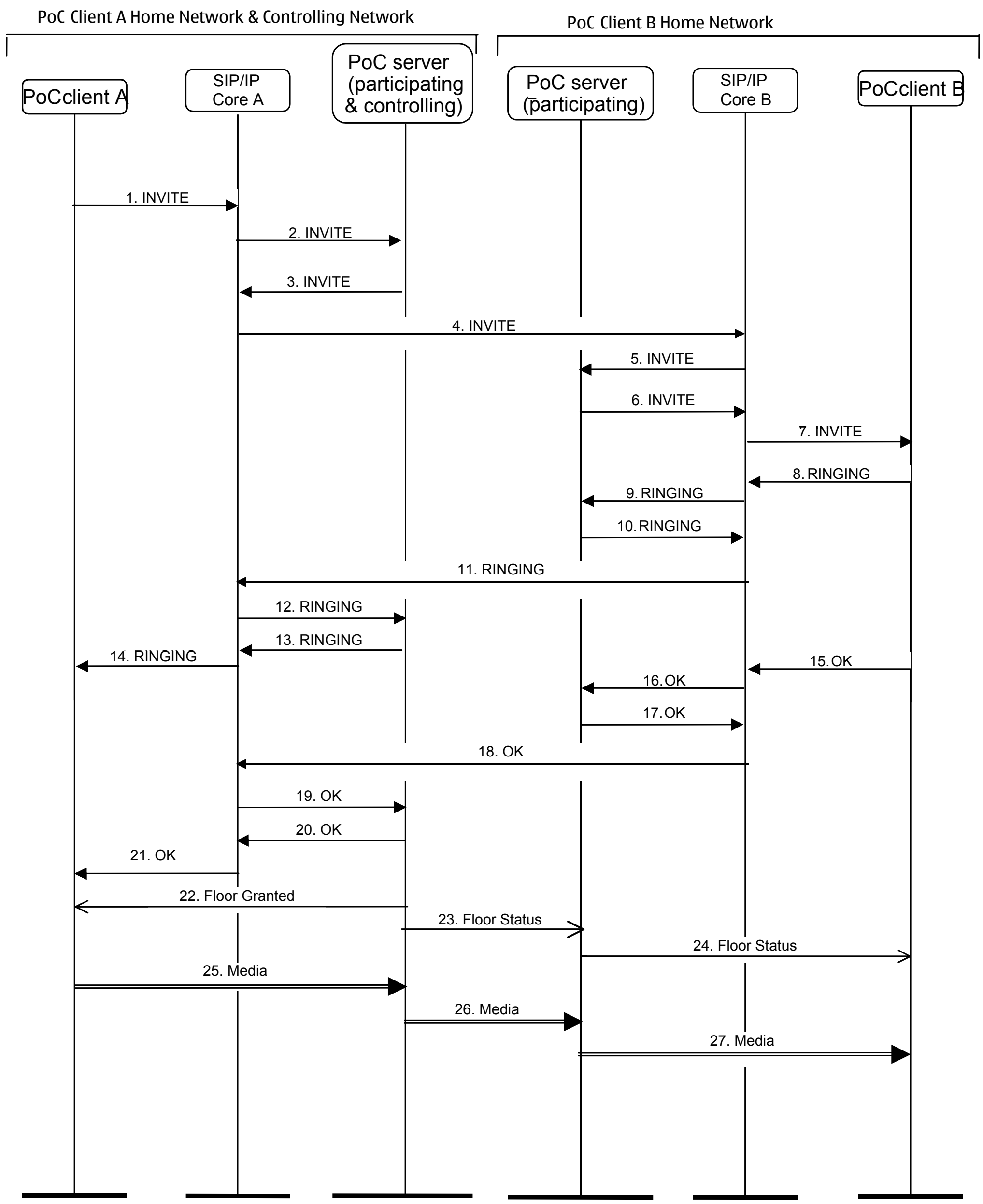

Figure 10: Message flow of one-to-one and ad hoc group session formation [37] (adapted)

parameters regarding PoC client A. The message is routed via the SIP/IP core to the PoC server of the home network of client A. The PoC server then locates the PoC servers of the invited PoC clients on the basis of the URI:s and forwards the INVITE-message via the SIP/IP cores of the home networks of PoC client A and B. The PoC server of PoC client B's home network then forwards the message to PoC client B via the SIP/IP core of PoC client B's home network.

8. - 14. The invited clients phone alerts the user and sends a RINGING-message to PoC client A. The message is conveyed in a similar manner as in steps 1. - 7.

15. - 21. When PoC client B accepts the invitation, an OK-message is sent to the inviting PoC client A. The message is conveyed in a similar manner as in steps 1. - 7.

22. The PoC server of PoC client A's home network acts as the controlling PoC server of the established session and thus handles e.g. floor control. A "Floor-Granted"-message is sent to PoC client A to indicate the right to speak.

23. - 24. A Floor Status message is conveyed to PoC client B in order to notify that the floor is occupied and an incoming talk burst is to be expected.

25. - 27. A PoC talk burst is conveyed over a media connection from PoC client A to PoC client B (and other members in the group).

The session initiation messages of steps 1. - 21. are conveyed over the If-, Is- and In interfaces. The floor control messages and media flow are conveyed over the It- and Itn interfaces. These interfaces are presented in Figure 8.

Pre-arranged Groups and Chat Groups

The signaling for pre-arranged and chat group sessions are similar to what is described above but a single URI is used for contacting all the group members.

4.4 Pre-established Session

A pre-established session may be created between a PoC client and a PoC server well in advance of any actual user-to-user communication. Basically a pre-established session is established in order to negotiate media parameters for the media and floor control communication between the PoC client and PoC server. By creating a pre-established session, the above-mentioned negotiation is not needed when e.g. the PoC client initiates a PoC session. If a pre-established session exists between the PoC client and server, the signaling flow is somewhat different from what is described above in Section 4.3.2 and Figure 10.

4.5 Leaving a PoC Group Session

A group participant may leave a group session by sending a BYE-message to the controlling PoC server. The PoC server removes the participant from the group and sends an OK-message to the user in question. The messages are conveyed in a similar manner as in Figure 10.

4.6 Adding a Participant to a PoC Session

A PoC subscriber participating in a PoC session may invite any other PoC subscriber to the ongoing PoC session, provided that the maximum number of participants is not exceeded. An example of this procedure is presented in Figure 11 and explained below. In the example the invitee is in automatic answer mode.

1. - 2. The inviting PoC subscriber invites another PoC subscriber to an existing PoC session with a REFER-message, which is conveyed to the controlling PoC server. In the example of Figure 11, the controlling PoC server is the PoC server of the home network of PoC client A.

3. - 5. The PoC server acknowledges the REFER-message and states that the invitation procedure is initiated by sending an OK-message to PoC client A.

6. - 7. The controlling PoC server sends an INVITE message to the PoC server of the invitee's home network.

9. - 10. The PoC server of the home network of PoC client B sends an Auto-Answer message to the controlling PoC server, since PoC client B is in auto answer mode.

10. - 12. The controlling PoC server sends a NOTIFY-message to the inviting PoC client A in order to notify that the invitation has been successful.

13. - 14. PoC client A acknowledges the NOTIFY-message by sending an OK-message to the controlling PoC server.

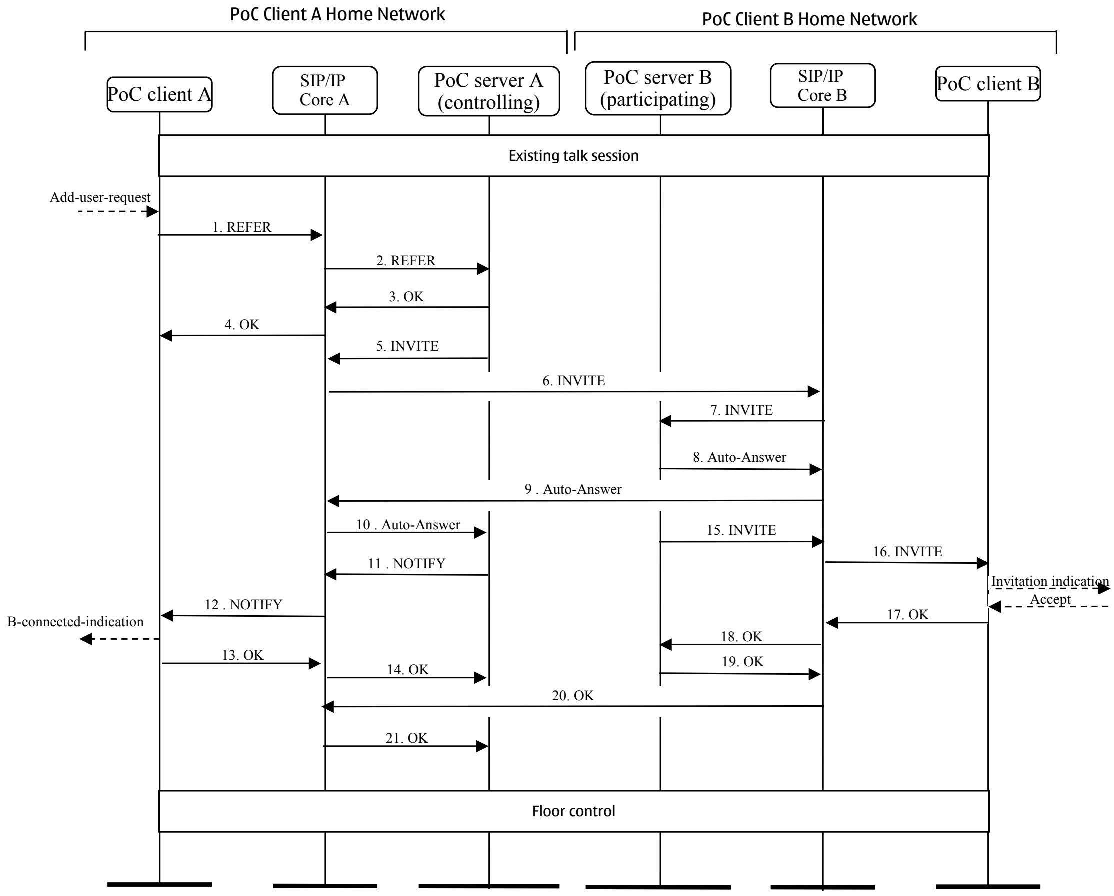

Figure 11: Message flow of adding a participant to a PoC session [37] (adapted)

15. - 16. The PoC server of the home network of PoC client B forwards the INVITE-message to PoC client B.

17. - 21. The user of PoC client B accepts the invitation and an OK-message is sent via the participating PoC server to the controlling PoC server. Hereafter, the PoC session proceeds as usual.

4.7 Floor Control

The OMA PoC requirements document defines floor control as a ”mechanism for the arbitration of the sequence of PoC participants to speak”. To put it simply, the PoC server acts like a chairman that gives the floor to the participants of a session on the basis of their requests to speak.

At the time of writing, the details of the floor control mechanism are an open issue.

However, the high level functional requirements are defined. Basically, each PoC session participant is able to request the floor upon which the PoC server either grants a permission to speak or, if the floor is already occupied by another user, denies the request. In addition, the PoC server may optionally queue request and notify the requester of the queueing. When the floor is granted to a participant, the other PoC clients in the group are notified of this, in order to avoid unnecessary requests. After a talk burst, the PoC client in question releases the floor by sending an indication of the releasing to the PoC server. The PoC server may also force the floor to be released, e.g. in case a PoC subscriber attempts to send a talk burst that is longer than the specified maximum. When the floor has been released, a notification of this is sent from the PoC server to all participating clients.

In the industry standard of the Ericsson, Motorola, Nokia, Siemens-consortium, the floor control mechanism is specified in more detail. [30]

4.8 Simultaneous PoC Sessions

A PoC subscriber may be a member in several PoC groups and in several PoC sessions simultaneously. Furthermore one of these sessions may be selected as the primary group whereas the others in that case are secondary groups. In order to prevent the user from hearing a mixture of talk bursts from different sessions the following rules are followed depending on whether a primary group is selected or not. The general policy is that an ongoing discussion is not interrupted by another discussion unless it has higher priority. A discussion is regarded as begun by the first talk burst and ended after a certain time of inactivity in the group in question.

No Primary Group Selected

If no primary group has been selected, an ongoing discussion is never interrupted. Incoming talk bursts from other groups are thus filtered and thereby missed by the user. After a discussion has ended, a discussion from another group is passed to the user.

Primary Group Selected

If a primary group has been selected, a discussion in the primary group always interrupts a discussion of a secondary group. An ongoing discussion in a secondary group is thus rejected and subsequent talk bursts are filtered and missed by the user until the discussion in the primary group has ended. Among secondary groups the priority is identical to the case where no primary group is selected. However, if a user happens to be talking in a secondary group, he/she is not interrupted by an incoming talk burst from the primary group but merely notified of the event. That is, a discussion in a secondary group is interrupted but an outgoing talk burst is not.

[36, 37]

4.9 Protocols

At the time of writing, the OMA PoC Architecture document does not explicitly define all the protocols to be used but does include clear indications that SIP and RTP will be used. It seems reasonable to assume that a protocol stack similar to the one of Figure 12, which is the protocol stack of Nokia's proprietary PoC solution [43], will be used. This assumption seems to be supported by several public contributions [44, 45, 46] as well as the descriptions of proprietary solutions in Visiongain's report [1]. The protocol stack employs an IP/UDP/RTP based user plane for conveying communication data, and a IP/UDP/SIP based control plane for signalling. In addition, the floor control also requires a signalling protocol, this remains an open issue in the OMA PoC documents at the time of writing.

If the protocol stack of Figure 12 is used, the headers of the IP, UDP and RTP layers may be as presented in Figure 13. The well known IP protocol is specified in RFC 0791 [47]. In the figure UDP Lite refers to a proposed alternative version of the well known UDP protocol [48]. UDP Lite differs from conventional UDP in the sense that the checksum covers only a part of the payload [49, 50]. This is advantageous e.g. when codecs that allow some errors in a part of the payload are used (see Section 4.10). The well known RTP is specified in [51] and explained in [52]. The Timestamp- and Synchronization Source-fields of the RTP header are used to synchronize real-time data conveyed in separate RTP packets. The initial

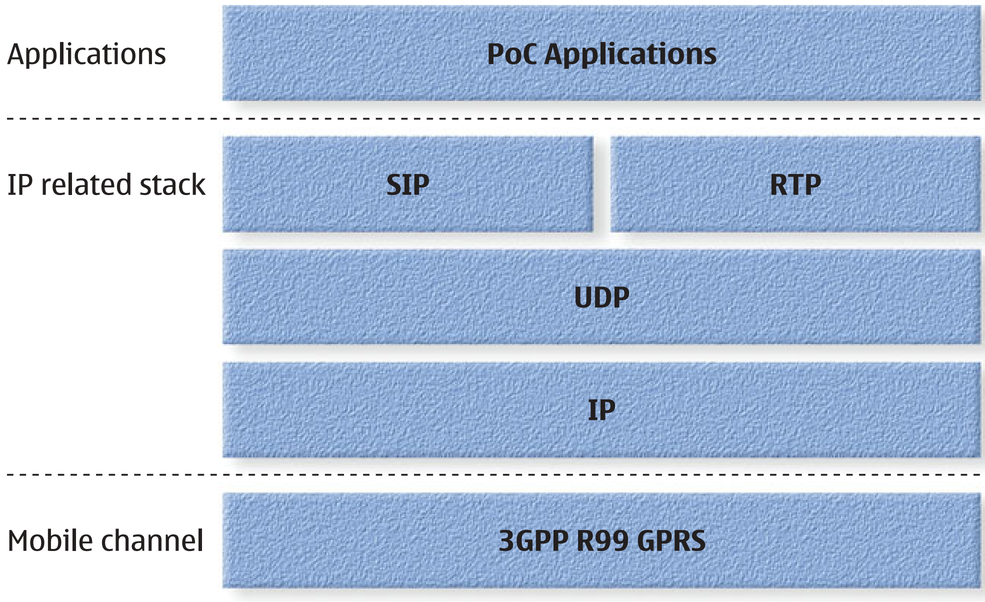

Figure 12: The protocol stack of Nokia's proprietary PoC solution [43]

Time Stamp-value is randomly chosen for each transmission [52]. The payload in the figure contains a payload header and a voice codec, these are explained in the following section.

4.10 Voice Coding

At the time of writing, the voice codecs also belong to the area of open issues in the publicly available OMA PoC specification drafts. It seems, however, reasonable to assume that it is probable that the Adaptive Multirate (AMR) codec will be used since the consortium of Ericsson, Motorola, Siemens, and Nokia specifies it in the industry standard [31].

