HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Electrical and Communications Engineering
Networking Laboratory

Mika Lehtinen

Session Border Controller and IP Multimedia Standards

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Helsinki, Finland November 11, 2005.

Supervisor	Professor Raimo Kantola
Instructor	M.Sc. Harri Välimäki

Helsinki University of Technology	Abstract of the Master’s Thesis
Author:	Mika Heikki Lehtinen
Name of the Thesis:	Session Border Controller and IP Multimedia Standards
Date:	13.11.2005
Number of pages:	109
Department:	Department of Electrical and Communications Engineering
Professorship:	Telecommunications Technology
Supervisor:	Professor Raimo Kantola
Instructor:	M.Sc. Harri Välimäki

The aim of this thesis is to study a network element often referred to as Session Border Controller, and analyze the relationship of the functions it performs to IP multimedia standards.

The first part of this thesis describes the concepts of IP multimedia technology. Short introduction to some of the common IP multimedia protocols is presented. Also the key concepts relevant to IP multimedia implementations such as QoS, NAT traversal and communication security are introduced. For practical reasons this study focuses on IP telephony and Voice over IP, as they are currently the dominant IP multimedia applications. SIP has a central role in current IP multimedia and SBC implementations. This study focuses on SIP, but other protocols are included, where applicable.

Different SBC deployment scenarios are described along with the functions a session border controller performs. It is also described, why each function is performed or what is achieved by performing it.

The second part of the thesis analyses and compares the SBC functions and standards. The purpose is to find out and identify standard and non-standard behaviour. Also as a secondary goal, the role of the SBC functions in architectures of different standard families is analysed.

The final chapter of the thesis contains results of the comparison and conclusion of the work performed. One of the key results of the thesis is a description of what SBC functions can be modelled by functionality specified in the standards.

Keywords:	Internet, IP telephony, Voice over IP, IP multimedia, standardization, session border controller, SIP

Teknillinen korkeakoulu — Diplomityön tiivistelmä

Tekijä:	Mika Heikki Lehtinen
Työn nimi:	Sovellusreitin ja IP multimediastandardit
Päivämäärä:	13.11.2005
Sivumäärä:	109
Osasto:	Sähkö- ja tietoliikennetekniikan osasto
Professuuri:	Tietoverkkotekniikka
Työn valvoja:	Prof. Raimo Kantola
Työn ohjaaja:	DI Harri Välimäki

Tämän työn tarkoituksena on tutkia sovellusreitittimeksi (Session Border Controller, SBC) kutsutun verkkoelementin suhdetta IP multimediastandardeihin. Tutkimus suoritetaan vertailemalla sovellusreitittimen toiminnallisuutta standardeihin.

Työn ensimmäinen osa esittelee IP multimediatekniikan peruskäsitteet. Esittely kattaa yleisimmät merkinantoprotokollat joilla IP multimediapalveluja toteutetaan. Lisäksi aihepiirin keskeiset käsitteet, kuten QoS palvelunlaatu, osoitteenmuunnoksen läpäisy sekä tietoturva esitellään. IP puhe eri muodoissaan on yleisimmin käytetty IP multimediapalvelu ja siksi sillä on keskeinen rooli tässä työssä. SIP puolestaan on keskeinen protokolla sekä IP multimediatoteutuksissa että sovellusreitittimien toiminnassa ja tässä työssä keskitytään tutkimaan sovellusreitittimen toiminnallisuutta SIP:n kautta. Muut protokollat huomioidaan silloin kun se on tarpeen toiminnallisuuden käsittelemiseksi. Työssä tarkastellaan erilaisia SBC:n käyttötapoja sen toiminnallisuutta. Toiminnallisuutta kuvattaessa selitetään miksi kyseinen toiminto suoritetaan, tai mitä sillä saavutetaan.

Työn toisessa osassa verrataan SBC:n toimintoja standardeihin. Tämä on työn päätavoite. Vertailun tarkoituksena selvittää, mikä osa toiminnallisuudesta on standardien mukaista ja mikä standardeissa määrittelemätöntä. Työn toinen tavoite on selvittää SBC:n rooli eri IP multimedia-arkkitehtuureissa.

Työn viimeinen kappale sisältää vertailun tulokset ja johtopäätökset tulosten perusteella. Eräs keskeisimmistä tuloksista on selvittää, mitkä sovellusreitittimen toiminnoista ovat standardien mukaisia ja mitkä eivät.

Avainsanat: Internet, IP puhe, IP multimedia, standardointi, sovellusreititin, SIP

Acknowledgements

This thesis was written at TeliaSonera Finland, Product & Service Development, Network Solutions department in Helsinki, Finland.

First of all, I would like to thank Professor Raimo Kantola for his guidance and views, and M.Sc. Harri Välimäki for advice and support during the writing of this thesis.

The subject of this thesis was chosen mostly out of personal interest, and I would like to thank my managers at TeliaSonera Finland for making this possible.

Most importantly, I would like to thank my friends and family, for their love and support during my studies and the writing of this thesis.

December 13, 2005

Helsinki, Finland

Mika Lehtinen

List of symbols and abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ACL	Access Control List
ARF	Access Relay Function
ALG	Application Level Gateway
A-MGF	Access Media Gateway Function
AMR	Adaptive Multi-Rate
AS	Application Server
ASN.1	Abstract Syntax Notation One
B2BUA	Back to Back User Agent
BGCF	Breakout Gateway Control Function
BGF	Border Gateway Function
CAC	Call Admission Control
CALEA	Communications Assistance for Law Enforcement Act
CCITT	International Telegraph and Telephone Consultative Committee
CDR	Call Detail Record
COPS	Common Open Policy Service
CSCF	Call Session Control Function
DNS	Domain Name System
DoS	Denial of Service

DSCP	DiffServ Code Point
DSL	Digital Subscriber Line
DSP	Digital Signal Processing
E911	FCC Enhanced 911 Service
ETSI	European Telecommunications Standards Institute
EU	European Union
FCC	Federal Communications Commission
FW	Firewall
GSM	Global System for Mobile Communication
GW	Gateway
H.323	Packet-Based Multimedia Communications Systems Recommendation H.323
HTTP	Hyper Text Transport Protocol
IBCF	Interconnection Border Control Function
ICE	Interactive Connectivity Establishment
I-CSCF	Interrogating Call Session Control Function
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
ITSP	Internet Telephony Service Provider