4.10.1 Adaptive Multirate Codec

AMR is specified in 3GPP TS 26.071 [53] and its references. As its name implies, the codec is adaptable and the number of bits allocated to speech coding is thus variable. If the number of bits allocated to speech coding is reduced, the number of bits allocated to channel coding can be increased without increasing the total capacity requirement. This enables e.g. GSM to adapt the ratio of bits allocated

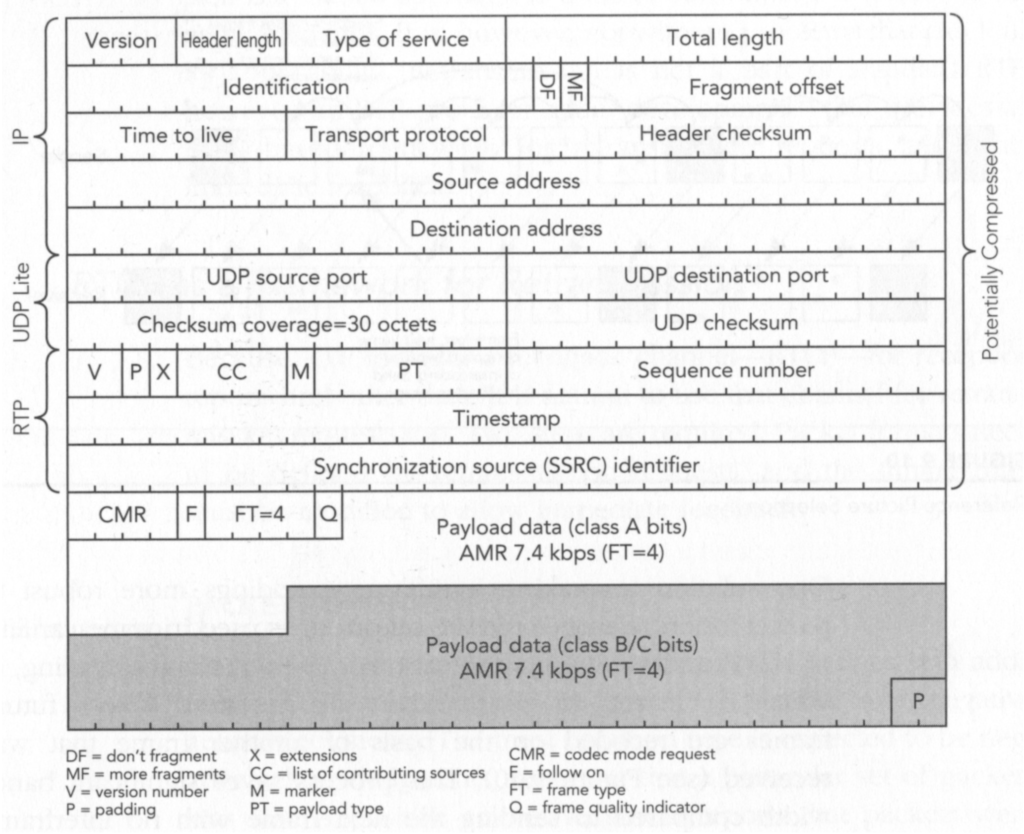

Figure 13: IP, UDP and RTP headers (and an AMR payload) corresponding to the protocol stack of Figure 12 [52]

to speech and channel coding depending on the prevailing interference and fading conditions of the radio link. It should be noted that the channel coding is not specified in AMR itself but it is left to the application exploiting AMR.

The frame sizes corresponding to the different modes of AMR are presented in Figure 14. The abbreviations FR and HR refer to full and half rate as used in GSM. The codec modes on the x-axis relate to the kbps-rate required for speech coding. Each AMR frame corresponds to a time interval of 20 ms. Frames must thus be sent at a rate of 50 frames per second.

Two different AMR codec formats are defined, the AMR Interface Format 1 (IF1) and Interface Format 2 (IF2) (a.k.a. octet aligned format).

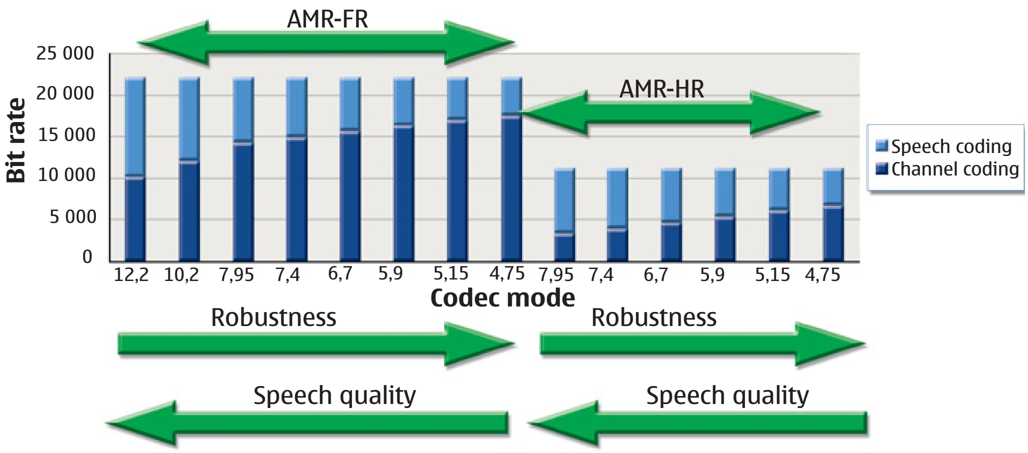

Figure 14: Bits allocated to voice and channel coding in different AMR modes [54]

AMR IF1

The general structure of an AMR IF1 codec is presented in Figure 15. The Frame Type (FT) and Mode Indication-fields indicate which one of the eight AMR modes (see Figure 14) is in use and whether the current frame is a conventional AMR frame or a comfort noise or empty frame. The Frame Quality Indicator-field (FQI) indicates whether the frame contains errors. The (Codec) Mode Request-field (CMR) is used to communicate that a mode change to the indicated mode is requested. The CRC field includes a CRC checksum for the class A bits described below. In addition five spare bits are added between the CRC field and the core frame in order to align the beginning of the core frame to the beginning of a bit octet.

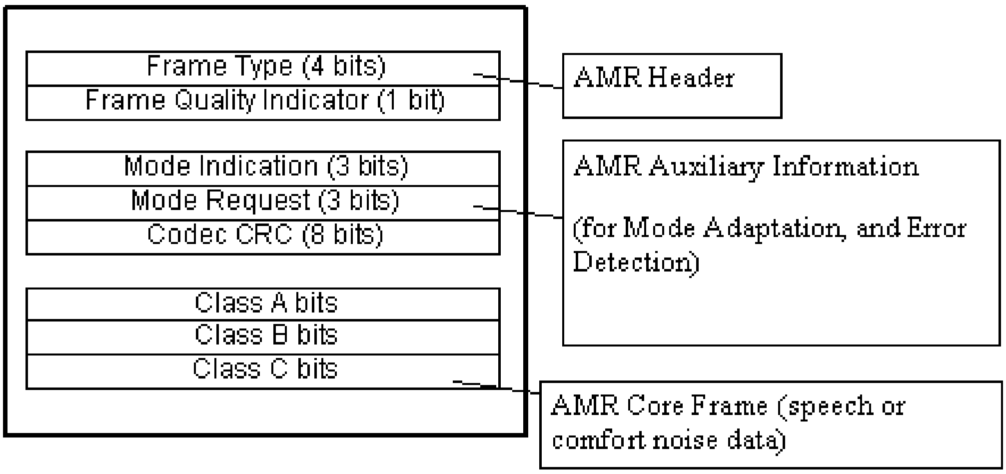

Figure 15: The generic structure of an AMR IF1 codec [55]

The core frame includes the actual coded voice, which is divided into A, B, and C bits. The class A bits are the most sensitive bits. Errors in these bits usually cause corruptions that require error concealment. Class B bits are less sensitive to errors than class A bits and C bits are the least sensitive.

AMR IF2

The general structure of an AMR IF2 codec is presented in Figure 16. As can be seen, the IF2 frame is somewhat simpler than the IF1 frame. When needed, bit stuffing is added to the end of the frame in order to octet align the frame i.e. to make the frame length a multiple of eight.

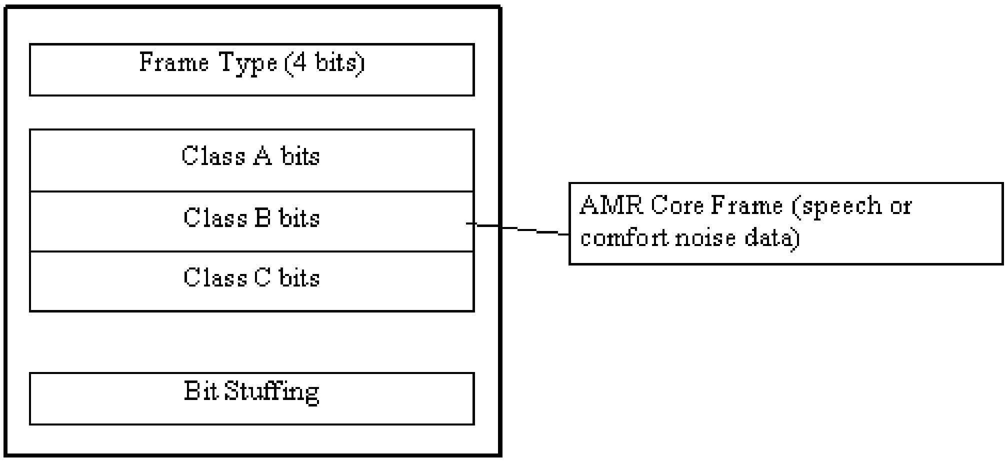

Figure 16: The generic structure of an AMR IF2 codec [55]

4.10.2 AMR Within RTP

The usage of the AMR codec within RTP is specified in RFC 3267 [56]. The conventions for IF1 and IF2 are briefly described below.

AMR IF1

An example of an AMR IF1 frame embedded in an RTP packet is presented in Figure 13. As depicted in the figure, the AMR Header and AMR Auxiliary Information fields are limited to the CMR, FT, and FQI-fields. In addition, a Follow On-field (FO) is used to indicate whether additional AMR frames are included in the RTP packet. Several AMR frames may thus be included in a single RTP packet, in which case a separate FT, FQI, and FO-field is included for each AMR frame. However, only one CMR-field is included per RTP packet.

AMR IF2

If AMR IF2 packets are used, the CMR-field is superseded by four padding bits, in order to achieve octet alignment. The FO, FT, and FQI-fields are treated as an entity of six bits and octet alignment is thus achieved by adding two padding bits after the FQI-field. In addition, optional CRC signalling fields related to interleaving may be included. Several AMR frames may also be included in an RTP packet when IF2 is used. As in IF1, only one CMR field is included per RTP packet.

The total number of bits in the RTP payload per AMR frame for each mode in IF1 and IF2 are presented in Table 5. The numbers correspond to the case where only one AMR frame is sent per packet and no optional fields are used. (The values do not include the IP, UDP, and RTP headers.) For IF1, additional overhead is caused by the fact that the IP frame length is a multiple of eight. This overhead is presented in the table in parenthesis. If several AMR frames are conveyed within an RTP packet, the AMR overhead is somewhat reduced since the CMR-field (4 bits in IF1 and 8 bits in IF2) occurs only once in each RTP packet.

Table 5: AMR frame sizes for IF1 and IF2 when a single AMR frame is sent per RTP packet. [55, 56]

AMR Mode	IF1	IF2
4,75	105 (+7)	120
5,15	113 (+7)	128
5,90	128 (+0)	144
6,70	144 (+0)	160
7,40	158 (+2)	168
7,95	169 (+7)	184
10,2	214 (+2)	224
12,2	254 (+2)	264

4.11 Chapter Summary

This chapter gave an overview of PoC as described in the latest public OMA standard drafts. The different features where described at a high level with a few examples giving some insight on the more detailed implementation issues. The emphasis was on ad-hoc group sessions and related issues. Some of the open issues i.e. protocols and voice coding where discussed as well. Most of the protocols are rather well known and they where thus not described in detail. The AMR codec was described at a general level as well. Emphasis was on presenting the protocol headers and AMR frame sizes in order to provide overhead information for calculation of capacity requirements.

5 IP OVER BLUETOOTH

The Bluetooth specification is divided into the core specification and the profile specifications. The core specification defines the common requirements for all Bluetooth devices and provides thus a platform for development of higher-level applications and functionalities. The profiles, on the contrary, are optional extensions, which offer building blocks for higher-level functionalities and interoperation. The selection of profiles also promotes interoperability between devices manufactured by different companies. One of the profiles, the Personal Area Networking (PAN) profile, is designed for IP-based networking. [57] This chapter describes the PAN profile and the Bluetooth Network Encapsulation Protocol (BNEP) that it uses. Some of the lower level protocols and capabilities are also briefly presented.

5.1 Protocols and Capabilities of the Bluetooth Core

This section is based on the Bluetooth Core specification versions 1.1. and 1.2 [58, 59].

An overview of the Bluetooth architecture, as specified in the Core Specification, is presented in Figure 17. The PAN profile exploits the asynchronous link provided by the Logical Link and Adaptation Protocol (L2CAP). The baseband provides the L2CAP with an asynchronous connection-less (ACL) link that provides error detection with a CRC algorithm. L2CAP also provides a QoS functionality, which is not, however, used by PAN and BNEP as currently specified. The Link Manager and Channel Manager are used to create and manage piconets and connections. The link managers of different devices communicate with the Link Manager Protocol (LMP). In addition, a service discovery protocol (SDiP[1]) is specified for discovering devices with the desired capabilities (not depicted in the figure). [59]

[1] In general, the acronym SDP is used to denote the Service Discovery Protocol. However, since this thesis discusses both the Session Description Protocol and the Bluetooth Service Discovery Protocol, which are both generally abbreviated as SDP, the acronyms SDeP and SDiP are used in this thesis for clarity.

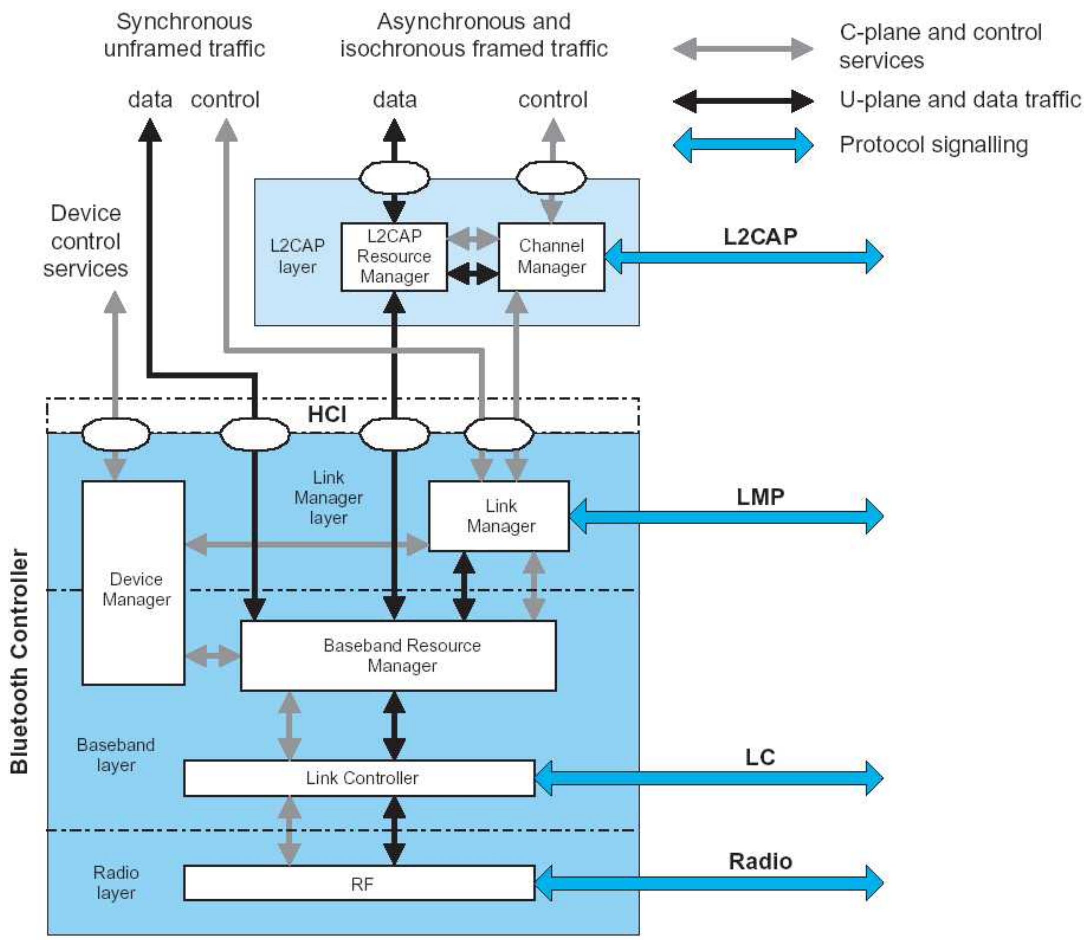

Figure 17: Bluetooth Core Architecture [59]

5.1.1 Piconets, Scatternets and Network Formation

Bluetooth employs a master-slave approach to communication between devices. That is, connections are set up between two devices of which one is the slave and the other is the master, which coordinates the communication. One master can have up to seven active slaves and a number of slaves in the power save modes Sniff, Hold and Park. Such a network is called a piconet. One device can be the master of only one piconet but can simultaneously be a slave in a number of other piconets. Such a network consisting of several interconnected piconets is called a scatternet. Examples of two piconets and a scatternet are depicted in Figure 18. In the figure, the scatternet includes one device that is a slave in two piconets and a device that is a slave in one piconet and the master in another.

Inquiry and Page procedures are specified for discovering nearby devices and connecting to devices respectively. The Inquiry procedure provides a device with a list of the other devices within the Bluetooth range that are receptive to Paging requests. The inquiring device sends inquiry requests and the devices in the vicinity respond with messages including their Bluetooth device address (BD_ADDR see Section 5.1.2). The Paging procedure establishes a connection between the initiating device, which becomes the master, and the responding device, which becomes a slave. The recipient of a Page message is indicated by the BD_ADDR.

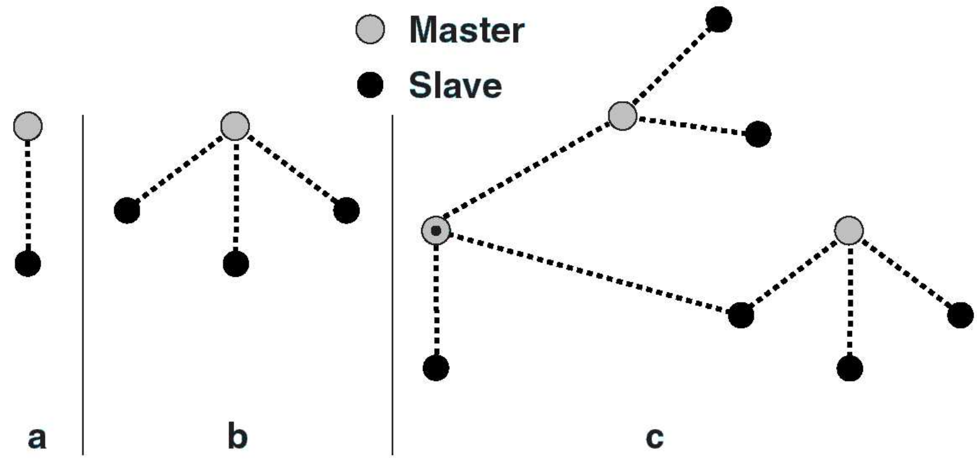

Figure 18: Examples of piconets (a and b) and a scatternet (c) [59]

5.1.2 Baseband Addressing

On the baseband level, four different Bluetooth address types are used: Bluetooth device address (BD_ADDR), active member address (AM_ADDR) parked member address (PM_ADDR) and access request address (AR_ADDR). The BD_ADDR is a universally unique and device-specific 48 bit address according to IEEE 802 [60]. The BD_ADDR is not used as such for communication between Bluetooth devices. Instead channel access codes (CAC), which are piconet specific, and device access codes (DAC), which are device specific, are derived from the BD_ADDR. These access codes are not, however, unique, so conflicts may occur. An AM_ADDR is a 3 bit address allocated to an active member in a piconet, which, as mentioned earlier, may have seven slaves. The AM_ADDR 000 is reserved for broadcasting. A PM_ADDR is an 8 bit address allocated to a parked member of a piconet. The master uses this address to activate a parked slave. The number of parked slaves is not, however, limited by the length of this address. If the number of parked slaves exceeds 255, the PM_ADDR 00000000 is assigned to any additional parked members, which are then differentiated by the BD_ADDR. An AR_ADDR is an 8 bit address allocated to a parked member of a piconet. A parked slave uses this address for requesting unparking, i.e. expressing its desire to become an active member to the master. In addition to these addresses, a Bluetooth device may be assigned a Bluetooth device name, which is a 248 byte name in ASCII (American standard code for information interchange) format. [57]

In the Bluetooth Core specification version 1.2, the AM_ADDR is replaced by the logical transport address (LT_ADDR), which differs from AM_ADDR mainly in the sense that a slave can have several LT_ADDR:s if it has extended synchronous connection-oriented (eSCO) links [59]. The current version of the PAN profile does not, however, use eSCO links.

5.1.3 Host Controller Interface

The Host Controller Interface (HCI) is an optional but commonly used interface e.g. between a Bluetooth chip (i.e. controller) and a processor controlled device (i.e. host) exploiting the Bluetooth capabilities of the Bluetooth chip. The partitioning of Bluetooth Core functions between the host and the controller is presented in Figure 17. L2CAP and SDiP belong to the host whereas the lower levels are handled by the controller.

Commands

HCI provides commands e.g. for setting up connections, sending data and monitoring the link quality. Some examples of commands are given below.

- **HCI_Read_BD_ADDR** This command enables the host to acquire the BD_ADDR of the controller.

- **HCI_Create_Connection** With this command. the host can request the controller to set up an ACL connection to a device with a given BD_ADDR.

- **HCI_Write_Automatic_Flush_Timeout** This command is used to set a timeout for baseband flow control of ACL packets. The timeout value can be set at any value between 0.625 ms and 1279.375 ms at intervals of 0.625 ms (i.e. Timeout = N * 0.625 ms). When transmitting data, the controller attempts to transmit a message to a device until the flush timeout occurs. When the timeout occurs, a Flush Occurred event is given to notify the host that the packet has been flushed i.e. transmission has failed.

- **HCI_Write_Page_Timeout** This command enables the host to set the paging timeout. Setting the timeout corresponds to the Flush Timeout case but the range is up to 40.9 s. The host can thus specify how long a paging attempt lasts.

- **HCI_Write_Link_Supervision_Timeout** This command is used for setting the supervision timeout, which defines how long the link manager waits before it regards an inactive link as broken. Setting the timeout corresponds to the Page Timeout case.

Connection Handles

When a connection is setup with the HCI_Create_Connection command, the controller assigns a 12 bit Connection Handle for the created connections. When the host sends data, it identifies the recipient with a Connection Handle. Similarly, when the controller conveys received data to the host, the originator of the packet is identified with a Connection Handle.

Interfaces

The HCI provides three alternative interfaces for inputting and outputting data, commands and events: Universal Asynchronous Receiver Transmitter (UART), Universal Serial Bus (USB) and RS323. UART is the simplest of these three. It has a four pin physical interface over which data is conveyed serially. Data and command packets are superseded by a one byte indicator, which indicates the packet type.

5.2 PAN Profile

This section is based on the Personal Area Networking Profile [61] specified by the Bluetooth Special Interest Group.

The PAN profile specifies how IP networking capabilities can be implemented in Bluetooth devices with the Bluetooth Network Encapsulation Protocol (BNEP). PAN supports both IPv4 and IPv6 and specifies e.g. IP address allocation, IP routing and manual network formation. However, automatic network formation, general ad-hoc networking with multiple piconets (e.g. scatternets), and QoS mechanisms are not defined by the current specification. Three types of PAN units are defined in the specification corresponding to the identified use case scenarios: Network access points (NAP), units providing group ad-hoc network (GN) service, and PAN users (PANU). It is also possible that a unit has several or all of these roles, i.e. a unit can e.g. be a PANU, a GN service node, and a NAP at the same time. NAP:s are units that act as bridges, proxies or routers between a Bluetooth network and some other network. NAP:s are of lesser relevance to the topic of this thesis and they are thus not discussed in detail. A unit providing GN service is a piconet master with 1-7 PANU slaves and possibly additional non-active PANU slaves in park mode. The GN master is able to forward data from one PANU to another with BNEP. PANU:s are units that are able to use the NAP and GN services. Peer-to-peer communication between two (but not several) PANU:s is also defined.

Communication between NAP:s and units providing GN service is not defined. The specified interaction scenarios are thus the ones presented in Table 6.

Table 6: Valid interactions between PAN profile units [61] (adapted)

		INITIATOR		
		NAP	**GN Service Unit**	**PANU**
ACCEPTOR	**NAP**	NO	NO	YES
	GN Service Unit	NO	NO	YES
	PANU	YES	YES	YES

The protocols used in communication between a PANU and a GN service unit or between two PANU:s are presented in Figure 19. At the transmitting end, the BNEP packets are segmented by L2CAP in order to fit them in the baseband ACL packets. Correspondingly, L2CAP reassembles the BNEP packets at the receiving end. An additional simple integrity check is also performed by comparing the value of the length field and the actual length of the received L2CAP packet. A Management Entity (ME) manages link establishment, takes care of some piconet administration issues, and enables encryption when required.

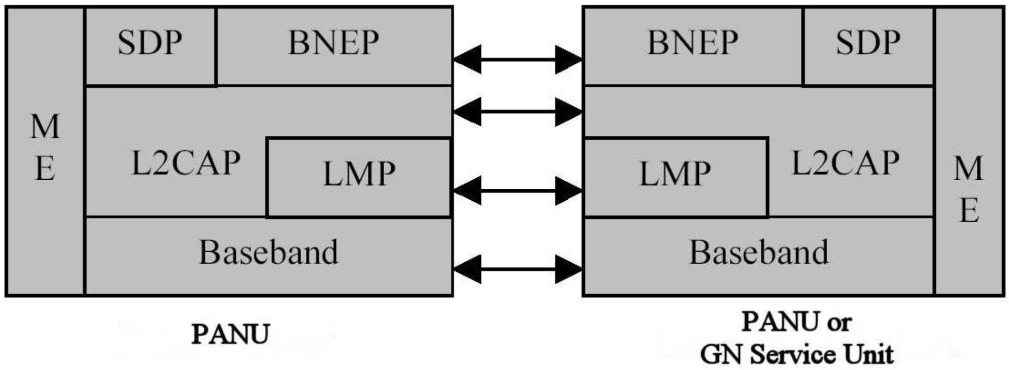

Figure 19: PAN protocol stack for PANU-PANU and PANU-GN service unit communications [61] (adapted)

5.3 Addressing

5.3.1 Address Allocation

According to the PAN profile specification, IP addresses are allocated to PANU:s and GN service units according to the "Dynamic Configuration of Link-Local IPv4 Addresses"-internet draft, anticipating a request for comments (RFC) of the Internet Engineering Task Force (IETF). In accordance with the internet draft, devices in a Bluetooth PAN are assigned with an address randomly selected from the IP-address space 169.254.1.0 - 169.254.254.255. This "link local"-address space is dedicated by the Internet Assigned Numbers Authority (IANA) for auto-configuration and communication between hosts on a single link [62]. These addresses are not universally unique and may thus be used only for communication between hosts on the same link, for example between devices in a Bluetooth PAN. The Internet draft also specifies mechanisms for address claiming, announcing, conflict detecting and defending. [63] If IPv6 is used, the IPv6 autoconfiguration method is used as defined by RFC 2462.

Since the link local addresses are not universally unique, interoperability problems may occur with applications that always assume addresses to be unique. Problems may occur e.g. if an application embeds a link-local address within packet that is communicated to a host outside the link. For example, the session initiation protocol (SIP) may embed a link-local address in a message that is sent to a SIP proxy outside the local link. This address is naturally improper outside the link and the SIP proxy would thus receive a dysfunctional address. [63] This problem may be solved by using an application level gateway (ALG) that resides in a device connecting the local link with another link (e.g. a router, firewall or Bluetooth NAP). The ALG transparently replaces the local IP addresses embedded in the SIP message payload with the routable address of the device in which the ALG. The destination address of incoming packets must naturally be replaced correspondingly. In order to achieve this, the ALG maintains a register of recent messages to SIP proxies. [64, 65, 66, 67, 22]

5.3.2 Address Resolution

The PAN profile implements the address resolution protocol (ARP) as specified by RFC 826 [68]. ARP is a method for acquiring the medium access control (MAC) address corresponding to a known IP address. According to the protocol, each node maintains a cache of associations between MAC and IP addresses. This cache is the primary source for resolving a MAC address. If no entries are found in the cache for a certain IP address, the node broadcasts an ARP query containing the IP address. Each node receiving the ARP query then checks whether the IP address matches its own IP address. If the addresses match, the node sends a response, which contains the MAC address, to the source of the query. [69]

The BT_ADDR is considered as the MAC address in the PAN profile and BNEP protocol. A host may be allocated several link-local IP addresses e.g. if it has interfaces to several physical links (e.g. two Bluetooth PAN:s). This is referred to as multihoming. Multihoming gives rise to the problem of scoped addresses, which means that addresses must be associated with the correct interface. In other words, a host must select the correct source address, which is proper for the interface in question, when sending a packet. Likewise, a host must select the correct interface when sending a packet to a certain destination address, because this address is proper only on the correct interface. This problem can be solved by exposing the address-interface associations to the application level and letting the application handle the problem. [63]

5.4 Data Transmission

The PAN profile uses Ethernet packets as specified in the PAN and BNEP specifications. NAP:s or GN service nodes act as Ethernet bridges and implement the relevant parts of the IEEE 802.1D standard [70] as specified in the PAN profile specification.[3] Only one BNEP connection is allowed between a PANU and a GN. The GN acts as a bridge with BNEP connections to the PANU:s. Incoming packets are forwarded by the GN over all BNEP connections unless there is an entry in the filtering database regarding the destination address in question. Each time the GN receives a packet, it checks whether there is an entry for the source address of the packet. If there is no entry, an entry is added to indicate that the address in question is reachable on the BNEP connection is question. Once an entry regarding an address is added to the database, the GN forwards subsequent packets destined to the address in question only to the BNEP connection specified in the database. In this way, the GN learns the structure of the network and redundant traffic is avoided.

5.5 BNEP

According to the BNEP specification: "Bluetooth network encapsulation supports the same networking protocols that are supported by IEEE 802.3/Ethernet encapsulation." The provided capabilities and the packet formats are based on the Ethernet standard [71] and the IEEE 802.3 standard [72]. The BD_ADDR:es are used for addressing in BNEP. In addition, IEEE 802.3 multicast and broadcast addresses may

[3]For clarity, only GN:s are discussed in this section from now on. However, the rules described in this section apply for NAP:s as well as for GN:s.

be used. BNEP uses connection oriented L2CAP connections with a minimum MTU (maximum transmission unit) requirement of 1691 bytes. This means that devices communicating with the BNEP protocol must be able to handle L2CAP packets of 1691 bytes. The BNEP packet format and the encapsulation of an Ethernet packet's payload is presented in Figure 20. In the figure, the L2CAP header is also shown. The length of the BNEP header may be up to 15 bytes, which is the length of the BNEP_GENERAL_ETHERNET packet type header. Compressed header formats are also available, but they are not specified for broadcast or multicast usage.

[73]

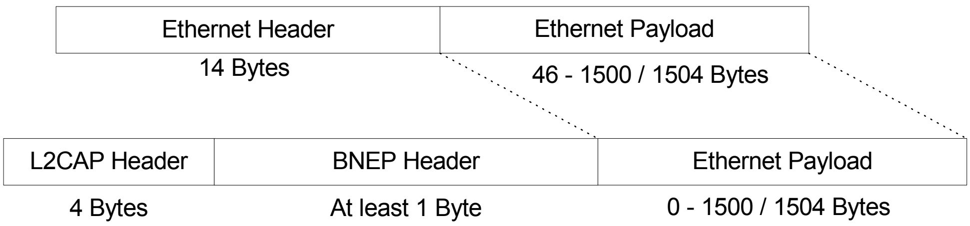

Figure 20: A BNEP packet with an Ethernet packet payload sent using L2CAP [73]

5.6 Chapter Summary

This chapter described the Bluetooth PAN profile as an enabler for IP communications between Bluetooth devices. The PAN profile merely provides a platform for using IP in a similar manner as in conventional Ethernet networks. The profile is thus rather simple. The major limitations of the PAN profile are the lack of automatic network formation and lack of support for inter-piconet communication i.e. scatternets. These issues must thus be solved on the application layer.

The terminology, architecture and capabilities of the underlying Bluetooth Core was also described to the extent necessary for understanding the PAN profile. The HCI was also briefly described in order to shed some light on the capabilities provided to the host above that interface.

6 USE CASES AND REQUIREMENTS

6.1 Use Cases

The OMA PoC Requirements [36] document describes a number of use cases, some of which apply to PoB as well. The following two use cases are quotations therefrom:

Where to Eat Lunch

> *This is a basic use case, which describes the PoC group call feature. In the example one group member (Julie) initiates a group call by selecting the appropriate group from her PoC terminal and starts talking to other group members. Other members can hear the speech from their terminals without a need to answer. Other group members can take part in the conversation by pressing the PoC button and waiting for a permission to speak*
>
> - *Julie wants to have lunch with her workmates and she makes a call to them using PoC.*
> - *She queries whether they would like to go to city, as she is not very fond of the menu at the nearby restaurant where they usually eat.*
> - *Julie first selects the group of people she wants to talk to and presses the Push to talk button.*
> - *After a short while the PoC server will send her a permission to speak notification and she can start talking to the group and all the people in the group will hear her.*
> - *Everybody in the group agrees that the menu at the local restaurant is not very attractive and they reply one-by-one to the group that they agree and would like to go to the city with Julie.*

Shopping at the Mall

> - *A group of people shopping together decided to keep in touch with each other using a PoC service to inform on the most challenging bargains. Therefore, one of them, Mary, requests the PoC service provider to set-up the PoC service for them.*
> - *As soon as the PoC service provider has set up the service, all the invited people get an indication on their terminal, asking whether they would accept the service. This service invitation contains the name of the inviting host (Mary) as well as the name of the group: "SHOPPING LIKE CRAZY". In addition, the PoC service provider has relayed the right to accept additional participants to Mary.*

- *Most of the invited people accept the service offer, becoming participants in the PoC group. However some do not accept, since they have other preferences. In the department store they meet another friend who would like to join. Being given the name of the group he sends a request to Mary to join the group. Mary allows him to join.*
- *Susie suddenly discovers an extremely cheap shoe shop, which she simply has to tell her friends of. So she pushes the talk button.*
- *As someone is speaking right now and Manfred had pushed the button before, Susie's request to speak is queued. Hearing Manfred talk, Susie realizes that Manfred is already talking about this shoe shop. So she cancels her request to speak. Alternatively, after Manfred had finished speaking, Susie would have received an indication, that she is now "on air".*
- *The voice is immediately distributed to the other participants. For the listeners, when they are ready to listen, their terminals receive the voice of the speaker without prior indication.*
- *One of the participants receives an incoming phone call. As determined by the preferences of the owner, the phone switches to "not ready to listen" mode of the PoC service. In this mode the PoC service silently continues in the background, after the end of the phone call the participant decides to return to listening to the PoC service.*
- *After a while Manfred gets bored with all this gossip and decides to leave the PoC group. He simply sends the unregister-request indication to the PoC service. The rest of the participants get an indication that Manfred has left the PoC group.*

In addition to the OMA PoC use cases above, many other use cases are also possible. One additional use case is presented below.

At the Beach

It's a hot Saturday in July. Jack, Bob, Mike, Jason, Maria, Trent, Julia, Billy and Jennifer are headed for the beach to spend the day enjoying the sun, swimming, socializing and maybe playing some beach volley. The beach is crowded with people.

1. Jack, Bob, Mike, Jason, Maria, Trent, Julia, Billy and Jennifer arrive at the beach. Jack and Bob decide to go and buy something to drink. The others head to the sand to find a nice spot.

2. Jack and Bob have bought their bottles of cola and head for the sand to join the others. Jack selects the "Friends"-group, pushes the PTT-button and says: "Where are you guys, I can't see you!"

3. The others have found a nice spot nearby and they have sat down and made themselves comfortable. They here Jack's call and Billy replies: "I can see you walking, we're right next to the group with that huge green parasol!"

4. Jack and Bob hear Billy's call and Jack replies: "OK, I see the parasol!"

5. Jack and Bob head towards the parasol. Suddenly, an ironic voice comes from Jack's mobile "Hey, are you blind? I've been waving at you like heck but you just keep ignoring me!"

6. Jack recognizes the voice as Kevin, his classmate. (Jack, Kevin and some other people from their class have formed another PoB group, which Jack has defined as a secondary group.) Jack looks around and see's Kevin. Jack and Bob then go to Kevin and sit down for a while to chat.

7. After a while Bob's mobile says: "Are you guys lost already or what?" It's Jason eager to have a sip from Jacks cola.

8. Bob replies: "We ran into Kevin, Jack's classmate. We'll be there in a while."

9. Jason replies: "OK, tell him to join us, we're about to dive in."

10. Kevin decides to join Jack and the others. Jack pushes the PTT-button and says: "Wait up, we'll join you right away! Kevin just picks up his stuff."