ITU	International Telecommunication Union
IWF	Interworking Function
L2TF	Layer 2 Termination Function
LAN	Local Area Network
LSP	Label Switched Path
MCU	Multi-Point Control Unit
MEGACO	Media Gateway Control Protocol
MG	Media Gateway
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MIDCOM	Middlebox Communication
MPEG	Motion Picture Experts Group
MPLS	Multi-Protocol Label Switching
MRFC	Multimedia Resource Function Controller
MRFP	Media Resource Function Processor
NAPT	Network Address and Port Translation
NASS	Network Attachment Subsystem
NAT	Network Address Translation
NGN	Next Generation Network
PBX	Private Branch Exchange
PCM	Pulse Code Modulation
P-CSCF	Proxy Call Session Control Function
PLMN	Public Land Mobile Network
PSTN	Public Switched Telephony Network

QoS	Quality of Service
RACS	Resource and Admission Control Subsystem
RCEF	Resource Control Enforcement Function
RFC	Request for Comments
RSIP	Realm Specific Internet Protocol
RSVP	Resource Reservation Protocol
RTCP	Real Time Control Protocol
RTP	Real Time Transport Protocol
SBC	Session Border Controller
S-CSCF	Serving Call Session Control Function
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SGF	Signalling Gateway Function
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMTP	Simple Mail Transport Protocol
SNMP	Simple Network Management Protocol
SRTP	Secure Real-time Transport Protocol
STUN	Simple Traversal of User Datagram Protocol Through Network Address Translators
TDM	Time Division Multiplexing
THIG	Topology Hiding Inter-network Gateway
TISPAN	Telecommunications and Internet converged Services and Protocols for Advanced Networking

TLS	Transport Layer Security
T-MGF	Trunking Media Gateway Function
ToS	Type of Service
TSAP	Transport Layer Service Access Point
TURN	Traversal Using Relay NAT
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UE	User Equipment
UPSF	User Profile Server Function
URI	Uniform Resource Indicator
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network

Table of Contents

Index of Figures

Index of Tables

1 Introduction

1.1 Background

Packet based networks, such as the Internet and other networks utilizing the Internet Protocol (IP), are increasingly being used for interactive conversational communication applications. These applications include for example telephony and multimedia conferencing. Traditionally these kinds of conversational applications have been implemented using circuit switched technology in the Public Switched Telephone Network (PSTN) or the Public Land Mobile Network (PLMN). The Internet on the other hand has been typically used to deliver things like email messages, or web pages, which are first viewed and read by the end user, potentially followed by a response, like writing a reply email message or following a web link.

The move from circuit switched networks towards IP based networks is related to convergence of network technologies. The term network convergence is used to define the developments towards a common network technology among a set of different networks such as the fixed and the mobile telephone networks and the Internet and private IP networks.

There are several drivers for convergence. One driver is improving cost efficiency by eliminating some of the parallel networks having wide coverage and serving the same geographical area, e.g. the PSTN, PLMN and the Internet. Another common driver is a vision of the ability to create new applications by combining the various means of communication originally found in separate non-interworking networks into one application combining things like voice, video, instant messaging, email, etc. into a seamless user experience.

The technical solutions and architectures of the various different networks have been initially developed to meet the requirements of the applications used in those networks and also to mach the business models or philosophies of the organizations operating

them. In addition to purely technical issues such as fundamental differences between circuit and packet switching or how access control to the network is done, there are many non-technical differences. These differences originate form things such as varying business models, government regulation including legal requirements for communications privacy and emergency services. Standards specifying the different networks are often developed by different standardization organizations with different goals.

The Session Border Controller (SBC) is a network element that addresses some of the issues that have surfaced when building real-time IP multimedia services in converged networks. The concept of a SBC has originated from the IP multimedia industry to meet the requirements of operators, service providers and enterprises. It has been developed by individual vendors outside the standards making processes of standardization organizations, such as the IETF, ITU-T, ETSI or IEEE. However the SBC interfaces to several standards and this thesis discusses and analyzes the relationship between the SBC concept and the relevant IP multimedia standards.

1.2 Goals of the Thesis

This thesis focuses on a network element often referred to as Session Border Controller and the relationship it has to IP multimedia standards and standard organizations. The main goal of the thesis is to compare the functionality of session border controllers to the standards and find out: What SBC functionality is standard behaviour and what is non-standard? We also describe what functions are performed by SBCs and why those functions are performed? The chosen method is comparing the functions of SBCs to the functions specified in the standards.

A secondary goal of this thesis is to compare the views or opinions that different standardization organizations specifying IP multimedia communication infrastructures have towards the kind of functionality a session border controller performs.

1.3 Scope of the Thesis

This thesis focuses on the functions of the session border controller and the protocols directly related to SBC and conversational real-time IP multimedia. Multimedia signalling protocol focus is on SIP. Other protocols relevant to SBC functionality are included in the scope, but are handled with less detail.

Protocols that are not multimedia signalling protocols, but are however directly related to the functions of SBCs are included in the scope. These protocols include e.g. the protocols used to carry media streams in IP multimedia communications.

1.4 Structure of the Thesis

This thesis is structured into two main parts. The first part, chapters 2 through 4, contains background information and description of the essential concepts related to real-time IP multimedia, standards and the session border controller, the network element under study.

Chapter 2 describes the concept of IP multimedia and the time dependence related to interactive and conversational communication. The key standard organizations that have contributed to the development of IP multimedia standards are introduced.

Chapter 3 presents the most common signalling protocols and entities found in IP multimedia communication infrastructures.

Chapter 4 introduces the session border controller.

The second part ranging from Chapter 5 to Chapter 6 contains the analysis and comparison of the session border controller functionality with the standards.

Chapter 5 focuses on the main goal of the thesis: Comparison of the functionality of session border controllers with the standards and distinguishing between standard and non-standard behaviour. In addition to the main goal, the secondary goal of analyzing the relationship of SBC and the architectures of different standards organizations is addressed in Chapter 5.

Chapter 6 contains the conclusions of the thesis.

2 IP Multimedia Overview

This chapter describes the concept of real-time IP multimedia and the time dependence related to interactive and conversational communication. Concept of quality of service is described along with implementation in IP networks. In addition, the key standard organizations that have contributed to the development of IP multimedia standards are introduced.

IP multimedia means the exchange of any digitalized information between the communication parties with a variety of different communication modes, such as interactive communication, streaming, and sharing. Examples include video call, sharing a whiteboard or a Web page during a phone call, or sharing a still image during a phone call [Nok04].

Real time means that the content has some kind of time dependence. When real time multimedia is used by people for interactive communication, the time dependence is related to the user experience. A perfect user experience from the real-time point of view can be thought to be achieved by people communicating in same space, such as a room.