11. Jack, Bob and Kevin join the group and they all go swimming.

The above use cases may not be implementable exactly as such in PoB. The limitations of the Bluetooth domain may cause problems that require the user interface and the functionalities to be somewhat different in PoB. For example simultaneous sessions may be too complex to implement. The electromagnetical environment may reduce the effective range to the extent that some of the above mentioned use cases become infeasible. These problems and possible solutions are discussed in the following chapter.

6.2 Performance Requirements

The OMA PoC Requirements document specifies certain performance requirements in order to ensure a satisfactory quality of experience (QoE) for the user. QoE itself is naturally very subjective and difficult to measure consistently. However, some of the factors impacting QoE are measurable and the performance requirements are therefore based on these. These measurable factors are presented below.

QoE1: Right-to-Speak Response Time

The right-to-speak response time when establishing a PoC session. That is, the time from the instant the user initiates a session to the instant he/she receives a "right-to-speak" indication. This time should typically be less than 2.0 seconds.

QoE2: Start-to-Speak Response Time

The start-to-speak response time in an active PoC session. That is, the time from the instant the user request the right to speak to the instant he/she receives either a start-to-speak indication, queuing indication or reject indication. This time should typically be less than 1.6 seconds.

QoE3: End-to-End Channel Delay

The end-to-end channel delay, i.e. the time from the instant one PoC session participant starts to speak until the instant the recipient(s) starts to hear the speech in question. This time should typically be less than 1.6 seconds.

QoE4: Mean Opinion Score

The perceived voice quality. The quality of the voice signal conveyed over a PoC session is dependent of several factors such as signal level, codec characteristics and RF channel conditions. The mean opinion score (MOS) is a commonly used measure for evaluating voice quality. The requirement set by OMA is that MOS should be at least 3 at a bit error rate (BER) of 2 per cent or less.

The requirements above can be unambiguously measured in a test environment. One additional requirement that is dependent on user behavior, is also defined, namely the turnaround time. The turnaround time is defined as the time from the instant a user releases the floor after he/she has stopped speaking until the instant the same user hears the beginning of a reply from another user. That is, the time from e.g. releasing the PTT button until hearing the following user's voice. The turnaround time should typically be less than 4 seconds.

[36]

6.3 Chapter Summary

This chapter gave a few examples of PoB use cases and presented the performance requirements of OMA PoC. The intention with the use cases was to show that some PoC use cases are played out within a rather small area and the range of class 1 Bluetooth devices could thus be sufficient. Several other use cases could be outlined for PoB. The OMA PoC requirements have been defined in accordance with what has been considered as appropriate both from a technological and a usability point of view. Since the goal is that PoB should provide a user experience which is as similar to that of PoC as possible, these requirements set the QoE targets for PoB as well. Fortunately, the performance requirements set out for OMA PoC are not very strict because the performance of current GPRS in GSM is not excellent.

7 Push-to-Talk over Bluetooth

In this chapter, a proposal for deploying a PoC-like PTT feature over the Bluetooth PAN profile is outlined. The problem is approached from a PoC point of view, i.e. whenever possible the features and characteristics of OMA PoC are adhered to and modifications are made only when required by the limitations of Bluetooth. In this way, the user experience of PoB resembles that of PoC as much as possible. Modularity benefits may also be gained, if PoC software components may be reused in PoB.

In addition to the solutions conceived as the preferred ones, other alternative solutions are also presented together with a discussion rationalizing the selection of the preferred solution. To limit the discussion to a reasonable amount of issues and alternatives on different levels of detail, the issues are discussed in a hierarchical manner wherein the subproblems of the alternative solutions are not considered in detail. That is, only the subproblems arising from the preferred solution of a higher level problem are discussed in more detail.

7.1 Features

The target is that all of the relevant high level features of PoC as described in Sections 6.1 and 6.2 of the OMA PoC requirements document [36] will be implemented in PoB as well. Pre-arranged, ad-hoc, and chat groups would thus be supported as well as instant personal alerts. However, some of the functionalities provided by the PoC servers may be curtailed in PoB. In addition, restrictions related to the implementation environment may further restrict some features such as multiple active groups and thus simultaneous sessions.

7.2 Transport Plane Protocols and Voice Coding

7.2.1 Preferred Solution: Employing the OMA PoC Protocols and Codec

Since the Bluetooth PAN profile provides a platform for IP communications, there are no obstacles for employing the IP/UDP/RTP protocol stack of OMA PoC in the Bluetooth environment providing that the throughput is sufficient. The AMR codec may thus also be used without complications. The benefit of this approach is that the same software components may be used in both PoC and PoB and the resources required for development and testing are reduced. In addition, this approach supports interoperability between PoC and PoB, as discussed in Chapter 8.

The throughput requirement of the above protocol stack depends on the AMR interface format and mode and on how many AMR frames are conveyed per packet. The throughput requirements for AMR IF2 with different AMR modes and frames per packet values are presented in Figure 21. (See Appendix A for calculations.) The highest requirement with these variables is 36.8 kbps, which occurs for AMR mode 12.2 with only one AMR frame per RTP packet. The lowest requirement on the contrary is 6.08 kbps, which corresponds to AMR mode 4.75 and 50 AMR frames per RTP packet, which implies a buffering delay of 1 s at the sender.

According to the PAN profile, broadcasting is implemented by multiple unicasting[4] on the baseband level. The throughput requirement for broadcast is thus multiplied by a factor corresponding to the number of slaves, if the master is the originator of the data. If the originator is one of the slaves the corresponding factor is $s - 1$, where s is the number of slaves in the piconet, since the data is not sent back to the originator.

One additional factor affecting the performance requirements in a scatternet is the bridge scheduling algorithm. In a scatternet, some devices act as bridges between two piconets and they must thus balance their presence in the piconets since they cannot be active in several piconets simultaneously. Scheduling refers to the method according to which a bridging device allocates its time between the two piconets. Since scheduling is not defined in the Bluetooth specifications, a number of schedul-

[4] In other words, broadcasting is implemented by sending data to each recipient separately.

ing algorithms have been proposed. An overview of these is given in a report by Misic et al [74]. Scheduling algorithms can be divided into algorithms with uninterrupted and interrupted transmission. Uninterrupted transmission means that once a need for transmission is detected in one piconet, a bridge remains active in this piconet until the need is fulfilled, i.e. all incoming data is accepted by the bridge. On the contrary, interrupted transmission means that a bridge may be inactive in any of the two piconets for a maximum period only. If this period is exceeded, the bridge must interrupt the transmission of data and become active in the other piconet. Since a bridge is active only in one piconet at a time, access delays occur. A sender must wait for the bridge to become active in the piconet in question, if the bridge is currently active in another piconet. In order to take access delays into account, the estimated throughput requirement for bridges is doubled compared to non-bridge devices. This estimate is very rough and is based on the assumption that a bridge spends half the time in each piconet and the time for transmitting data is thus halved. In the case a device is both a master and a bridge, the worst case throughput requirement can thus be roughly approximated by multiplying the throughput requirements of Figure 21 by a factor of $s + 1$.

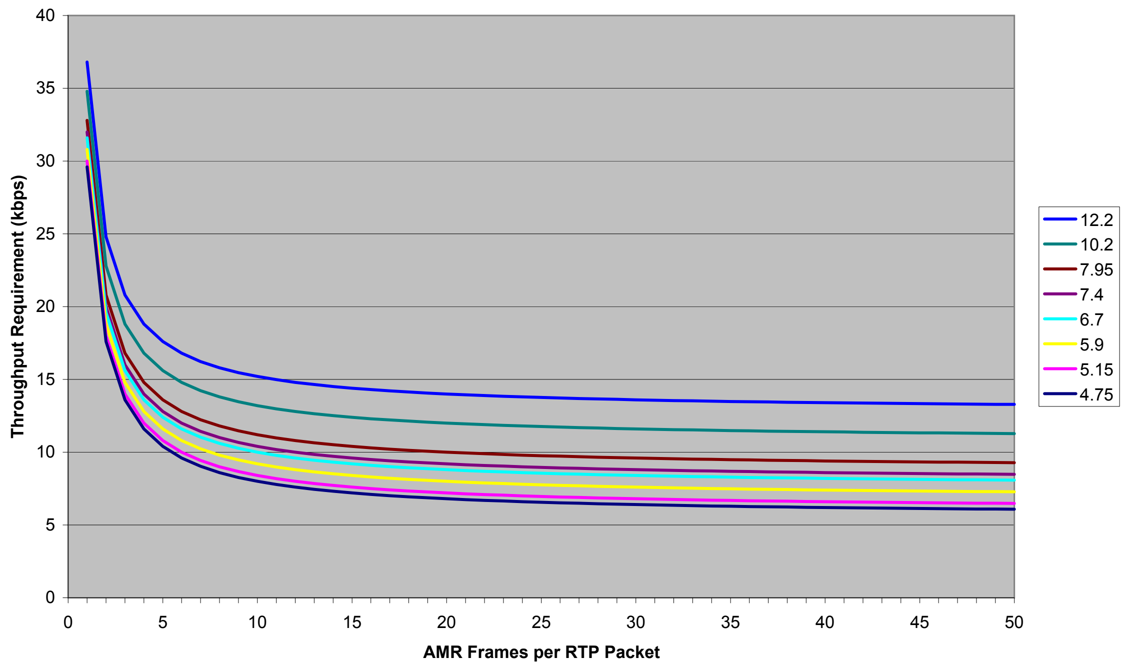

Figure 21: Throughput Requirements for Different AMR Modes with Different Numbers of AMR Frames per RTP Packet

The theoretical maximum throughput of a Bluetooth ACL link is dependent on the ACL packet type. For DM1 (Data - Medium Rate 1) packets the maximum rate is 108.8 kbps and for DH5 packets (Data - High Rate 5) 723.2 kbps. Zürbes [75] simulated the throughputs for ACL links with different packet types in a scenario with a number of piconets in a single room of 10 m × 20 m. In the simulations, the piconets consisted of one master with one slave, which communicate unidirectionally (master to slave) at maximum rate (i.e. each time-slot is occupied either by data sent by the master or an acknowledgement sent by the slave). The findings are presented in Figure 22. According to Zürbes, the values are valid for the case of multislave piconets as well, because only one device at a time transmits in a piconet. These results seem to suggest that e.g. the worst case requirement of 67.2 kbps[5] corresponding to AMR mode 5.15 and 10 AMR frames per RTP packet, would be achievable with DM1 packets even if 50 interfering piconets exist in the vicinity.

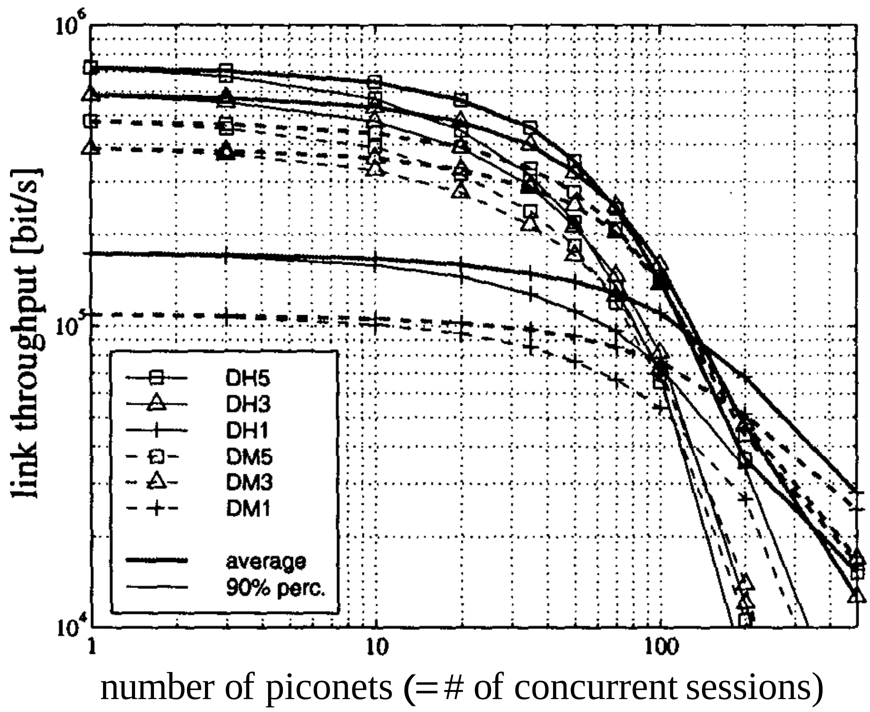

Figure 22: Average and 90% Percentile Link Throughputs for Unidirectional Traffic in an Environment with 1-500 Fully Loaded Piconets [75]

A model for calculating the throughput of a similar scenario is presented by Zanella et al [76]. The model suggests clearly lower throughput results than Zürbes simu-

[5] Assuming that the device is acting as both a bridge between two piconets and as a master for seven slaves: 8 × 8.4 kbps = 67.2 kbps

lations. In their analysis of the model Zanella et al found, however, that the model becomes less accurate as the number of interfering devices increases.

The above mentioned studies seem to suggest that the throughput requirements stated earlier are achievable if the AMR mode, number of AMR frames per RTP packet, and baseband packet are suitably chosen. However, simulations and/or tests are needed to ensure sufficient throughput.

Chen et al [77] propose an adaptive Automatic Repeat Request (ARQ) scheme for audio streaming over Bluetooth. The scheme seems to improve throughput and delay considerably in noisy environments and could thus be beneficial. The idea is basically to measure the Round Trip Time (RTT) of e.g. RTP packets and adapt the HCI Flush Timeout depending on the RTT. The disadvantage of this scheme is that it increases the complexity and buffering requirements somewhat.

In addition to the throughput requirements, delay issues must be taken into account in order to ensure that the QoE requirements presented in Section 6.2 on page 53 are met. However, the delays occurring in a Bluetooth scatternet are relatively small compared to the GPRS environment and the end-to-end requirement of 1.6 s, should give plenty of room for e.g. access and transmission delays (assuming that the amount of non-PoB traffic in the scatternet is marginal, and the access delays thus small). As an example, the tests of Kargl et al [4] produced an average delay of 28 ms, when transferring data between a master and one slave in an interference free environment. Nevertheless, tests and/or simulations are needed to ensure that the delays remain within the required limits.

7.2.2 Other Alternatives

One obvious alternative would be to design a protocol dedicated for PoB. This alternative is supported by the fact that the IP/UDP/RTP stack does include capabilities that are unnecessary in the PoB context as outlined in this thesis. For example, the 16 bit source and destination port numbers of the UDP header could easily be avoided if a dedicated protocol would be used, since a similar capability is provided by the HCI connection handles in Bluetooth. In a dedicated protocol, the total overhead and protocol methods could be optimized for PoB thus reducing the capacity requirement and possibly the processing time as well. However, the benefit of using readily available, verified and widely used protocols is considered to outweigh the benefits of a dedicated protocol. In addition, the chosen protocol stack provides better possibilities for interoperability and further developments such as multihop routing.

Another alternative would be to use the SCO or Extended SCO (eSCO) links of Bluetooth, which are specifically specified for conveying speech and other real-time data. However, SCO usage incurs some problems. One problem is that the strict timing scheme required by SCO links deteriorates the performance of simultaneous ACL links and thus stifles other data traffic on these channels. Mišić et al consider this problem and propose a scheme for SCO-like communication over ACL links, which would not affect other ACL links as much [78]. In addition, Kapoor et al [79] compared the occurring delays and the throughput of simultaneous data transfer when conveying speech over ACL and SCO links. The conclusion was that ACL links are preferable for voice data unless there is a need for extremely high quality voice. Kapoor et al also predict that the presence of SCO links make the piconet and scatternet scheduling extremely difficult and suggest that the scheduling algorithms are far less complex if ACL links are used for e.g. voice.

Other codecs such as Pulse Code Modulation codecs could also be considered for PoB. This discussion is, however, outside the scope of this thesis.

7.3 Control Plane and Signalling

Since the OMA PoC architecture relies on client-server based signalling, which is not available as such in the Bluetooth domain, an alternative way to govern communications must be designed.

7.3.1 Preferred Solution: Peer-to-Peer

In an architectural sense, a simple approach is to avoid the client-server model altogether and utilize a peer-to-peer based signalling scheme. Instead of SIP signalling, an application specific signalling method is used on the application level. To simplify this signalling scheme as much as possible, the group creation, scatternet formation, and communication methods are designed with the objective of reducing the need for signalling traffic. This implies avoiding complexity in general. These methods are described in the corresponding Sections 7.4 and 7.5.

The advantage of this approach is that the diverse problems associated with maintaining centralized or cluster based control in a dynamic ad-hoc network are avoided. Schemes for cluster based control is discussed e.g. by Steenstrup [80], who states that the deployment of these methods has been limited to experimental setups. This technology is thus far from mature which implies a reliability risk. The disadvantage of the peer-to-peer alternative, on the other hand, is that a centralized point of control cannot be used to handle e.g. floor control. This problem is discussed in Section 7.8.

7.3.2 Alternative Solutions

Signalling over GPRS and User Data over Bluetooth

One potential alternative would be to use Bluetooth for conveying user data, i.e. the speech itself, and GPRS for handling the signalling. However, the signalling system of PoC could not be used as such, since the URI addressing scheme would not be functional. The link local addresses allocated by the Bluetooth PAN profile are not routable outside the local link, and associating an URI with such an address would not yield a successful result. Instead, a system based on BD_ADDR-addressing and a server maintaining neighbor lists and providing a routing service could be designed. A server could maintain a complete chart of the network on the basis of neighbor list notifications from the clients (i.e. PoB phones). Each client would periodically do inquiry scans to obtain a neighbor list with the BD_ADDR:es of the devices in the vicinity, and then send it to the server. By combining the neighbor lists from all active clients, the server would be able to create and maintain a relatively accurate chart of the network (and possibly estimate its dynamics by maintaining information regarding the recent mobility of clients). With this chart, the server could determine the shortest multihop path from one client to another e.g. with the aid of Dijkstra's famous algorithm [81]. (Another alternative would be to determine the shortest path including only clients with a recent history of low mobility.) A client desiring to communicate with another client could thus send a request including the BD_ADDR of the recipient to the server. As a response, the server would send a list of BD_ADDR:es constituting a path to the recipient.