The IP in real-time IP multimedia means that the multimedia traffic is transported by networks utilizing the Internet Protocol. IP is a packet based protocol defined by RFC 791 and later specifications. The networks utilizing IP are called IP networks. The Internet is the world’s largest IP network, but most of the corporate enterprise and community private networks (intranets) are also IP based. IP multimedia applications are used in both the Internet and intranets and are gaining popularity in mobile wireless networks.

2.1 Quality of Service

By default IP networks offer best effort service when delivering packets between users. This means that the network makes no attempt to actively differentiate its service response between the individual packets or traffic streams generated by concurrent users of the network. As a result of this, individual IP packets experience varying response times when travelling across the network from source to destination.

Interactive communication for example in a room is virtually instantaneous and there is no distortion of sound and vision. Deviations from the natural user experience result in degradation of perceived quality of service (QoS) [Har03]. Characteristics of a best effort transport network with variable service response affect perceived QoS indirectly.

Those characteristics that can be measured without reference to user perception of quality but that will, affect user perception of quality are referred to as intrinsic QoS [Har03]. Intrinsic QoS is characterized by:

- Latency, or delay–The time it takes a packet to get across the network to its destination
- Jitter–The variability in packet latency
- Dropped packet rate or packet loss–The frequency or percentage with which packets do not get to their destination in time to be used

The ITU-T standard E.800 defines QoS as *The collective effect of service performance which determine the degree of satisfaction of a user of a service.* This broad concept includes aspects such as the quality of customer support functions, etc. E.800 Serviceability or technical QoS is one dimension of the “collective effect” and in its turn includes concepts like accessibility, retainability and integrity of service. In this thesis however, the term QoS is used in even more narrow manner. It is used to reflect the impact of intrinsic QoS parameters (latency, jitter, packet loss) on perceived QoS of communication medium such as voice and video.

2.2 Controlling QoS in IP Networks

The response time in networks working properly (under no error condition) can be anything starting from less than one millisecond to several seconds or even more. Packets may also be completely discarded while in transit. This causes degradation of perceived QoS. Many mechanisms have been developed to overcome the varying default best-effort service response of IP networks to better match the requirements of interactive communication. Describing them thoroughly is out of the scope of this thesis and only some are introduced. For interactive applications needing better than best-effort service, integrated services (IntServ) [Bra94] and differentiated services (DiffServ) [Bla98] are often used.

2.2.1 Integrated Services

Integrated services or IntServ is an architecture, which specifies the ways to guarantee QoS in IP networks. IntServ can be used to allow real-time traffic to be delivered to the receiver in a better than best effort fashion.

IntServ specifies a fine-grained QoS system. The idea of IntServ is that every router in the path implements it, and every application that requires guarantees makes a reservation. Resource Reservation Protocol (RSVP) is used as the underlying mechanism to signal reservations across the network.

IntServ requires that some flow state information on the reservations to be kept in every network element such as routers in the network. This limits the scalability of IntServ.

2.2.2 Differentiated Services

Differentiated services or DiffServ attempts to provide better than best effort QoS on IP networks. DiffServ deals with aggregated flows of data rather than individual flows and single reservations, like IntServ. A single reservation may be made for all of the packets of an aggregated flow. When packets enter a DiffServ network they are first classified by the sender or a router at the edge of the network. The classification is marked in the DiffServ Code Point (DSCP) in the IP header.

Within the DiffServ network, routers queue and forward packets class/priority indicated by the DSCP header field. As a result of this, no flow state needs to be kept in routers.

2.3 IP Multimedia Standards and Standard Organisations

IP multimedia and related standards are developed and published by several independent standardization organisations. The following sections introduce the organizations that are relevant in the scope of this thesis.

2.3.1 ITU-T

ITU-T is the Telecommunication Standardization Sector of the International Telecommunication Union (ITU). It was established on 1 March 1993 replacing the former International Telegraph and Telephone Consultative Committee (CCITT). The ITU is an international organization within the United Nations.

An ITU-T Recommendation H.323 [ITU96], was the first standard published, describing a system that can be considered an IP multimedia communication system. H.323 describes terminals, equipment, and services for multimedia communication over Local Area Networks (LAN). H.323 terminals and equipment may carry real-time voice, data, and video, or any combination, including video telephony [ITU96]. The second release of H.323 recommendation H.323 v2 [ITU97] expanded generalized the scope of the standard from local area networks to interconnected LANs. This generalization allowed H.323 to be used in any IP network.

The ITU-T continues to develop H.323, but has also activities in the Next Generation Network (NGN) area that are relevant to this thesis. The ITU-T Recommendation Y.2001 [ITU04] states that an NGN is *"A packet-based network able to provide telecommunications services and able to make use of multiple broadband, QoS-enabled transport technologies in which service-related functions are independent from underlying transport-related technologies."*

2.3.2 IETF

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual [IET05].

The IETF develops and publishes standards (RFC) that the Internet and other IP networks are based on. In addition to network standards many key specifications, such as the Session Initiation Protocol (SIP) is specified by the IETF. SIP is a major standard for IP multimedia communication systems.

2.3.3 3GPP

The 3rd Generation Partnership Project (3GPP) is a collaboration agreement that was established in December 1998. The collaboration agreement brings together a number of telecommunications standards bodies.

The original scope of 3GPP was to produce globally applicable Technical Specifications and Technical Reports for a 3rd Generation Mobile System based on evolved GSM core networks and the radio access technologies that they support. The 3GPP originally decided to prepare specifications on a yearly basis. The first specification was Release 99 [Poi04].

The key concept of 3GPP standardisation in the scope of this thesis is the IP Multimedia Subsystems (IMS). The first version of IMS was included in 3GPP Release 4, frozen and officially completed in March 2001 [Poi04]. The current version is Release 6, March 2005 [3GP05a].

2.3.4 ETSI

The European Telecommunications Standards Institute (ETSI) is an independent, non-profit organization, whose mission is to produce telecommunications standards for today and for the future [ETS05a].

A particularly relevant area of ETSI standardisation is the ETSI TISPAN NGN functional architecture. This architecture complies with the ITU-T general reference model for next generation networks [ETS05]. The TISPAN NGN functional architecture contains several service layer components, of which the core IP Multimedia Subsystem (IMS) is in the scope of this thesis, as it implements real-time IP multimedia services. The IP multimedia services of ETSI TISPAN NGN are based on Release 7 of 3GPP IMS but have been further extended by TISPAN NGN.

3 IP Multimedia Signalling Protocols

IP multimedia signalling protocols have been developed for different purposes and by different standardization organizations. This chapter introduces common IP multimedia signalling protocols and entities that are relevant to the SBC in the scope of this thesis.