This approach would bring the significant advantage of multihop communications with centralized route determination and thus means for coping with the problems caused by the network dynamics. Unfortunately, the disadvantages of this approach seem to render it infeasible. The major problem is that this approach would require the clients to frequently transmit data to the server over the GPRS network. The cost incurred by this to the users, would decimate the original benefit of PoB, i.e. the lack of cost. In addition, the companies maintaining the servers would obviously need to charge for this service, raising the end user cost additionally.

7.4 Group Formation and Addressing

According to OMA PoC, the groups are managed by the GLMS, which is accessible by the clients for group management. Obviously, maintaining such a server is not possible, if the preferred solutions of the above sections are chosen. Therefore, a new method for creating and maintaining groups is needed.

7.4.1 Preferred Solution: Bluetooth-Address Based Group Management

As mentioned in Section 1.2, the inquiry and page delays are rather long and cause thus significant delays when a piconet or scatternet is formed. One way to reduce the impact of these delays is to avoid the inquiry method altogether. This can be achieved by defining the groups by the BD_ADDR:es of the group members. A group is thus defined by a GROUP list of BD_ADDR:es.

The creation of such a list can be performed by sending an invitation message over e.g. Bluetooth, SMS, or an Infrared Data Association (IrDA) link to the desired group members. Upon reception of an invitation, the PoB application queries the user whether the group should be joined or not. If the users reply is affirmative, a reply including the BD_ADDR is sent as a response to the invitor. When all invitees have replied, the group is defined as a GROUP list containing the BD_ADDR:es. To complete the group formation, the list of BD_ADDR:es is sent from the invitor to all the other group members. Groups may also be modified in a corresponding manner later to add or remove members.

An example of this procedure is presented below and in Figure 23:

1. Jack creates a group by sending invitations to his friends Bill, Bob and Laura over Bluetooth, IrDA or SMS with the PoB application.

2. The PoB applications of Bill's, Bob's and Laura's phones receive the invitation and query the user "Accept PoB-group invitation from Jack?" (Where Jack is e.g. the Bluetooth device name [59] of Jack's phone). If the user accepts, a message is sent to Jack containing the BD_ADDR of the user's phone. Jack's phone receives thus the BD_ADDR:es of Laura's and Bill's devices and the group is defined on Jack's phone. (But Bill and Laura do not have each other's BD_ADDR:es.)

3. Jack's phone sends a message to Bill and Laura containing the group i.e. the BD_ADDR:es of Jack, Bill and Laura. Now all group members have the group defined as a set of BD_ADDR:es on their devices

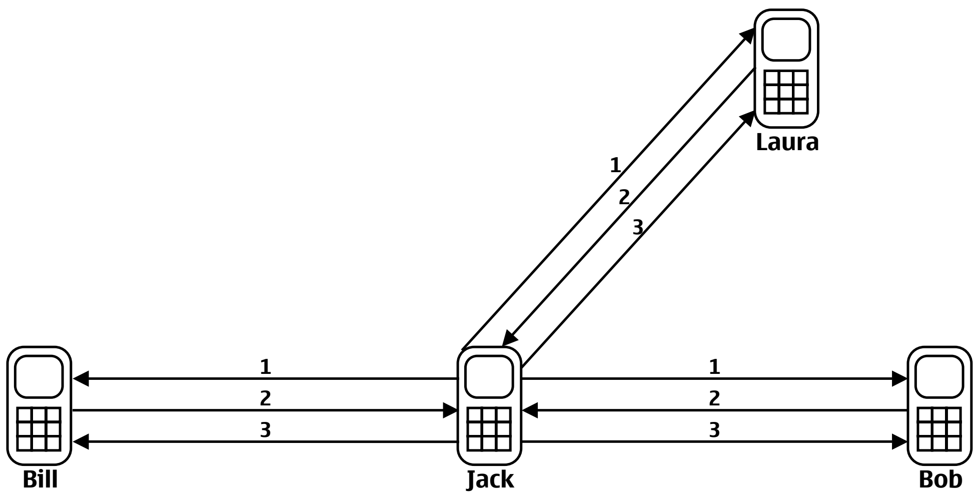

Figure 23: An example of how a group can be formed

One problem of this approach is that mischievous invitations with spoofed Bluetooth device names are possible, if the invitation is sent over Bluetooth. For example, a person can set the Bluetooth device name of his/her phone to a name which an invitee associates with another person. If the invitee accepts the invitation, the spoofer can then send mischievous PoB messages to the victim. This problem is considered minor, since the victim can easily remove the group in question. Since the invitor and invitee are normally near each other, the originator of an invitation can usually be confirmed verbally. Such attacks can also be avoided by block or allow lists based on BD_ADDR:es.

The group members could also exchange e.g. a randomly generated encryption key in the group formation phase. This could be done securely over SMS or IrDA. Since the security issues are outside the scope of this thesis, these issues are not discussed further.

7.4.2 Alternative Solutions

Group Formation with Radio Frequency Identification

One potential alternative would be to exploit Radio Frequency Identification (RFID) for group formation. The BD_ADDR:es could be exchanged over RFID e.g. as presented by Hall et al [82] and Busboom et al [83]. A group could thus be formed by bringing the users' devices close to each other whereby the BD_ADDR:es can be exchanged over RFID. The complete list of a groups BD_ADDR:es could also be distributed in a similar manner. The advantages of this would be that group formation would be easier; the user merely has to select e.g. a "Form Group with RFID"-command and then let his or her friends bring their phones together. This approach would also solve the problem of mischievous invitations mentioned in Section 7.4.1. The problem with this approach is that mobile phones with RFID transceivers are not available currently.

7.5 Network Formation and Maintenance

A wide range of methods for Bluetooth scatternet formation have been proposed in recent years. An overview and bibliography of these is given in Chapter 3 of [84] and additional methods are presented in e.g. [85, 86]. This area of research has received a considerable amount of attention lately but a clearly superior solution, balancing complexity and reliability of connectivity, has not been presented. Therefore, the network formation problem is approached with an aim for simplicity in this thesis.

7.5.1 Preferred Solution: Network Formed Among Group Members

Since each device knows which devices it desires to communicate with, the preferred solution is to establish connections with these devices only. The scatternet is thus formed by including only devices with BD_ADDR:es that are on the GROUP list (see Section 7.4.1). However, in order to enable easy broadcasting, loops are not allowed in the network. An example of such a network is presented in Figure 24. In the figure, the names denote group members' devices, the single letters outsiders'[6] devices, the arrows Bluetooth ACL links, the arrowheads the slaves, and the arrow color different piconets. As can be seen, all group members but one ("Rick" in the upper right corner) are connected to the Scatternet. A method for forming such a network is presented in the flowchart of Figure 25 and explained below. In order to form the desired network, each PoB device runs this method.

For clarity, it is assumed in the description that a device does not have any existing ACL links when the method is initiated. In some cases, however, other applications may have established ACL links before the PoB application is initiated. This special case should be taken into account when implementing the method.

Each device needs to keep track of which devices are reachable via the scatternet and which are not. This can be done by maintaining CONNECTED and UNCONNECTED lists, which include the BD_ADDR:es of the devices that are and are not connected to the scatternet respectively. Each entry in the CONNECTED list is associated with two additional BD_ADDR:es which indicate via which device

[6] The term "outsider" refers to people whose devices are not members of a particular PoB group.

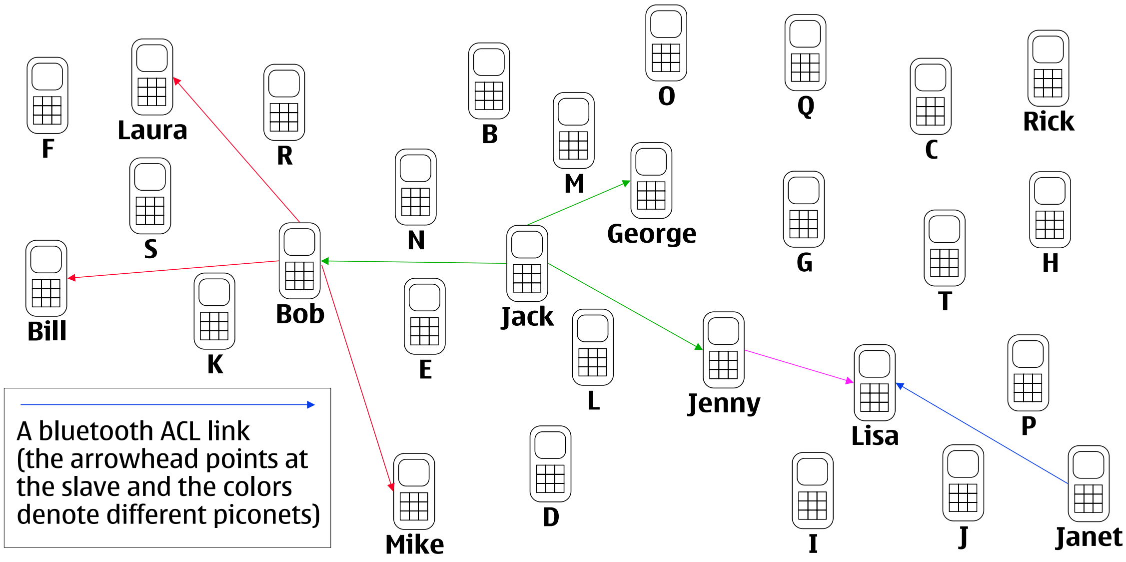

Figure 24: An example of a Scatternet According to the Presented Method

the device in question is reachable and which device initiated the connection that connects the device in question to the network. An example of a CONNECTED list is presented in Table 7, which corresponds to the perspective of Jack's device in Figure 24. The left hand column indicates which devices are reachable and the middle column indicates via which link the device in question is reachable. The right hand column is related to loop avoidance which is discussed in more detail later on in this Section. As an example, the last entry of Table 7 indicates that the device with the BD_ADDR "00001000..." (i.e. Janet) is reachable via the device with the BD_ADDR "00000011..." (i.e. Jenny). It should be emphasized that the device maintaining the CONNECTED list of Table 7 (i.e. Jack's device) has a *direct* ACL link to the device with the BD_ADDR "00000011..." (i.e. Jenny) and all other devices in the middle column. However, the devices in the left hand column are not necessarily *directly* connected with device "00000000..." (i.e. Jack). In order to save memory, Bluetooth HCI Connection Handles may be used instead of the Reachable Via BD_ADDR:es of the CONNECTED list.

A device initiates the method by creating an UNCONNECTED list by copying the GROUP list. Then the device attempts to connect to one of the devices on the UNCONNECTED list by initiating a Bluetooth Page process.[7] If the attempt is

[7] It may be beneficial to perform Inquiry prior to Paging in order to avoid unnecessary Pages. However, since the Inquiry procedure of Bluetooth 1.1 is rather time-consuming it may not be

Table 7: An example of a CONNECTED list (The Bluetooth device addresses are simplified for clarity.)

Connected Device (BD_ADDR)	**Reachable Via (BD_ADDR)**	**Connection Initiator (BD_ADDR)**
00000000... (Jack)	00000000... (Jack)	
00000001... (George)	00000001... (George)	00000000... (Jack)
00000010... (Bob)	00000010... (Bob)	00000000... (Jack)
00000011... (Jenny)	00000011... (Jenny)	00000000... (Jack)
00000100... (Mike)	00000010... (Bob)	00000010... (Bob)
00000101... (Laura)	00000010... (Bob)	00000010... (Bob)
00000110... (Bill)	00000010... (Bob)	00000010... (Bob)
00000111... (Lisa)	00000011... (Jenny)	00000011... (Jenny)
00001000... (Janet)	00000011... (Jenny)	00001000... (Janet)

unsuccessful, another device from the UNCONNECTED list is Paged. This cyclic Paging procedure continues until the UNCONNECTED list is empty or the device is a master of seven slaves[8]. To reduce power consumption and avoid causing interference, the frequency of Page attempts can be gradually reduced e.g. by exponential backoff (i.e. by waiting for a certain period of time between Page procedures and doubling this period every time the list wraps around). The user may, however, be allowed to reset this period in order to "refresh" the network.

Upon a successful Page process and the creation of an ACL connection, the devices exchange REACHABLE messages to inform each other about their connectivity information. The REACHABLE messages include a list of BD_ADDR of the devices that are reachable via the device sending the message and the corresponding Connection Initiator information. In other words, a REACHABLE message includes the left hand column and right hand column of the senders CONNECTED list. After the exchange of REACHABLE messages, the devices can thus update their CONNECTED and UNCONNECTED lists by adding and removing the BD_ADDR:es respectively. The REACHABLE messages of two newly connected devices should never include overlapping BD_ADDR:es because this would imply that a loop has been formed. Loop avoidance is discussed in more detail later on.

feasible.

[8]It may be beneficial to limit the number of slaves to less than seven, in order to improve performance.

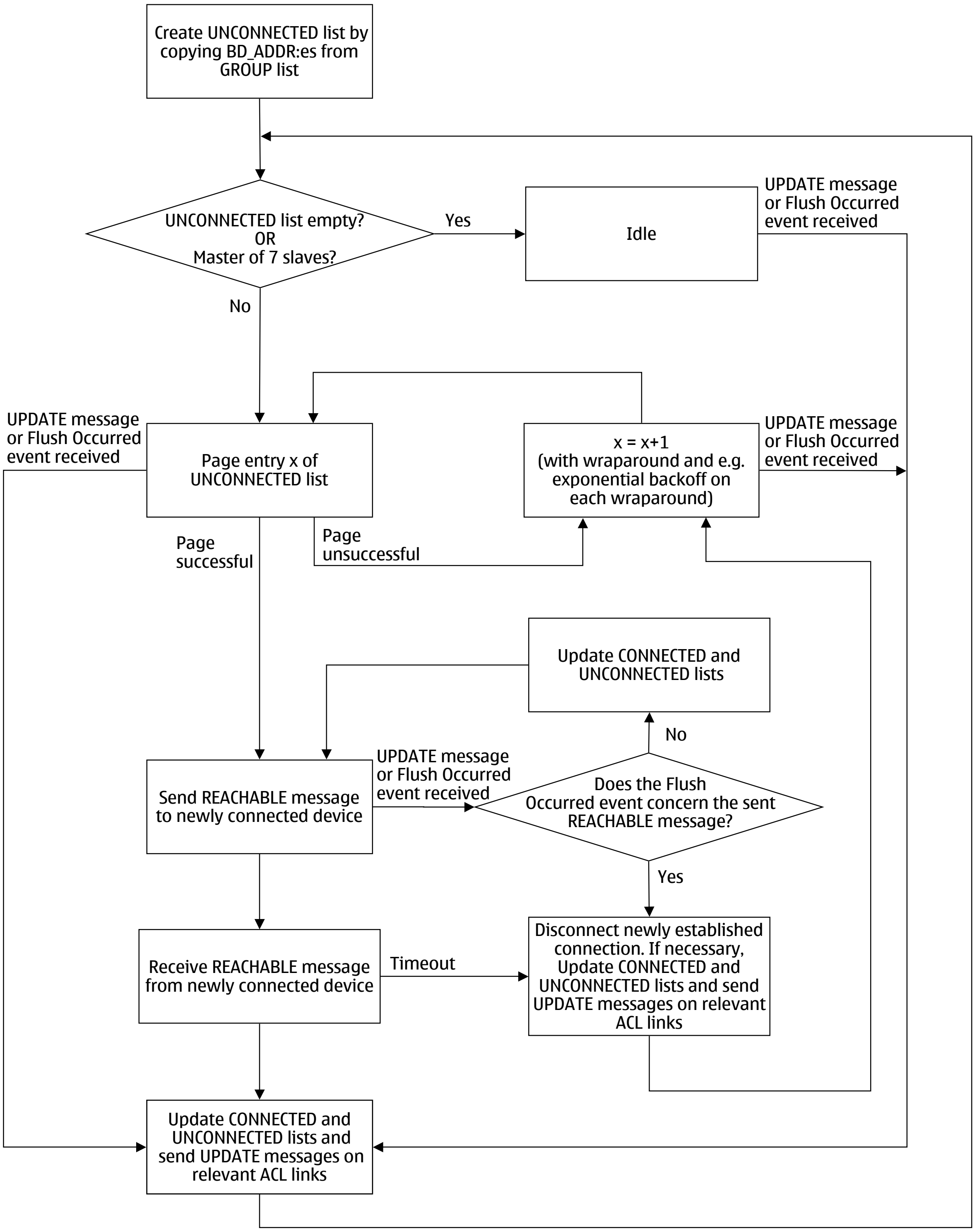

Figure 25: Network Formation and Maintenance Method

When a connection is established or broken, an UPDATE message is sent to all directly connected devices (i.e. the PoB devices to which a direct ACL link exists). The UPDATE message notifies other devices of changes in the CONNECTED list. If a new connection has been established, the UPDATE message contains the

BD_ADDR of the newly connected device, the list of BD_ADDR:es of the REACHABLE message received from it and the corresponding Connection Initiator information. If a connection is broken, the UPDATE message correspondingly contains the BD_ADDR of the lost device and the possible BD_ADDR:es of the devices which used to be reachable via the lost device. On the basis of the UPDATE message, the recipients can update their own CONNECTED an UNCONNECTED lists and thus keep track of which devices are reachable. An UPDATE message includes a list of BD_ADDR:es with indications whether they should be added or removed from the recipients CONNECTED list and, in the case of adding, the Connection Initiator information. UPDATE messages are also sent on relevant links when the CONNECTED list is updated on the basis of a received UPDATE message. Information regarding changes in the network are thus conveyed to all the devices connected to the network.

As mentioned in Section 5.1.3 a Flush Occurred event is given on the HCI when a message has been discarded because transmission has failed. On the other hand, a link is regarded as broken if no traffic exists on it for a certain period. To maintain links and keep track of the link statuses, some traffic thus needs to be sent periodically over all links, unless data or control messages are conveyed. These link maintenance messages should preferably be very small in order to reduce battery consumption and common channel interference and avoid congestion. The link maintenance messages maintain the links and cause Flush Occurred events on the HCI when a device is no longer reachable. When this event occurs, the CONNECTED and UNCONNECTED lists are updated and SETUP messages are sent. However, since a device may be a member in several piconets, it is possible that a device is not reachable because it is currently sending or receiving data in another piconet [87]. To avoid unnecessary disconnections, the period between the maintenance messages should thus be optimized. It is possible that in order to ensure stable functioning, a link cannot be regarded as broken until a number (e.g. three or five) of sequential maintenance messages have been flushed (in Figure 25 and the discussion of this Chapter, this alternative is not used).

In most cases, the CONNECTED and UNCONNECTED lists are updated and the UPDATE messages sent right away when an UPDATE message or Flush Occurred event is received. Any other tasks are thus interrupted as presented in Figure 25.

However, if the device has just connected to another device and the REACHABLE message has not yet been successfully sent, the procedure differs somewhat. If a Flush Occurred event is received in this situation, the device should first check whether the event concerns the REACHABLE message that is currently being sent (e.g. by comparing the Connection Handle parameter of the event with the Connection Handle associated with the BD_ADDR of the device to which the REACHABLE message is being sent). If this is the case, the newly established connection should be disconnected and the device should return to the cyclic Page procedure. If, however, the interrupt is caused by a received UPDATE message, the device should flush the REACHABLE message it is currently sending, update its CONNECTED and UNCONNECTED lists, and then send a new REACHABLE message according to the updated CONNECTED list. In this way some unnecessary traffic can be avoided.

Similarly, if the device has sent a REACHABLE message to a newly connected device but a REACHABLE message is not received as a response within a certain period of time, the connection is disconnected. In this case the CONNECTED and UNCONNECTED lists may need to be updated and UPDATE messages sent, if an UPDATE message was received while awaiting a REACHABLE message from the newly connected device.

Loop Avoidance

In order to avoid and eliminate loops, the devices should adhere to the following rules.

- A device should not Page devices that are on its CONNECTED list.
- A device should ignore Page requests from devices that are on its CONNECTED list.

- If a device is in the process of establishing a connection with another device (i.e. it has received or sent a Page request but has not yet received e.g. a HCI Connection Complete event and exchanged REACHABLE messages) the device ignores all incoming Page requests.

- If a device receives an UPDATE message including an entry that is in conflict with an entry in its current CONNECTED list, the conflict must be resolved e.g. by prioritizing the entry with the larger BD_ADDR value in the Connection Initiator field. For example, if a device receives an UPDATE message indicating that a BD_ADDR should be added to the CONNECTED list but the identical BD_ADDR is already listed on the CONNECTED list, the Connection Initiator values of the CONNECTED list and the UPDATE message must be compared. If the Connection Initiator value of the UPDATE message is smaller, the UPDATE message is considered incorrect and its originator is sent a correct UPDATE message, which includes the correct data from the CONNECTED list. Correspondingly, if the Connection Initiator value of the UPDATE message is larger than that of the CONNECTED list, the UPDATE message is considered correct. In this case the CONNECTED list needs to be corrected and corresponding UPDATE messages must be sent over the relevant connections. [9]

These restrictions do not apply to other applications running besides the PoB application. However, the PoB application should not use the connections established by other applications unless a method for handling this special case is used.

Network Formation Delay

One issue related to network formation is when the network should be formed. One alternative is to form the network as soon as the PoB application is initiated. The problem with this alternative is that maintaining the network consumes some power since some periodical traffic is needed in order to maintain the links and keep the network updated. The other extreme alternative is to form the network on demand

[9] Despite the other rules, loops may form e.g. in case two separate networks are merged by two separate paging procedures at different locations in the networks. This rule is therefore needed to eliminate such loops. Since it takes some time to eliminate these loops, a timeout may be needed after each reception of an UPDATE message in order to ensure proper functioning.

i.e. when the PTT button is pressed. Even though this alternative would be optimal from a power consumption point of view, it incurs the network formation delay issue.

According to the calculations of Busboom et al [83], the average Paging delay without prior Inquiry is 1.28 s for Bluetooth version 1.1, if transmission is error free and the devices do not have any SCO links. Kargl et al [4] reached a corresponding value of 1.1 s in their tests. Even though the Paging delay is approximately halved in Bluetooth version 1.2 [7], the delay is unacceptably long for the alternative of creating the network on demand. Therefore, a solution balancing between the two extremes of on-demand and immediate network formation is selected.

The proposal is that the network is formed on-demand when the user either selects e.g. a "Connect" command or presses the PTT button. When the network has been formed, the user can be given e.g. a "Connected to PoB Network"-notification. The established network may then be maintained until a period of e.g. 15 or 30 minutes has elapsed without any incoming or outgoing PoB traffic. This period may be user definable. When the period has elapsed the user can be given e.g. a "Disconnection from PoB network in 30 seconds! Continue? Yes/No"-indication. The user can thus be given the chance to maintain the network for another period by selecting the "Yes"-option.

The advantage of the method described above is its simplicity compared to alternative solutions. The disadvantage is that due to implementation restrictions, it is possible that a device is left outside the scatternet even though it is within Bluetooth range of a device in the scatternet. Currently available Bluetooth chips provide only limited scatternet support and may thus be unable to connect with an unconnected device.

Additional Utilizations

The network formation method presented in this section may also be utilized for realizing other applications and features besides PoB. Examples of such applications are multiplayer gaming, chat, and shared folders and calendars. In multiplayer gaming, the actions performed by different players may e.g. be broadcasted to the other game participants over a scatternet formed according to the method. The real-time and delay requirements associated with many games may, however, cause problems in the Bluetooth environment. On the other hand, this should not be an issue for turn-based games, such as strategy, role-play, and board games. A text-based chat feature could also be implemented according to the guidelines of PoB. The talk bursts would simply be replaced by text messages. Shared folders may be used for sharing pictures, e-mail-type jokes or other files. A user may store such files in a shared folder on her mobile phone whereby group members can download them or automatically update their own folders to include these files. Calendar information could be shared in a similar manner. If a group of friends or colleagues would like to make an appointment with each other, calendar data may be exchanged between group members in order to find a time that suits all members.

7.5.2 Alternative Solutions

Scatternet Formed with any Nearby Devices

On alternative would be to use e.g. one of the methods referred to in Section 7.5. The network would in this case include any devices willing and able to connect to the scatternet, including both group members' devices and bystanders' devices. Once the network has been formed, e.g. AODV multicasting or some other routing protocol for ad hoc networks could be used. The advantage of this approach would be that some group members' devices that would not be reachable with the method presented in Section 7.5 could possibly be reachable, since outsiders' devices could also be used for conveying data. (As an example, in Figure 24 devices Q and C could be used to reach Rick's device.). However, the disadvantages are the complexity and doubtful reliability of both the network formation methods and AODV.

Network of Group Members Extended with Other Devices

A hybrid of the two alternatives presented above would be to rely primarily on the method presented in Section 7.5 and extend it when necessary by including outsiders' devices, as in the above alternative. First a scatternet would be formed with any available devices belonging to the group members. If unreachable devices still remain in the UNCONNECTED list, an AODV-like algorithm would be used to locate them via outsiders' devices. Paths including outsiders' device(s) would then be maintained, in order to be able to communicate with these (electromagnetically) more distant group members. The advantage of this approach would be that the range perceived by the user would be extended further. The disadvantage, however, is that exploiting outsiders' devices increases their power consumption and incurs thus the question of whether outsiders would allow such exploitation. A reciprocity policy (allowing the own device to be exploited by other devices that in turn allow others to exploit themselves) might serve a solution to this problem. However, this may make the settings and power consumption characteristics of PoB confusing from the user perspective.

7.6 Communication

Since the network formation method of Section 7.5 creates a network that contains all the reachable group members without allowing loops, the communication scheme itself is quite simple.

7.6.1 Preferred Solution: Repeaters and Broadcasting

Since the formed Scatternet consists of group members only, broadcasting can be used to convey PoB data. The broadcasting capability of the PAN profile [61] is thus used to enable IP broadcasting [88] to all devices in a particular piconet. Since the PAN profile does not support Scatternets, the inter-piconet communication must be handled on a higher layer e.g. by the PoB application. A straightforward way to solve this problem is to let every device that is a member in several piconets act as a repeater between the piconets. In other words, any PoB messages (both speech data and control) that are received from one piconet are broadcasted on the other piconets. In this way, all messages are broadcasted over the entire scatternet. The obvious advantage of this solution is its simplicity. A disadvantage is, however, the possibly higher complexity in multiple group operation and handling of simultaneous sessions as discussed in Section 7.7, compared to other alternatives.

7.6.2 Alternative Solutions

One alternative would be to use some other network formation method, which would allow both group members' and outsiders' devices to join the scatternet, and to use e.g. IP multicasting [89] for conveying messages between group members. However, IP multicasting requires the Internet Control Message Protocol [90, 91], which increases complexity and control traffic.

7.7 Multiple Group Operation

OMA PoC specifies multiple active groups and simultaneous sessions as described in Section 4.8 on page 32. Since the broadcasting method of the preferred solution conveys any sent data to all scatternet members, a method for handling multiple PoB group operation and simultaneous PoB sessions is needed. However, since simultaneous sessions is an optional feature of OMA PoC, it may be omitted in PoB as well.

7.7.1 Preferred Solution: Separate Scatternets for Separate Groups

One way to handle multiple group membership is to maintain completely separate scatternets for different groups. That is, if a device is a member in several groups it maintains separate piconets and separate CONNECTED and UNCONNECTED lists for each group and distinguishes between these groups when communicating. Since one device can be a master in one piconet only, this approach limits the network formation somewhat. One group should preferably be set as the primary group. In this case, a device primarily forms and maintains the primary group scatternet and then attempts to join secondary group scatternets. When a device is a master in a piconet consisting of members of the primary group, it cannot, however, actively Page devices in a secondary group. Responding to Page requests from other devices of the secondary group is, however, possible, and thus also joining the secondary group(s). The advantage of this solution is that it does not require alterations to the communications method of Section 7.6. In addition, if PoB is first implemented without support for multiple groups, support for it can be added later with backward compatibility. The disadvantage is that the scatternet formation and maintenance algorithm becomes more complex, since it needs to form and maintain several scatternets.

7.7.2 Alternative Solutions

Network Formed with All Groups' Members

Another alternative for handling multiple groups is to form the scatternet with all devices belonging to any of the active groups. The formed scatternet is thus larger than the scatternet of one group only. The advantage of this solution is that the scatternet formation algorithm would be of the same complexity level as the one of Section 7.5. However, the traffic of the different groups would have to be separated e.g. by adding a group identifier to the messages. This identifier could be exchanged during group formation as explained in Section 7.4. Messages are thus broadcasted over the entire scatternet, but only the devices belonging to the indicated group decode the message and play it back to the user. The disadvantage of this solution is thereby that the communication method is somewhat more complex. More importantly, the amount of traffic in the scatternet is increased and the risk of congestion and increased delays is thus increased as well. Furthermore, the scatternet may grow very large if e.g. each member in a group is a member in other groups which are different from each other. (For example, if a group consists of five members and each member belongs to two further groups which do not overlap, the scatternet is formed among eleven groups.) In such a setup, the traffic and delays may be increased beyond acceptable levels. In addition, eavesdropping may be possible with altered PoB applications neglecting the group identifiers of messages.

7.8 Floor Control

In PoC, floor control is one of the services provided by the PoC server. In the PoB environment, a similar method is needed for controlling who is allowed to talk and when in order to avoid talk burst collisions.

7.8.1 Preferred Solution: Request to Speak

One way to solve this problem is to broadcast a time stamped Request-to-Send (RTS) message, when the user pushes the PTT button. When such as message is received, the recipients await the incoming talk burst and refrain from sending RTS:s themselves. After sending an RTS message, the originator waits until a timeout expires in order to ensure that incoming RTS:s were not underway when the RTS message was sent. If an RTS message is received from another device within this timeout period, the time stamps are compared and the sender of the earlier message takes the floor.

The time stamping procedure naturally requires a way of synchronizing the time stamping clocks or counters of the PoB devices. This can be done by sending synchronization data periodically (possibly piggybacked in link maintenance or other messages). If two users press the PTT buttons of their devices within an interval of say 10 ms, the users perceive the presses as more or less concurrent. Therefore, the accuracy requirement for the time stamp clocks is rather lenient. A simple method for synchronizing distributed clocks with each other is to let each device periodically broadcast the reading of its own clock. The recipients then compare the received value with their own clock reading. If the received value is larger than the reading of the own clock, the own clock is set to the received value. Otherwise, the synchronization message is ignored. In this way, all clocks remain within a reasonable tolerance of each other, assuming that they are originally synchronized at some accuracy. To achieve this initial synchronization, a single PoB device adopts the clock value of the device it connects to when joining a scatternet. This value may be included in a REACHABLE message. In some cases, e.g. when two single devices connect or two scatternets merge, the clock values may differ significantly. In order to avoid unstable situations, a special procedure may be used in these cases if the clock values differ e.g. by more than a quarter of the clock register space. (That is, if e.g. an eight bit counter is used as a clock, the special procedure is used if the clocks differ by a value of 64 or more.) In this case a simple rule may be used for selecting the dominant counter value. When exchanging the REACHABLE messages, the counter value of the device with the larger BD_ADDR:es value may e.g. be ruled as dominant. In this case the overridden device broadcasts a synchronization message, with an indication of that the counters must be set to the sent value, on its original scatternet.

Another alternative for synchronization would be to use the time information provided by the GSM operator. This would be beneficial since the exchange of synchronization messages would be unnecessary. However, since all operators do not necessarily provide this service, this alternative cannot be relied upon. In addition, the time information of different operators may differ somewhat and thus cause incorrect operation. Other synchronization methods for distributed networks are discussed e.g. by Lamport et al [92].

7.8.2 Other Alternatives

Centralized Floor Control over GPRS

One alternative similar to the control-over-GPRS alternative of Section 7.3.2 is to use a centralized floor control server for floor control. The server maintains group lists on the basis of notifications from e.g. the creators of groups. Floor control could thus be handled in a similar manner as in PoC. The advantage of this approach would be that floor control and synchronization traffic could be avoided in the scatternet. The disadvantages are, however, the incurred costs similar to the ones discussed in Section 7.3.2.

7.9 Chapter Summary

This chapter presented a proposed method for enabling a PoC-like feature over Bluetooth. The target was to reach a solution, which would resemble PoC as closely as possible from the user's perspective. The proposal is divided into sections outlining group formation, network formation, communication, multiple group operation, and floor control. The network formation method is the most complex of these but it simplifies on the other hand the communication method significantly. It formes a loop-free network consisting of group members' devices only. The method for handling multiple group operation increases complexity considerably and should therefore be considered as optional. The floor control method itself is rather simple but it requires that the devices maintain time stamp clocks, which are mutually synchronized. The synchronization procedure increase complexity and traffic somewhat.

The throughput and delay requirements are also discussed to some extent. Since the delay requirements of PoC are rather lenient, the occurring delays should not cause problems in reasonably sized scatternets i.e. PoB groups. The throughput requirements are thus the critical performance issue. Results of prior work seem to suggest that sufficient throughput may be reached. However, further simulations and/or tests are needed in order to ensure this.

8 PoC and PoB Interoperation

In this chapter, a method combining PoB and PoC is briefly outlined. This combination is preferable, since some group members may be reachable via PoC even though they are not reachable via PoB. From the user point of view, this combination is thus desirable in many cases. Since the combining is rather straightforward, a very detailed discussion regarding this option is not presented.

8.1 Group Formation

Group formation is identical with Section 7.4 with the difference that SIP URI:s according to Section 4.2 are included for each member. The members are thus associated with both a BD_ADDRand a SIP URI and thus identifiable both in the PoB and PoC domain.

8.2 Communication

The PoB methods are performed as presented in Section 7. However, when the user presses the PTT button, the device also establishes a PoC ad-hoc group with the devices on the UNCONNECTED list according to the procedure described in Section 4.3.2. If the invited devices are in automatic answer mode, the talk burst can be sent both over PoB and PoC as soon as a Floor Granted message is received from the PoC server and the RTS timeout has occurred. The talk burst thus reaches all group members if they are reachable via either PoB or PoC. This procedure is performed separately by each device when they are about to send a talk burst. A number of ad-hoc groups are thus created, since each sender needs to have a separate ad-hoc group for communicating with the devices not reachable via PoB. (One common group should not be used, since it could eventually include all group members and the data would thus be conveyed to all members over PoC.) However, the devices outside the reach of PoB should create and use one common ad-hoc group including all group members. Talk bursts from these devices are thus conveyed over PoC without any actual PoB operation.

Examples clarifying the PoC group memberships and communication from the perspective of a connected and an unconnected device are presented in Figures 26 and 27 respectively.

In Figure 26, Jack's device is connected to the PoB scatternet and the communication with the other connected devices is handled as in Section 7. Rick's device, however, is not connected to the PoB scatternet and it can thus only be reached via PoC. When Jack presses the PTT button in order to speak to the group, "Ad-Hoc PoC Group 1" is created in order to be able to convey a talk burst to Rick. When Rick presses the PTT button in order to speak to the group, "Ad-Hoc PoC Group 2" is created in order for Rick to be able to convey a talk burst to all other group members. These ad-hoc groups are maintained throughout the entire PoB-PoC session. As indicated by the arrows in the figure, any outgoing talk bursts sent by Jack are sent to "Ad-Hoc PoC Group 1" and the PoB scatternet. Similarly, all incoming traffic from the connected group members is received from the PoB scatternet but incoming talk bursts from Rick are received from "Ad-Hoc PoC Group 2". If additional unconnected devices would exist, they would be members of both ad-hoc groups in the figure.