3.1 Session Initiation Protocol

The Session Initiation Protocol, SIP is a signalling protocol developed by the IETF for use in IP networks. The first RFC was published in 1999 [Han99] and the current one in 2002 [Ros02]. SIP is an application layer protocol that can be used to establish, modify and tear down communication sessions between users. Examples of these sessions include multimedia conferencing sessions and telephony. SIP can be used with different transport protocols, like UDP, TCP and SCTP and has been defined for use with both IPv4 and IPv6 [Ros02].

SIP uses UTF-8 text based request and response messages. It borrows elements of two widely used Internet protocols: Hyper Text Transport Protocol (HTTP) and Simple Mail Transport Protocol (SMTP). From HTTP, SIP uses a client-server design and the use of URLs and URIs. The text-encoding scheme and header style of messages is borrowed from SMTP. For example SIP reuses SMTP headers such as To, From, Date, and Subject. [Joh04].

The core SIP is simple and the text based messages are easily readable. These properties make it easy to start developing applications using SIP. The SIP architecture is designed to be scalable and new functionality can be added by extensions [Joh04].

3.1.1 SIP Entities

The SIP specification defines several entities: User Agents, Redirect Servers, Proxy Servers, Registrars and Location Servers. Their role in the SIP architecture is discussed in the following sections.

User Agents

SIP uses a client-server design as a basis of operation. A SIP end device is called a SIP user agent. A user agent can be for example a fixed telephone device, mobile phone, personal workstation or a network element such as a media gateway. SIP user agents contain both a client and a server part: User Agent Client (UAC) and User Agent Server (UAS). The UAC initiates requests while the UAS generates responses. During a session, a user agent will usually operate as both a UAC and a UAS. This approach is different from other client-server Internet protocols such as HTTP. The Web browser is always an HTTP client, and the Web server is always an HTTP server. During a SIP session, an end point will act in both roles.

A back-to-back user agent (B2BUA) is a type of SIP device that receives a SIP request, then reformulates the request and sends it out as a new request. Responses to the request are also reformulated and sent back in the opposite direction. A B2BUA maintains dialog state and must participate in all requests sent on the dialogs it has established. Since it is a concatenation of a UAC and UAS, no explicit definitions are needed for its behaviour.

The Figure 1 describes a request and response sequence between two user agents through a B2BUA. Arrows from left to right represent requests. Arrows from right to left are responses. The dashed arrows represent stateful processing by the B2BUA in order to determine how to reformulate the request (2) and the response (5).

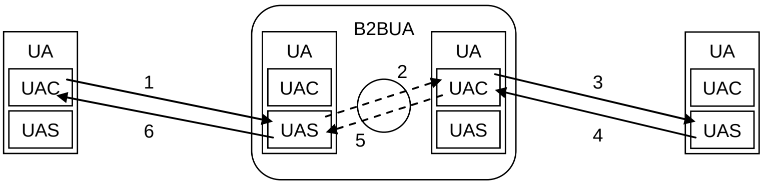

Figure 1 Request-response flow via B2BUA

SIP gateways are entities that contain a user agent but instead of a human, interface to another protocol, like ISUP or H.323. A gateway terminates signalling and may terminate media if required. Media termination is required for example at a media gateway between an IP network and circuit switched PSTN/PLMN. A gateway between SIP and H.323 does not have to terminate media. SIP and H.323 endpoints in an IP network can exchange media directly and only the signalling needs to be processed by a gateway.

Proxy Servers

A SIP proxy server or proxy receives requests from a user agent and other proxies. It acts on behalf of a user agent by responding to requests or forwarding them. A proxy often has access to a database or location service, which it uses in order to determine the forwarding destination of a received request. A proxy is not required to understand the full content of a message in order to pass it on and should not change the order of header fields or in general modify or delete header fields.

A proxy differs from a user agent in the following ways:

1. A proxy server does not send requests independently, but only responds to requests from a user agent. (A CANCEL request is an exception. A proxy sends A CANCEL without first receiving it from a UA)
2. A proxy server does not handle media streams
3. A proxy server does not parse message bodies, but only looks at message headers

A proxy can be either stateless or stateful. A stateless proxy server processes each SIP request or response based solely on the header field information in that particular message. No dialog state information is stored in the proxy server after processing and forwarding a message. As no state information is kept, a stateless proxy never retransmits messages.

A stateful proxy keeps track of requests and responses it has received and uses that information in processing future requests and responses. A stateful proxy can have timers for retransmitting requests that have not been responded to after a retransmit timer has expired. Also, a stateful proxy can require user agent authentication, thus it can be used to implement authentication.

A transaction stateful proxy server keeps transaction state from the beginning of a transaction until it has completed. For example, a transaction stateful proxy would start keeping state after receiving an INVITE request and stop after receiving a 200 OK. This kind of operation in between stateful and stateless allows a proxy to perform useful search services while minimizing the amount of state storage required.

Yet another kind of proxy server is a forking proxy. It can forward a request to multiple destinations. On use for a forking proxy is creating services where several user agents are offered a session simultaneously in order to reach the user [Joh04].

Redirect Servers

A redirect server receives SIP requests using a UAS element but instead of forwarding the request to the direction of the destination like a proxy, it notifies the initiator of the original request on the location of the destination. This is done using a 3xx redirection class response. Like a proxy, the redirect can utilize databases or other location services to be able to determine where the other party of communication can be reached at [Joh04].

Registrars

A registration server, or a registrar, accepts only SIP REGISTER requests. All other received requests are responded with a 501 Not Implemented response. The contact information obtained by a registrar during register transaction is then made available to other SIP servers within the same administrative domain, such as proxies and redirect servers [Joh04]. Registrars are often co-located with proxies in order to make the publication of contact information straightforward [Cam02].

Location Servers

A location server provides location services used by a SIP redirect or proxy servers to obtain information about a possible location of the called party. It contains a list of bindings of address-of-record keys to zero or more contact addresses. The bindings can be created and removed in many ways. SIP specification [Ros02] defines a REGISTER method that updates the bindings.

3.2 SIP Addresses

SIP uses e-mail-like names for addressing users and devices. The addressing scheme belongs to a family of Internet addresses called URIs. SIP URIs can handle telephone numbers, transport parameters, and a number of other items. The key point is that a SIP URI is a name that is resolved to an IP address by using SIP proxy server and DNS.

SIP has two broad categories of URIs: ones that correspond to a user, and ones that correspond to a device or end point. The user URI is known as an address of record (AOR) and a device URI as a contact. A request sent to an AOR will require database lookups and service operations to complete. A request sent to a contact typically does not require database lookups. An AOR URI is usually used in To and From headers to reach a person. A device URI used in a Contact header field and is associated with a particular device through which the user can be reached at that time.

SIP supports a number of URI schemes including sip, sips, tel, pres, and im. Their meaning is respectively SIP, secure SIP using TLS, telephone, presence, and instant message. The most commonly used are sip and sips. [Joh04].

A simple SIP URI could be for example:

sip:mika.lehtinen@iki.fi

sip:mlehtine@hut.fi

sip:1234@192.168.0.10

3.2.1 SIP Messages

All SIP messages are either requests or responses. They are formatted according to RFC 2822 and contain: request line / status-line, message headers, empty line, and message-body. Figure 2 is an example of a SIP message.

Request line	`INVITE sip:user@provider.com SIP/2.0`
Message header	`Via: SIP/2.0/UDP myhost.hut.fi:5060`
	`From: sip:mika.lehtinen@hut.fi;tag=94919237389882988`
	`To: sip:user@provider.com`
	`Call-ID: 456456456@myhost.com`
	`CSeq: 1 INVITE`
	`Contact: sip:mlehtine@myhost.hut.fi:5060`
	`Content-Type: application/sdp`
	`Content-Length: 123`
Empty line	
	`v=0`
Message body	`o=0 0 IN IP4 192.168.0.10`
	`...`

Figure 2 SIP request message

The first row is called the starting line and it distinguishes between request and response message types. Message headers are followed by the starting line. An empty line ends the sequence of message headers and may be followed by an optional message body [Ros02].

SIP Requests

SIP request messages request another SIP entity to perform a task described by the message. The RFC 3261 defines six methods INVITE, REGISTER, BYE, ACK, CANCEL and OPTIONS. Other methods have been defined later [Joh04].

The INVITE method is used to invite another user agent to a media session. An invite often contains a session description (SDP) [Han98] in the message body, which describes the properties of the media session. Parameters of an existing media session may be modified using a re-INVITE message.

The REGISTER method is used by a user agent to inform the SIP network servers of the current location of the user. The current contact URI of a user agent is published using REGISTER. A registration can have a finite lifetime and the registration must be refreshed periodically if continuous validity of the registration is desired.

The BYE method ends an established session. A session is considered to be established, if an INVITE has been responded to with a positive response, or an ACK has been transmitted. Only user agents that are part of an established session may send BYE messages. Proxy servers or other third parties may not send BYE messages.

The ACK method is used to acknowledge final responses to INVITE requests. Message body in an ACK response may contain SDP to describe properties of the media session. The ACK method may not be used to modify media session parameters of INVITE messages. Re-INVITE must be used for that purpose.

CANCEL is used to cancel pending transactions. It ends a transaction started by INVITE. A CANCEL message may be sent by a user agent or a proxy server in some cases. A user agent uses CANCEL to tear down a call attempt it has initiated. A forking proxy may use CANCEL to cancel calls progressing with other destinations once a session with one user agent has been established.

SIP Responses

A SIP response is a message sent by a UAS or a SIP server as a response to a request from a UAC. A SIP response contains status information related to a request. A response may contain additional header fields with information needed by the UAC or, it may be a simple acknowledgment to prevent retransmissions of the request by the UAC.

SIP responses are formatted like requests, but the first line contains status information instead of a method including SIP version, status code and description, like “SIP/2.0 180 Ringing”. The response classes in Table 1 are defined in SIP [Ros02].

1xx	Informational	Indicates status of call prior to completion. Is the first informational or provisional response.
2xx	Success	Request has succeeded. If for an INVITE, ACK should be sent; otherwise, stop retransmissions of request.
3xx	Redirection	Server has returned possible locations. The client should retry the request at another server.
4xx	Client error	The request has failed due to an error by the client. The client may retry the request if reformulated according to response.
5xx	Server failure	The request has failed due to an error by the server. The request may be retried at another server.
6xx	Global failure	The request has failed. The request should not be tried again at this or other servers.

Table 1 SIP response codes

Below are some examples of responses of above classes [Joh04]:

100 Trying This response is only a hop-by-hop request. It is never forwarded and may not contain a message body. This response can be generated by either a proxy server or a user agent to indicate that some kind of action is being taken to process the call.

180 Ringing This response is used to indicate that the INVITE has been received by the user agent, and that alerting is taking place.

200 OK This response has two uses in SIP. When used to accept a session invitation, it will contain a message body containing the media properties of the called party. When used in response to other requests, it indicates successful completion or receipt of the request.

302 Moved Temporarily This redirection response contains a URI that is currently valid but that is not permanent. The Contact header contains a temporarily valid destination.

407 Proxy Authentication Required This response sent by a proxy indicates that the UAC must first authenticate itself with the proxy before the request can be processed. The response should contain information about the type of credentials required by the proxy in a Proxy-Authenticate header field.

500 Server Internal Error This server error class response indicates that the server has experienced some kind of error that is preventing it from processing the request.

600 Busy Everywhere This response is used to indicate that the user agent cannot accept the call and that the request should not be tried elsewhere either i.e. the user does not wish to receive any calls at the moment.

3.2.2 SIP Header Fields

SIP header fields may be categorized as request and response, request only, response only, and message body header fields. This distinction is not based on the SIP protocol itself, but on the part of SIP messages they appear in [Joh04]. Numerous header fields have been defined in IETF specifications, but only the typical ones and the ones that are important in the scope of this study, are introduced.

A tag is a cryptographically random number with at least 32 bits of randomness. A Tag is not a header field itself, but is added To and From headers to uniquely identify a dialog. To header in the initial INVITE will not contain a tag, but a caller must include a tag in the From header. A tag returned in a 200 OK response is then used as a dialog identifier in all future requests for a particular Call-ID.

To This field is a required header field in every SIP message used to indicate the recipient of the request. Any requests generated by a user agent contain this header field with the addition of a tag. Any response generated by a proxy must have a tag added to the To header field. The To header field is never used for routing.

From This field is a request and response required header field and indicates the originator of the request. A From header field may contain a tag, to identify a particular call. If there is both a URI parameter and a tag, then the URI including any parameters must be enclosed in <>.

Subject This request and response header field is optional and used to indicate the subject of the session. The contents of this header field can e.g. be displayed during alerting.

Call-ID This request and response header field is mandatory in all SIP requests and responses. It is used to uniquely identify a call between two user agents. Call-ID is unique across calls, except in the for registration requests. All registrations for a user agent should use the same Call-ID.

Via This is a required request and response header field and is used to record the SIP route taken by a request. It is used to route a response back to the originator of a request. A user agent generating a request records its own address in a Via header field. The order of via entries in the message is significant as it is used to route responses.