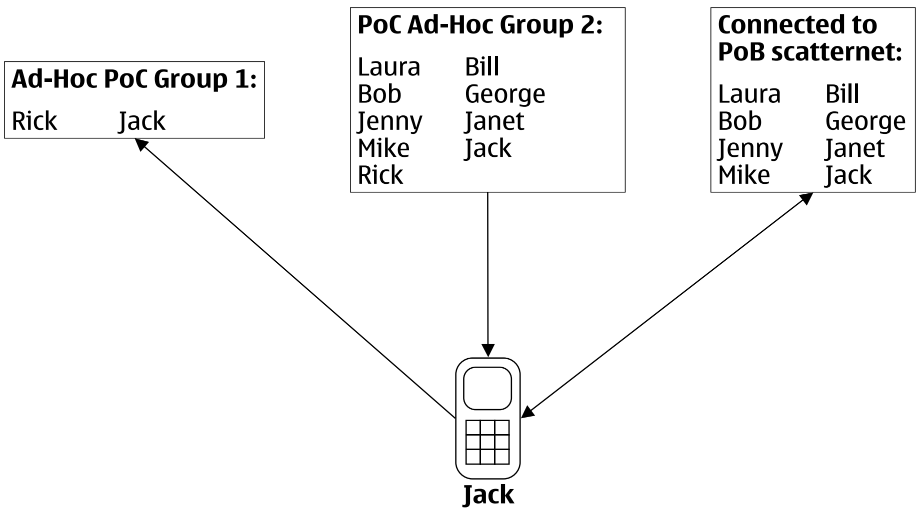

Figure 26: Example of Group Memberships and Communication; Perspective of a Device Connected to the PoB Scatternet

The perspective of an unconnected device, for example that of Rick's, is presented in Figure 27. As in Figure 26, Rick's device is a member of "Ad-Hoc PoC Group 1" and

"Ad-Hoc PoC Group 2". In addition, Rick's device is a member in two additional ad-hoc groups, since Laura and Rick have sent talk bursts to the group and these ad-hoc groups have thus been formed. In addition, Rick will be invited to additional ad-hoc groups, if e.g. Bob, George, Janet, etc. press the PTT button. As the arrows in the figure indicate, Rick's talk bursts are sent to "Ad-Hoc PoC Group 2", whereas talk bursts are received from other ad-hoc groups. However, if additional unconnected devices would exist, they would be members of "Ad-Hoc PoC Group 2". Talk bursts from these devices would be received via "Ad-Hoc PoC Group 2".

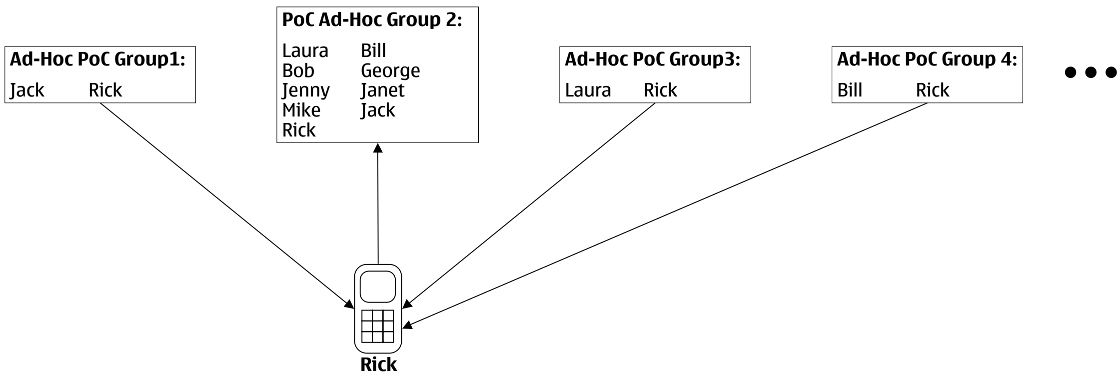

Figure 27: Example of Group Memberships and Communication; Perspective of a Device not Connected to the PoB Scatternet

In the figures, the group members of the ad-hoc PoC groups and the devices connected to the PoB scatternet are listed.

8.3 Floor Control

Floor control is somewhat tricky in PoB and PoC interoperation, and requires that the application can communicate with the PoC server as well as the PoB scatternet. When a device connected to the PoB scatternet wishes to send a talk burst, it sends an RTS over the scatternet according to Section 7.8 and requests the floor in the PoC ad-hoc groups according to PoC procedures. If any of these result in an indication that the floor is already occupied, the device refrains from transmitting. If however, Floor Granted-messages are received from the PoC server for all ad-hoc PoC groups and no conflicting RTS:s are received from the scatternet, the device informs the user of the right to speak and begins transmitting the talk burst.

In the example of Figure 26, Jack's device would thus broadcast an RTS on the PoB scatternet and send floor requests to both ad-hoc PoC groups. The talk burst itself, however, is not sent to "Ad-Hoc PoC Group 2". If the request is denied in either one of the PoC groups or by a received RTS from the PoB scatternet, Jack's device refrains from transmitting a talk burst.

The devices unconnected to the scatternet do not receive the RTS:es broadcasted in the PoB scatternet.[10] They should thus rely on the floor control services of the PoC server and should thus make sure that the floor is free in all PoB related ad-hoc PoC groups. In other words, if a PoC floor control message indicating that the floor is occupied is received in any of the groups the device belongs to, the device must refrain from transmitting talk bursts. (Assuming that the device only belongs to PoC groups related to the PoB group in question.)

In the example of Figure 27, Rick's device would send floor requests to all ad-hoc PoC groups. The talk burst itself, however, would be sent to "Ad-Hoc PoC Group 2" only. If the request is denied in any of the PoC groups, Rick's device refrains from transmitting a talk burst.

The number of used ad-hoc PoC groups may be up to $c+1$, where c is the number of devices connected to the PoB scatternet. The delays caused by sending floor requests to all of these and receiving the corresponding replies may become unacceptably large. This may be avoided if the method is altered in such a way that a floor request is only sent to the ad-hoc group to which a talk burst will be sent. This is possible since Floor Status messages will be received from the PoC server if any other device has requested the floor in any of the groups. If a device receives such a Floor Status message, it must refrain from sending a talk burst.

8.4 Chapter Summary

This chapter presents a proposed method for enabling a hybrid PoB-PoC service. The method is based on PoB as outlined in Section 7 but extended to allow PoC

[10] Notifications regarding the RTS:es could possibly be conveyed over GPRS to the unconnected devices. This alternative is not considered in detail, because convenient peer-to-peer data communication is currently not available in most of the currently existing networks.

communication with devices that are currently not reachable over Bluetooth. The method exploits the ad-hoc group capability of PoC and should provide smooth communication if the automatic answer mode is enabled in the recipients' devices. The number of PoC ad-hoc groups may, however, be rather large, and the handling of these may thus become somewhat complex.

9 Summary and Conclusions

9.1 Summary

As stated in the introduction, the main objective of this thesis is to find an answer to the question: "How can mobile phone users be provided with a free-of-charge PTT-feature with PoC-like user experience by means of Bluetooth technology?". The secondary objective is to give a comprehensive description of PoC as specified in the currently available OMA standard drafts and in other standards referred to therein. The fulfillment of the secondary objective provides a basis for undertaking the strive for the primary objective. In addition, such a unified description of PoC and its enablers has not, to the knowledge of the author, been available previously.

PoC and its enablers (GPRS, IMS, SIP, SDeP, and RTP-AMR) are described in chapters 2–4. Chapter 5 briefly describes Bluetooth in general and focuses then on its PAN profile. Chapter 6 presents use cases which are applicable to both PoC and PoB and the performance requirements of OMA PoC, which should preferably be met by PoB as well. Chapters 7 and 8 hold the actual substance of this thesis.

In chapter 7, a proposal for PoB is outlined. The proposal includes methods for group formation, network formation, communication and floor control. An optional method for handling multiple active groups is also included. The group formation method is based on exchanging messages including the BD_ADDR:es of the group members and distributing the complete list of these among group members. The network formation method is the most complicated of the presented methods but it is nevertheless based on rather simple principles. The principles it bases on are forming the scatternet among group members' devices only and disallowing loops. Loop avoidance is based on exchanging data regarding already connected devices and prohibiting connecting to these. Since the network formation makes sure only group members' devices are connected to the scatternet and that no loops exist, communication can be based on broadcasting data over the entire scatternet. Devices bridging two piconets act as repeaters in order to let the data reach every device in the network. The floor control method is simply based on broadcasting a time-stamped RTS message and beginning transmission unless RTS:s from other devices are received within a certain time frame. If two RTS:s traverse the network simultaneously, the time stamps are used to judge priority. In order to enable time stamping, a simple and rather inaccurate one-way synchronization method based on broadcasting is used. The inaccuracy is tolerable because users perceive closely adjacent presses of the PTT-button as concurrent. The optional method for handling multiple active groups and thus simultaneous session is based on maintaining several separate scatternets. This method is optional because it increases complexity and system requirements considerably.

Chapter 8 proposes an extension to the PoB feature of Chapter 7. The general principle is that PoC is used for communicating with the devices that are unreachable over Bluetooth. The devices unreachable in the PoB domain send data to all other group members via one ad-hoc PoC group. Incoming traffic to these unreachable devices, however, requires one ad-hoc PoC group per each sender connected to the PoB scatternet (otherwise all communications could eventually be conveyed over PoC). The group formation and floor control methods are adapted to suit the requirements of this context.

9.2 Results and Contribution

In chapters 2–6, publicly known subject matter is described. However, a corresponding tutorial of PoC and its enablers has not been publicly available until now. Even though many details are merely mentioned, the tutorial can be used as an introduction to PoC, which gives some insight on the underlying protocols and methods as well. It could be suitable for both students and professionals with basic knowledge of GSM and IP communication but limited knowledge of e.g. IMS, SIP, SDeP, and RTP.

In the technological and scientific sense, the contribution of this thesis is the proposed outlines for PoB (Chapter 7) and for the hybrid of PoB and PoC (Chapter 8). These outlines can be used as a baseline for creating such features by mobile phone manufacturers and software application producers. In addition to mobile phones, other devices, such as headsets or personal digital assistants, could also provide this feature. Similar text-based features and applications could also be developed for e.g. laptop and palmtop devices.

The network formation method of Section 7.5 may also be used for various other purposes besides PoB. For example, applications enabling multiplayer gaming, chat or shared folders or calendars may be based on communication over a scatternet formed according to the method. Furthermore, broadcasting according to Section 7.6 may also be used in such applications.

9.2.1 Evaluation of Results

The outlines of PoB and hybrid PoB-PoC proposed in this thesis provide a baseline for developing these notions further and possibly implementing them. The proposed outlines exploit existing conventions and protocols rather extensively and the new methods were designed with an aim for simplicity. Implementing PoB and developing PoB products on the basis of these outlines may therefore be relatively uncomplicated and cost effective. In addition, this may enable integrating PoB and PoC into one unitary application. Alternatively, separate PoB and PoC applications could possibly exploit the same capabilities of underlying software components.

On the other hand, the greatest flaw of the proposal of Chapter 7 is that there is no factual evidence that the throughput provided by Bluetooth in all or most real-life situations will be sufficient for satisfactory PoB operation. Therefore, the proposal does not provide a solid platform for further development but requires further research to verify its feasibility. Another risk factor is the fact that the OMA PoC standards are work in progress and the proposal is thus at least partly based on speculative assumptions. In addition, one issue that is not thoroughly considered is the consequences of operating other Bluetooth applications besides PoB. In order to enable satisfactory operation of both PoB and other simultaneous Bluetooth communications, this issue should be addressed.

9.2.2 Significance

Throughout modern history, technology has affected the behavior of people and vice versa. Some technology driven products and services have become commodities of everyday life whereas others have flopped in spite of great expectations. The typical examples of these are the Short Message Service (SMS) and Wireless Applications Protocol (WAP) respectively. Even though the apparent success of iDEN-based PTT in the Americas seems to indicate that PoC has the potential to soar globally, there are naturally no guarantees for this. Some of the risks which may make PoC unattractive in the eyes of the customer are interoperability, availability and pricing.

Currently, many parties are developing more or less competing solutions which may not be interoperable. OMA PoC could be the standard that provides interoperability across device manufacturer, operator and national borders. However, this requires that OMA PoC is adopted widely and acquires the acceptance of a majority of players.

Availability is naturally another factor that plays a major role in the adopting of PoC on the market. The service must obviously be available in order to succeed and operator acceptance is thus critical. The operators must be convinced that they can profit from providing PoC to their customers. Among operators, there is a clear interest towards PoC. However, even though PoC is envisioned as a totally new service that satisfies a different need than conventional phone calls, some operators may still see it as a service that cannibalizes this core service.

The third risk, pricing, is interrelated with operator acceptance. In a pursuit of increased revenues and in fear of cannibalization, operators may set the price of PoC at a level which renders PoC unattractive to the customers. If a strong demand for PoC develops, this risk will be abolished. However, it may be possible that a silent cartel chokes the build-up of popular demand before it has a possibility to emerge. It could be argued that this has been the case for WAP and the Multimedia Messaging Service (MMS).

In this context the role of PoB may develop to be significant. The above risks could potentially be avoided or alleviated by PoB.

Considering interoperability and availability, for example, these problems may be solved among a group of friends, if PoB is independent of the operator's services. Interoperability and availability could thus be reached if everyone in the group selects a phone that enables PoB. From the user point of view, the switching costs for reaching this interoperability within a group may in some cases be too high. On the other hand, if the majority of a group already has interoperable PoB phones and use them extensively, the minority might be very tempted to join this subgroup and consider the switching costs reasonable.

The success of PoB could also be beneficial from the PoC point of view. If PoB is successful and widely adopted on the markets, operators will probably be rather easy to convince that a need for PTT exists and that operators can profit from it. After all, PoB is will always be limited to a local area but the need for PTT may not. PoB could thus be a potential way to market PoC to both operators and end-users. In addition, the pricing risk is naturally eliminated in the PoB context.

PoB could also instigate new behavior models within local communication. Currently, telephony is usually associated with more or less long range communication because it does not make sense to pay for a phone call and go through the trouble of calling, if the called party is nearby. Since PoB is free of charge and the message can be conveyed without an actual calling procedure, the threshold for communicating is lower for PoB than for conventional phone calls or PoC. For example, consider the situation where two employees, Bill and Mike, are working at a fast food restaurant. Bill goes downstairs to the storage room to get some more French fries. Meanwhile, Mike, still at the counter, realizes that the chicken nugget box is almost empty as well. Mike will most probably not call Mike, at least not at his own expense, to tell him to bring some nuggets as well. However, he might consider telling it over PoB since it is easier and free. Many similar scenarios in which a need for communication exists but the required cost and/or effort seems too big in comparison to the benefit. Such situations occur rather frequently in family and social life, during hobbies and at work.

New ways to exploit PoB may also be developed. For example, PoB could potentially be used at different events and locations such as ice hockey games, museums or fairs for announcements or commentary e.g. by placing PoB "base stations" around the area and allowing PoB users to sign up for their services. The density of Bluetooth devices may, however, affect the functioning of such a service.

9.3 Evaluation of Method and Approach

In this thesis, the approach is top-down oriented in the sense that the target is to strive for PoB with a user experience resembling that of PoC as closely as possible. The codec and protocol stack of PoC was also adopted for the sake of simple implementation and in pursuit of modularity benefits as well as upholding the possibility to integrate PoB and PoC. These advantages are evident and may be crucial from the business aspect. The downside, however, is that this target setting and protocol stack adoption may render the solution infeasible, if the throughput in real-life situations is insufficient. Implementing PoB according to the outline without obtaining any throughput estimates by simulation or testing thus implies a considerable risk.

A more pragmatic approach would be to start out with simulating or testing the throughput and delays of both the PAN profile and scatternets consisting of Bluetooth class 1 devices. The tests should preferably be performed in different environments with various densities of Bluetooth and other devices operating in the ISM-band (Industrial, Scientific, and Medical). On the basis of the test and/or simulation results a bottom-up approach could be adopted. The speech codec and transport protocols could in this case be selected or designed in accordance with these findings. The methods and protocols could thus be optimized specifically for PoB, whereby significant advantages could possibly be gained. However, the downside of this approach is that the cost of developing a solution that is dedicated and optimized for Bluetooth may render PoB commercially infeasible. Integrating PoB and PoC or developing a hybrid PoB-PoC feature as in Chapter 8 would probably be more difficult as well.

9.4 Further Work

As indicated in the above discussion, one obvious issue requiring further research is the performance characteristics of scatternets and the PAN profile in various scenarios and with various settings. Tests and/or simulations are needed in order to acquire reasonably reliable estimates of these. The outcome of such tests or simulations will guide further work. If the requirements of the PoB proposal could be met, the following step would obviously be to develop PoB further on the basis of the proposed outline. On the other hand, if the results would suggest the contrary, PoB could possibly be redesigned to confine within the estimated throughput and delay characteristics. Alternatively, features with lower requirements such as text-based chat, turn-based gaming, and shared folders or calendars could be pursued.

The outline presented in this thesis merely set a framework for implementing PoB. Many details such as timeout values, intervals for periodical transmissions, and the time stamping counter characteristics remain to be specified and optimized. Other open issues are the Bluetooth baseband packet selection, specifying message formats, AMR mode selection, and the selection of a suitable AMR-frame per RTP-packet ratio.

As stated in the introduction, the security issues are delimited outside the scope of this thesis. This issue must nevertheless be considered, if PoB is to be provided to customers. The alternatives range from no security whatsoever, as in many traditional walkie-talkies, to providing a security level equivalent to PLMN conventions. Encryption may be based on the capabilities of Bluetooth or carried out on a higher level in the protocol stack.

Another topic that requires further work is the implementation environment. Symbian is one possible platform for developing the application. Even though Symbian does provide a versatile set of capabilities devoted for developing applications for mobile phones, it has its limitations. These limitations are not touched upon in this thesis and one task is thus to do this evaluation. In addition, other platforms and environments should be considered as well.

An additional issue related to implementation is the possibility that other applications exploiting Bluetooth may be run concurrently with the PoB application. Preferably, the PoB application should not interfere with other applications and vice versa. However, if other applications besides PoB create and use Bluetooth connections, a method for administering the connections and the media access is needed.

Appendix: Throughput Requirement Calculations

This appendix explains how the throughput requirements presented in Figure 21 on page 57 have been calculated.

The calculations correspond to the HCI level (see Figure 17 on page 41). The throughput requirement values thus represent the throughput requirement imposed on the Bluetooth Controller. The overhead caused by the L2CAP is thus included.