A proxy forwarding the request adds a Via header field containing its own address to the top of the list. A proxy or user agent generating a response to a request copies all the Via header fields from the request into the response, then sends the response to the address specified in the top Via header field. A proxy receiving a response checks the top Via header field and checks that it matches its own address. If it does not, the response has been misrouted and is discarded. The proxy then removes the top Via header field, and forwards the response to the address specified in the next Via header field.

Contact This request and response header field is used to carry a URI that identifies the resource requested or the request originator. In a request it identifies the request originator and in a response, the requested resource.

A received Contact header field can be cached and used to contact the other user agent directly, bypassing proxies. However, if a Record-Route header fields in an earlier request or default proxy routing configuration may override direct connection.

Record-Route This request and response header field is used to force routing through a proxy for all subsequent requests in a session between two user agents. Normally, a Contact header field allows user agents to send messages directly bypassing the proxy chain used in the initial request. A proxy inserting its address into a Record-Route header field overrides this and forces this proxy to be included.

Cseq This request and response header field is required in every request and indicates the command sequence of requests. The CSeq header field contains a decimal number that increases for each request. Usually, it increases by 1 for each request, with the exception of CANCEL and ACK requests, which use the CSeq number of the INVITE to refer to the correct request. The CSeq count is used to identify out-of-sequence requests, new requests and retransmissions.

Max-Forwards This mandatory request header field is used to indicate the maximum number of hops that a SIP request may travel. The value of the header field is decremented by each proxy that forwards the request. A proxy receiving the header field with a value of zero discards the message sending a 483 Too Many Hops response to the originator.

Content-Type This message body header field is used to specify the Internet media type [Pos94] in the message body. Media types have the form type/sub-type. If this header field is not present, application/sdp is assumed.

Content-Length This message body header field indicates the number of octets in the message body. A Content-Length: 0 indicates no message body.

3.2.3 SIP Message Body

A SIP message body contains a description of the session to be established. Both requests and responses may contain message bodies, but all do not. The message body in a SIP message usually is a session description, but it can consist of any object. SIP proxies do not need to examine the message body, thus the content is transparent to them. As a result, session descriptions are transmitted end to end between user agents [Cam02].

3.3 Session Description Protocol

Session Description Protocol (SDP) is a protocol defined by RFC 2327. It is more of a description of syntax than a protocol in that it does not provide a full-range media negotiation capability. The original purpose of SDP was to describe multicast sessions set up over the Internet’s multicast backbone, the MBONE. Today SDP is used with SIP and MGCP.

An SDP session description consists of a number of lines of text of the form <type>=<value>. <type> is always exactly one character and is case-significant. White space is not permitted either side of the `=' sign. In general <value> is either a number of fields delimited by a single space character or a free format string. Figure 3 shows a session description.

SDP contains the following information about the media session [Joh04]: IP Address or host name, port number used by UDP or TCP for transport, media type (audio, video, etc.) and media encoding scheme (PCM A-Law, MPEG II video, etc)

In addition, SDP contains information about the following: Subject of the session, start and stop times and contact information about the session.

```
v=0
o=395231 691550547 691550577 IN IP4 192.168.0.37
s=X-Lite
c=IN IP4 192.168.0.37
t=0 0
m=audio 8000 RTP/AVP 3 8 0 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

```
Session description
        v=  (protocol version)
        o=  (owner/creator and session identifier)
        s=  (session name)
        i=* (session information)

        u=* (URI of description)
        e=* (email address)
        p=* (phone number)
        c=* (connection information – not
            required if included in all media)
        b=* (bandwidth information)
        One or more time descriptions (see below)
        z=* (time zone adjustments)
        k=* (encryption key)
        a=* (zero or more session attribute lines)
        Zero or more media descriptions (see below)

Time description
        t=  (time the session is active)
        r=* (zero or more repeat times)

Media description
        m=  (media name and transport address)
        i=* (media title)
        c=* (connection information - optional if
             included at session-level)
        b=* (bandwidth information)
        k=* (encryption key)
        a=* (zero or more media attribute lines)