As mentioned in Sections 4.9 and 4.10, the protocols and voice codec to be used in OMA PoC is not specified at the time of writing. It seems reasonable, however, to assume that usage of an IP/UDP/RTP based protocol stack (as presented in Figure 12 on page 34) is probable. A straightforward way to implement this protocol stack on top of Bluetooth is to use the PAN Profile and the BNEP protocol, and this approach is therefore assumed in these calculations.

The overhead caused by the L2CAP/BNEP/IP/UDP/RTP protocol stack is presented below. (See also Figure 13 on page 35.)

RTP	96 bits	[51]
UDP	64 bits	[48, 49]
IP	160 bits	[47]
BNEP (uncompressed header)	120 bits	[73]
L2CAP	32 bits	[59]
TOTAL	**472 bits**	

Usage of AMR in RTP involves the following variables, which influence the throughput requirement: interface format (IF1 or IF2), AMR Mode (See Figure 14 on page 36), and the number of AMR frames per RTP packet. In the calculations below, usage of AMR IF2 is assumed, because its capacity requirement is higher than that of IF1. The requirement of IF2 can thus be used as an upper bound for approximating the requirement of IF1. If a more accurate requirement is needed for IF1, it may be calculated in a corresponding manner as that of IF2. The frame sizes of different AMR modes, when one frame is included per RTP packet, are presented in Table 5 on page 38). Since only one occurrence of the CMR field is included per RTP packet, the RTP payload size S_P is

$$S_P = S_F + (x - 1)(S_F - 8) \tag{1}$$

bits per RTP packet when x AMR IF2 frames are included per RTP packet. The value of S_F corresponds to the AMR IF2 frame size depending on the AMR mode. The valid values of S_F are presented in the IF2-column of Table 5.

Since the frame rate of AMR is 50Hz, the number of RTP packets sent per second is

$$f = \frac{50}{x} \tag{2}$$

and the required throughput capacity C is thus

$$C = f \cdot (472 + S_P) = \frac{50 \cdot (472 + S_F + (x - 1)(S_F - 8))}{x} \tag{3}$$

bits per second.

References

[1] "Push to Talk Report, Market analysis and forecasts 2003–2009", London, 2004, Visiongain Ltd., 107p.

[2] Push to Talk Over Cellular (PoC) Charter, OMA Push to talk over Cellular Working Group, 2003, 3p.

[3] Kopranen, T., Nokia internal document, April 2004, information source: Strategy Analytics.

[4] Kargl, F. et al, "Bluetooth-based Ad-Hoc Networks for Voice Transmission", Proceedings of Hawaii International Conference on System Sciences 36, Hawaii, USA, January 2003, 9p.

[5] Perkins, C. et al, "Ad-Hoc On-Demand Distance Vector Routing", Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999, pp.90–100.

[6] Johnson, D. et al, "Dynamic Source Routing in Ad Hoc Wireless Networks", In: Imielinksi , T. et al, eds., "Mobile Computing: The Kluwer International Series in Engineering and Computer Science (vol. 35)", Norwood 1996, Kluwer Academic Publishers, pp.153–181.

[7] Linsky, J et al, "Bluetooth 1.2: Version 1.2 of Bluetooth has now been launched, but what does it bring?", Wireless Business and Technology, Volume 4, Issue 1, 2004, pp.48–50 (available at: www.siliconwave.com/pdf/articles/0104_article_wireless_business_technology.pdf).

[8] Prasad, N. R., "GSM Evolution Towards Third Generation UMTS/IMT2000" Proc. IEEE International Conference on Personal Wireless Communication, 1999, pp.50–54.

[9] Hakaste, M. et al, "GSM/EDGE Standards Evolution (up to Rel'4)" In:Halonen, T. et al eds., "GSM, GPRS and EDGE Performance, Evolution Towards 3G/UMTS, Second Edition", Chichester 2003, John Wiley et Sons, pp.3–56.

[10] 3GPP TS 29.060 V5.7.0 (2003–09), 3rd Generation Partnership Project; Technical Specification Group Core Network; General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface (Release 5), 94p.

[11] 3GPP TS 23.002 V5.12.0 (2003–09), 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Network architecture (Release 5), 52p.

[12] Bettstetter, C, et al "GSM Phase 2+ General Packet Radio Service: Architecture, Protocols, and Air Interface", IEEE Communications Surveys, Volume 2, Issue 3, 1999, pp.2–14.

[13] 3GPP TS 03.60 V7.9.0 (2002–09), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Stage 2 (Release 1998), 110p.

[14] Stuckmann, P., "The GSM Evolution, Mobile Packet Data Services", Chichester 2003, John Wiley et Sons, 241p.

[15] 3GPP TS 23.207 V5.8.0 (2003–06), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; End-to-end Quality of Service (QoS) concept and architecture (Release 5), 47p.

[16] Poikselkä, M. et al, "The IMS, IP Multimedia Concepts and Services in the Mobile Domain", Chichester, 2004, John Wiley & Sons Ltd, 419p.

[17] Wei, Z. et al, "Policy-Based QoS Architecture in the IP Multimedia Subsystem for UMTS", IEEE Network, Volume 17, Issue 3, 2003, pp.51–57.

[18] Asprino, P. et al, "Implementation of a conference service in 3G architecture", Proceedings of the 10th International Conference on Software, Telecommunications & Computer Networks – Split–Venice–Ancona–Dubrovnik, October 8–11, 2002, pp.23–27, (available at: www.coritel.it, referred on 8 March 2004).

[19] 3GPP TS 23.228 V6.3.0 (2003–09), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 6), 158p.

[20] Wong, D et al, "Supporting Real-Time IP Multimedia Services in UMTS", IEEE Communications Magazine, Volume 41, Issue 11, 2003, pp.148–155.

[21] Rosenberg, J. et al, "SIP: Session Initiation Protocol", RFC 3261, 2002, 269p.

[22] Johnston, A. B., "Understanding the Session Initiation Protocol", second edition, Norwood, 2004, Artech House, 283p.

[23] Handley, M. et al, "SDP: Session Description Protocol", RFC 2327, 1998, 42p.

[24] Rosenberg, J. et al, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, 2002, 25p.

[25] Schulzrinne, H. et al, "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 3551, 2003, 44p.

[26] IANA, "RTP Parameters", 8 April 2004 (available at: www.iana.org/assignments/rtp-parameters referred on 15 April 2004).

[27] "Mobile Industry Leaders Complete Joint Specification for 3GPP IMS Based Push to Talk over Cellular (PoC) Technology", Nokia press release, September 10 2003 (available at: http://press.nokia.com/PR/200309/916980_5.html referred on 28 January 2004).

[28] User Requirements V1.1.0 (2003–08); Push-to-talk over Cellular (PoC); User Requirements; PoC Release 1.0, Ericsson, Motorola, Siemens, Nokia, 27p. (available at: www.ericsson.com/multiservicenetworks/distr/PoC_specifications.ZIP referred on 24 May 2004).

[29] Architecture V1.1.0 (2003–08); Push-to-talk over Cellular (PoC); Architecture; PoC Release 1.0, Ericsson, Motorola, Siemens, Nokia, 23p. (available at: www.ericsson.com/multiservicenetworks/distr/PoC_specifications.ZIP referred on 24 May 2004).

[30] Transport Protocols V1.1.0 (2003–08); Push-to-Talk over Cellular (PoC) User Plane; Transport Protocols; PoC Release 1.0; Ericsson, Motorola, Siemens, Nokia, 37p. (available at: www.ericsson.com/multiservicenetworks/distr/PoC_specifications.ZIP referred on 24 May 2004).

[31] (E)GPRS / UMTS Specification V1.1.0 (2003-08); Push-to-Talk over Cellular (PoC) User Plane; (E)GPRS / UMTS Specification; PoC Release 1.0; Ericsson, Motorola, Nokia, Siemens, 16p. (available at: www.ericsson.com/multiservicenetworks/distr/PoC_specifications.ZIP referred on 24 May 2004).

[32] List Management and Do-not-Disturb V1.1.3 (2003–08; Push to Talk over Cellular (PoC); List Management and Do-not-Disturb; PoC Release 1.0, Ericsson, Motorola, Siemens, Nokia 53p. (available at: www.ericsson.com/multiservicenetworks/distr/PoC_specifications.ZIP referred on 24 May 2004).

[33] Signaling Flows V 1.1.3 (2003–08); Push-To-Talk over Cellular (PoC); Signaling Flows; PoC Release 1.0; Ericsson, Motorola, Nokia, Siemens 115p. (available at: www.ericsson.com/multiservicenetworks/distr/PoC_specifications.ZIP referred on 24 May 2004).

[34] "Open Mobile Alliance, Overview", OMA document, 2003, 5p. (available at: www.openmobilealliance.org/docs/OMAShortPaper2003.pdf referred on 25 March 2004).

[35] "Open Mobile Alliance, Overview", OMA document, May 2004, 5p. (available at: www.openmobilealliance.org/docs/OMAShortPaper_May2004v.1.doc, referred on 30 June 2004).

[36] OMA Push to Talk over Cellular Requirements, Version 1.0 – 3 June 2004, OMA-RD_PoC-V1_0-20040603-C, 75p.

[37] OMA Push to talk over Cellular (PoC) – Architecture, Draft Version 1.0 – 7 April 2004, OMA-AD_PoC-V1_0-20040325-D, 102p.

[38] OMA Calendar view (public), March 11, 2004, 3p. (available at: www.openmobilealliance.org/tech/workprogram.htm referred on 18 March 2004).

[39] OMA Work Programme and Release Handling Processes, Version 1.1.1, 2004, 17p. (available at: http://member.openmobilealliance.org/ftp/public_documents/rel/Permanent_documents/ referred on 27 March 2004).

[40] 3GPP2 X.S0013-000-0 All-IP Core Network Multimedia: Overview, version 1.0, December 2003, 18p.

[41] 3GPP TR 23.979 V0.4.0 (2004–05); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP enablers for OMA PoC Services; Stage 2 (Release 6), 15p.

[42] Vähä-Sipilä, A., "URLs for Telephone Calls", IETF RFC 2806, 2000, 21p.

[43] "Push to Talk over Cellular Real-time always-on voice service", Espoo, 2003, Nokia Networks Ltd., 12p.(available at: www.nokia.com/poc/PoC_WP_A4.pdf referred on 24 May 2004).

[44] Chen, B., "Input Contribution: Proposal of the protocol for the It reference point", London, March 2004, Open Mobile Alliance, 2p. (availabel at: http://member.openmobilealliance.org/ftp/Public_documents/POC/2004/ referred on 24 May 2004).

[45] Synnergren, P., "Input Contribution: User Datagram Protocol", London, April 2004, Open Mobile Alliance, 1p. (availabel at: http://member.openmobilealliance.org/ftp/Public_documents/POC/2004/ referred on 24 May 2004).

[46] Synnergren, P., "Input Contribution: Real-time Transport Protocol", London, April 2004, Open Mobile Alliance, 1p. (availabel at: http://member.openmobilealliance.org/ftp/Public_documents/POC/2004/ referred on 24 May 2004).

[47] Postel, J., "Internet Protocol" RFC 0760, 1981, 45p.

[48] Postel, J., "User Datagram Protocol", RFC 0768, 1980, 3p.

[49] Larzon, L-A. et al, "The UDP-Lite Protocol", Internet Draft, IETF Network Working Group, 2003, 13p. (available at: www.ietf.org/internet-drafts/draft-ietf-tsvwg-udp-lite-02.txt referred on 26 May 2004).

[50] Larzon, L-A. et al, "UDP Lite for Real Time Multimedia Applications" (Article), Bristol, 1999, Hewlett Packard Laboratories, 6p. (available at: www.hpl.hp.com/techreports/1999/HPL-IRI-1999-001.pdf referred on 26 May 2004).

[51] Schulzrinne, H., "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, 2003, 104p.

[52] Perkins, C., "RTP: Audio and Video for the Internet", Boston, 2003, Addison Wesley (Pearson Education), 414p.

[53] 3GPP TS 26.071 V5.0.0 (2002–06); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mandatory speech CODEC speech processing functions; AMR speech CODEC; General description (Release 5), 12p.

[54] Adaptive Multi Rate Codec: Cost effective route to network capacity expansion, Espoo, 2003, Nokia Mobile Phones Ltd., 2p.(available at: www.nokia.com/nokia/0,,55213,00.html referred on 1 June 2004).

[55] 3GPP TS 26.101 V5.0.0 (2002–06); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mandatory speech codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec frame structure; (Release 5).

[56] Sjoberg, L. et al, "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 3267, 2002, 49p.

[57] Morrow, R., "Bluetooth Operation and Use", New York, 2002, McGraw-Hill, 567p.

[58] Specification of the Bluetooth System, Wireless connections made easy, Core, Volume 1, Version 1.1, Bluetooth Special Interest Group, 2001, 1084p.

[59] Specification of the Bluetooth System, Wireless connections made easy, Volumes 0–3, Version 1.2, 2003, 1200p.

[60] IEEE 802 Standard for Local and Metropolitan Area Networks: Overview and Architecture, 2002, 47p.

[61] Bluetooth Personal Area Networking Profile, Version 1.0, Bluetooth Special Interest Group, 2003, 65p.

[62] IANA, "Special-Use IPv4 Addresses", IETF RFC 3330, 2002, 7p.

[63] Cheshire, S. et al, "Dynamic Configuration of Link-Local IPv4 Addresses", Internet Draft, IETF ZEROCONF Working Group, 2004, 31p. (available at: www.ietf.org/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-15.txt referred on 30 June 2004).

[64] Srisuresh, P. et al, "IP Network Address Translator (NAT) Terminology and Considerations", IETF RFC 2663, 1999, 30p.

[65] Holdrege, M. et al, "Protocol Complications with the IP Network Address Translator", IETF RFC 3027, 2001, 20p.

[66] Kuthan, J., "Internet Telephony Traversal across Decomposed Firewalls and NATs", 2nd IP Telephony Workshop in New York, 2001, 8p. (available at: www.fokus.fraunhofer.de/research/cc/mobis/publ/doc/2001/iptel2001-kuthan-fcp-01.pdf referred on 14 April 2004).

[67] Thernelius, F., "SIP, NAT, and Firewalls", Master's Thesis, 2000, 69p. (available at:
www.cs.columbia.edu/sip/drafts/Ther0005_SIP.pdf referred on 15 April 2004).

[68] Plummer, D., "An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", IETF RFC 826, 1982.

[69] Peterson, L. et al, "Computer Networks: A Systems Approach", second edition, San Francisco, 2000, Morgan Kaufmann Publishers, 748p.

[70] ANSI/IEEE Standard 802.1D, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks-Common specifications-Part 3: Media Access Control (MAC) Bridges, 1998, 375p.

[71] "The Ethernet – A Local Area Network", Version 1.0 Digital Equipment Corporation, Intel Corporation, Xerox Corporation, September 1980.

[72] IEEE Standard 802.3, Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks-Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2000, 1552p.

[73] Bluetooth Network Encapsulation Protocol (BNEP) Specification, Version 1.0, Bluetooth Special Interest Group, 2003, 55p.

[74] Mišić, V. et al, "Polling and Bridge Scheduling Algorithms in Bluetooth", Technical Report TR 03/04, Department of Computer Science, University of Manitoba, September 2003, 17p. (available at: www.cs.umanitoba.ca/~vmisic/pubs/tr0304.pdf, referred on 29 June 2004.

[75] Zürbes, S., "Considerations on link and system throughput of Bluetooth networks", The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, Volume 2, September 2000, pp.1315–1319.

[76] Zanella, A. et al, "On the impact of fading and inter-piconet interference on Bluetooth performance", The 5th International Symposium on Wireless Personal Multimedia Communications, Volume 1 , October 2002, pp.218–222.

[77] Chen, L.-J. et al, "Audio streaming over Bluetooth: an adaptive ARQ timeout approach" Proceedings of 24th International Conference on Distributed Computing Systems Workshops, 23–24 Mar. 2004, pp.196–201.

[78] Mišić, J. et al, "Talk and let talk: performance of Bluetooth piconets with synchronous traffic", Ad Hoc Networks, in Press, (available online 24 December 2003), 27p.

[79] Kapoor, R. et al, "Bluetooth: carrying voice over ACL links", Proceedings of 4th International Workshop on Mobile and Wireless Communications Network, Los Angeles, CA, 2002, pp.379383.

[80] Steenstrup, M. "Cluster-Based Networks" In:Perkins, C. ed., "Ad Hoc Networking", Boston 2000, Addison-Wesley, pp.75–138.

[81] Dijkstra, E., "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, Volume 1, 1959, pp.260271.

[82] Hall, E. et al, "RF Rendez-Blue: reducing power and inquiry costs in Bluetooth-enabled mobile systems", proceedings of the Eleventh International Conference on Computer Communications and Networks, 14–16 October 2002, pp.640–645.

[83] Busboom, A. et al, "Unambiguous Device Identification and Fast Connection Setup in Bluetooth", European Wireless 2002 Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, Florence, Italy, February 25–28, 2002, 5p.

[84] Daptardar, A., "Meshes and Cubes: Distributed Scaternet Formations for Bluetooth Personal Area Networks", Master's thesis, May 2004, 112p. (available at: www.eecs.wsu.edu/∼mmedidi/research/blueware/wsuthesis.pdf, referred on 22 June 2004).

[85] Persson, K., "Bluetooth scatternet formation: criteria, models and classification", First IEEE Consumer Communications and Networking Conference, 5–8 January 2004, pp.59–64.

[86] Li, X.-Y. et al, "Partial Delaunay triangulation and degree limited localized Bluetooth scatternet formation" IEEE Transactions on Parallel and Distributed Systems, Volume 15, Issue 4, April 2004, pp.350–361.

[87] Mišić, J. et al, "Bridges of Bluetooth County: Topologies, Scheduling, and Performance", IEEE Journal on Selected Areas in Communications, Volume 21, Issue 2, February 2003, pp.240–258.

[88] Mogul, J., "Broadcasting Internet Datagrams", RFC 0919, 1984, 8p.

[89] Deering, S., "Host Extensions for IP Multicasting" RFC 1112, 1989, 17p.

[90] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, 1997, 24p.

[91] Cain, B. et al, "Internet Group Management Protocol, Version 3", RFC 3376, 2002, 53p.

[92] Lamport, L. et al, "Byzantine Clock Synchronization", Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing, August 1984, pp.68–74.