*) optional
```

Figure 3 SDP example with descriptions

3.4 H.323

The first version H.323 standard was approved in 1996 by ITU-T. It specifies protocols and architectural components of an IP multimedia system. The H.323 is an umbrella for a family of ITU-T recommendations

The H.323 network is divided into administrative domains. An administrative domain may be defined as a collection of H.323 functional entities that one administrative entity manages. One administrative domain can be composed of the entities of an enterprise, and another of the entities administered by a service provider [Kum01].

3.4.1 H.323 Entities

The core elements of a H.323 system are terminals, gateways, gatekeepers, multi-point control units. Additional entities are specified by H.323 recommendations, but they are not introduced here.

Terminals

A terminal is an entity that terminates signalling and media at an end-user's location. A terminal can be for example a PC running a H.323 conferencing application, or a H.323 telephone. The terminal is sometimes called a terminal client.

Gateways

A gateway is an entity connecting a H.323 network to a circuit switched network. It performs signalling protocol and media format conversion between networks. Gateways are used for example to establish connectivity between H.323 and ISDN networks.

A Multi-Point Control Unit is an entity used for multiparty (>2) conferences. A MCU mixes audio and switches video from all participants and then provides it to the end-user's terminals.

Gatekeepers

A Gatekeeper is an entity that provides services to end users and routes messages to their destinations. The services that are provided by a gatekeeper include authentication, authorisation and accounting (AAA) and address resolution to routable IP addresses. A gatekeeper controls a zone. A zone has one gatekeeper, one or more terminals and may have GWs and MCUs.

H.323 terminals, gateways and multi-point control units all terminate both signalling and media. They are referred to as endpoints.

3.4.2 H.323 Addresses

A H.323 endpoint has one or more network addresses. In the case of IP, they are IP addresses.

An endpoint may have one or many alias addresses. An alias address can be a URL (e.g. h323://mlehtine@hut.fi), E.164 number or a character string. A gatekeeper is needed to resolve aliases and without gatekeeper host names must be used [Bei98].

For each network address, each H.323 entity may have several TSAP (Transport Layer Service Access Point) identifiers. These TSAP Identifiers allow multiplexing of several channels sharing the same network address [ITU00].

3.4.3 H.323 Signalling

Most of the control messages in H.323 are encoded in the Abstract Syntax Notation One (ASN.1) scheme using the Packet Encoding Rules (PER).

In H.323 call signalling is carried in channels. The RAS channel is used to carry messages used in the Gatekeeper discovery and endpoint registration processes, which associate an endpoint's alias address with its Call Signalling Channel Transport Address. The RAS channel is an unreliable channel. The Call Signalling channel must be used to carry H.225.0 call control messages.

In networks that do not contain a Gatekeeper call signalling messages are passed directly between the endpoints using the Call Signalling Transport Addresses. In networks that contain a Gatekeeper the initial admission message exchange takes place between the calling endpoint and the Gatekeeper using the Gatekeeper's RAS Channel Transport Address.

With a gatekeeper two call signalling routing methods are available. The first method is Gatekeeper Routed Call Signalling. In this method signalling messages are routed through the gatekeeper. The second method is Direct Endpoint Call Signalling. In this method the call signalling messages are passed directly between the endpoints. [ITU00].

A modified subset of Q.931 ISDN D-channel signalling is used for call setup between terminals. H.245 is used for control signalling or media negotiation and capability exchange between terminals. T.120 is used for multipoint graphic communications. [Joh04]

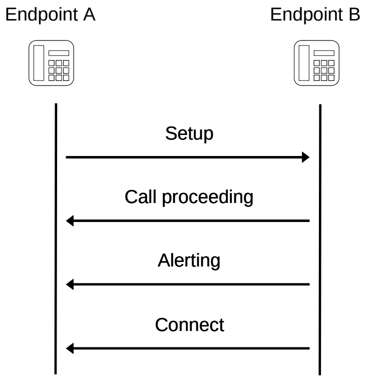

Figure 4 Basic H.323 call setup

Figure 4 shows a simplified example. Practical scenarios often include a gatekeeper with gatekeeper registration. Also terminal capability negotiation, master-slave determination opening of logical channels contribute to a significant number of signalling message exchange. Call set-up time can increases relative to the amount of signalling. To speed up H.323 call setup, a FastStart method has been defined in H323 v2.

3.4.4 Media in H.323

H.323 uses the IETF RTP and RTCP, for the media transport. The H.323 signalling can perform codec negotiations to find a suitable codec between endpoints. H.323 audio codecs are specified in the ITU G.7xx series and video codecs in the H.26x series.

3.4.5 H.323 Versions

Numerous versions of H.323 have been released. The current one is version 5. Full backward compatibility is required by H.323 specifications, but forward compatibility may be an issue between versions and might require protocol conversion.

3.5 MGCP

MGCP stands for Media Gateway Control Protocol and specifies a method for controlling media gateways from external call control elements called media gateway controllers or Call Agents.

A media gateway typically provides conversion between the signals carried in a circuit switched network and packet networks. A media gateway only processes media and not signalling [And03].

3.5.1 MGCP Architecture

MGCP assumes a call control architecture where the call control "intelligence" is outside the gateways and handled by external call control elements known as Call Agents. MGCP assumes a connection model where the basic constructs are endpoints and connections. Endpoints are sources or sinks of data and can be physical or virtual. A physical endpoint is for example an interface on a gateway to a PSTN switch. An example of a virtual endpoint is an audio source in a media server. Examples of connection are the transmission of a RTP media stream over IP, or transmission of a TDM signal in the backplane of a TDM switch.

MGCP is defined in an informational, non-standard IETF document, RFC 3435, which obsoletes an earlier definition in RFC 2705. MGCP is intended to be an internal protocol used within a distributed system that can appear to the outside world as a single VoIP gateway. The approved IETF protocol for the same purpose, although not as widely implemented as MGCP, is Megaco, defined in RFC 3015.

3.6 MEGACO / H.248

Megaco or H.248 is a Media Gateway Control Protocol [Cue00] designed for control of elements in a physically decomposed multimedia gateway enabling separation of call control from media conversion. Megaco is a result of joint efforts of the IETF and the ITU-T Study Group 16. Therefore, the IETF defined Megaco is the same as ITU-T Recommendation H.248.

The Megaco is a protocol used between elements of a physically decomposed multimedia gateway. There are no functional differences from a system view between a decomposed gateway and a monolithic gateway [Cue00].

Megaco addresses the relationship between the Media Gateway (MG), which converts circuit-switched voice or other media to packet-based traffic, and the Media Gateway Controller (MGC), sometimes called a call agent or softswitch. The MGC implements call processing and service logic while the MG implements media processing. Figure 5 shows a decomposed gateway.

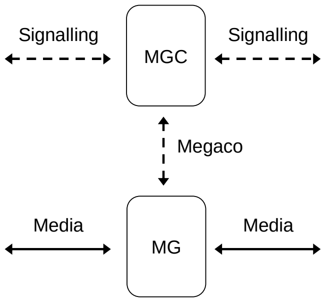

Figure 5 A decomposed gateway

Megaco Connection Model

The connection model for the protocol describes the logical entities, within the Media Gateway that can be controlled by the Media Gateway Controller. The main abstractions used in the connection model are Terminations and Contexts.

A Termination sources and/or sinks one or more streams. In a multimedia conference, a Termination can be multimedia. This means, it can source or sink multiple media streams. A Context is an association between two or more Terminations. The Context describes the topology (who hears/sees whom) and the media mixing and/or switching parameters if more than two Terminations are involved in the association. For example two endpoints sending and receiving media can be terminations and a media session between them is described by a context. Contexts are created and released by the MG under command of the MGC. A context is created by adding the first termination, and it is released by removing (subtracting) the last termination [Cue00].

The Megaco protocol provides commands for manipulating the logical entities of the connection model: contexts and terminations [Cue00]. The commands and descriptions are listed in Table 2.

Add	The Add command adds a termination to a context. The Add command on the first Termination in a Context is used to create a Context.
Modify	The Modify command modifies the properties, events and signals of a termination.

Subtract	The Subtract command disconnects a Termination from its Context and returns statistics on the Termination's participation in the Context. The Subtract command on the last Termination in a Context deletes the Context.
Move	The Move command atomically moves a Termination to another context.
AuditValue	The AuditValue command returns the current state of properties, events, signals and statistics of Terminations.
AuditCapabilities	The AuditCapabilities command returns all the possible values for Termination properties, events and signals allowed by the Media Gateway.
Notify	The Notify command allows the Media Gateway to inform the Media Gateway Controller of the occurrence of events in the Media Gateway.
ServiceChange	The ServiceChange Command allows the Media Gateway to notify the Media Gateway Controller that a Termination or group of Terminations is about to be taken out of service or has just been returned to service.

Table 2 Megaco commands

3.7 Real-Time Transport Protocol

Real-Time Transport Protocol (RTP) is defined by RFC 3550. It is designed to carry real-time traffic across IP networks. RTP does not provide any quality of service over the IP network. This means, that RTP packets are handled the same as other packets in an IP network. However, some of the impairments, such as packet loss, jitter, out of sequence packets and asymmetric routing, can be detected [Joh04].

RTP is an application layer protocol that uses UDP for transport. RTP includes a bit-oriented header similar to UDP and IP. RTP was designed to be very general. Most of the headers are only loosely defined in the standard and the details are left to profile documents. The RTP specification defines a RTP companion protocol called the RTP Control Protocol (RTCP). It allows participants in an RTP session to exchange quality reports, statistics and some basic identity information.

RTP is the dominant protocol to transport real-time multimedia in IP networks. All signalling protocols in this thesis utilize RTP.

4 The Session Border Controller

The role of this chapter is to introduce the session border controller along with the functions it performs. Deployment scenarios are presented in order to show how and why the functionality is used in IP multimedia networks.

The session border controller (SBC) is a multi function network element or a network device used as one building block of real-time IP multimedia service platforms. It is a relatively new concept when compared to other IP network elements, like routers or firewalls. The concept of session border control was present on the Fall Voice on the Net 2002 conference agenda, but the first early SBC like devices emerged already in 2001.

The session border controller is not defined by any single standardisation organisation and SBCs are sometimes called session controllers, border controllers, IP-IP gateways or application routers.

IP Telephony and VoIP are typical IP multimedia applications and SBCs are often used in platforms implementing these services. SBCs are, however, not limited to voice communication solutions and can handle other interactive real-time IP multimedia, such as video conferencing and instant messaging (IM).

Carriers and service providers are the typical users of SBCs as they help to manage operating services across the boundaries of administrative domains of IP networks. Enterprises may use SBCs to manage IP multimedia traffic across internal network boundaries, or more commonly at the edge of the enterprise network and the Internet service provider network.

IP multimedia services can be implemented using several standard protocols. Common protocols are H.323, SIP and MGCP. Typical SBCs can interface with multiple protocols simultaneously and all vendors appear to support SIP.

SBCs operate on the session layer (layer 5) and can process both signalling messages and media streams in the context of communication sessions. They provide layer 5 control and management in the network, which is beyond the scope of routers and firewalls.

SBCs have emerged to fill in some of the gaps left open by protocol standards that make it difficult for service providers and enterprises to implement IP multimedia services. The development of SBCs is mostly industry driven, but since late 2004, there has been emerging activity in the IETF on the SBC.

4.1 What Problems Does a SBC Address?

Session border controllers are usually located at network borders. The border can be the boundary between administrative domains, or a boundary of domains defined by technology related criteria.

Common administrative domain borders are borders between two different network operators, between a network operator and a service provider, between service provider and an enterprise, or service provider and residential. Sometimes enterprises peer directly with each other, forming an administrative border.

Common technology domain borders are borders between networks using different addressing such as public and private IP addresses. Networks that use different versions of the IP protocol (IPv4 / IPv6) form a technology border as well. Services using different IP multimedia protocols such as SIP and H.323 form borders. Likewise services using different variants of the same IP multimedia standards such as IETF SIP and 3GPP SIP form technology borders.

The above two domains, administrative and technology, can also overlap simultaneously: For example two peering VoIP network operators A and B might be using different signalling protocols in their network. One might use SIP and the other H.323, thus forming a non-interoperable technology border. On the other hand, the network operators A and B are separate companies belonging to different administrative domains. This results in security and commercial requirements for interconnecting A and B.

Also a more future scenario can be seen in a fixed-mobile convergence case between 3G mobile services and Internet services. 3GPP IMS IP multimedia services are heavily based on IPv6. The Internet is mostly an IPv4 network. In order for a mobile service provider C and an internet service provider D to establish service interoperation, both administrative and IP protocol version related issues in SIP service must be solved.

A traditional solution to solve the inter-domain issues is to use back-to-back IP-TDM media gateways at each network domain border. Although this solution provides some level of control and protection between domains, it also has serious drawbacks. To begin with, the solution is limited to voice services only. No other IP multimedia services can be implemented this way. The method of terminating an IP/RTP encapsulated packet voice media stream and converting it into a synchronous TDM signal and then back to IP/RTP stream introduces additional delay. The TDM part of the solution can typically only handle only G.711 signals and this leads to the requirement to transcode the media stream in one or both of the media gateways. Transcoding is needed if common compression algorithms, like G.729 are used in IP part of the call leg. Each transcoding step adds distortion and delay to the signal, lowering audio quality and harming conversational quality. Transcoding requires DSP (digital signal processing), which adds cost to the implementation.

Firewalls are the typical network elements implementing security policy for an administrative IP network domain. VoIP and other interactive IP multimedia traffic require real-time packet delivery: short delay, low jitter and low packet loss end-to-end. Traditional data firewalls are not designed for real-time applications. Among other issues they have difficulties coping with NAT and IP multimedia protocols. NAT is required for service interworking across private and public IP addressing. Traditional firewalls that are not designed for SIP or other multimedia protocols used require that some well-known ports be opened for the signalling. In addition to the well-known ports however, a broad range of UDP ports must be permanently opened for the RTP media streams in order for the service to work. This essentially leaves the firewall open and is not really secure.

Firewalls that include ALGs (Application Layer Gateways) for SIP or other relevant protocols used solve the issue of permanent open port ranges that traditional firewalls cannot handle. The ALG approach works by interpreting the signalling to determine which ports need to be opened for each communication session. This also makes it possible to close the opened ports, when the session ends. Firewalls offer only a partial solution to the network border issues, leaving most of the issues unsolved.

The session border controller network element originated from the need to solve the domain border issues in order to implement robust and manageable IP multimedia services by carriers, service providers and enterprises. Many of the requirements emerging from the inter-administrative and technology borders are left open by the standards. If unsolved they form a technical obstacle for practical, commercial implementation of IP multimedia services on a large scale.

4.2 SIPPING SBC Definition Approach

The above domain border cases generate several issues to solve. One way to categorize them is presented by a draft [Cam05] by IETF SIPPING working group. This draft document lists three groups: Perimeter defence, Functionality not available in endpoints and Network management.

4.2.1 Perimeter Defence

Perimeter defence includes dealing with issues related to protecting and securing an administrative domain from neighbouring, interconnected administrative domains. Issues in this group are:

- Access control
- Topology hiding
- DoS prevention
- DoS detection

4.2.2 Functionality Not Available in Endpoints

Functionality not available in endpoints means the set of functions that is required or desired in deployment of a service, but is not solved by available standards based implementations. Functionality in this group is:

- NAT traversal
- Protocol interworking
- Protocol repair

4.2.3 Network management

Network management in this case focuses on the real-time requirements of the interactive IP multimedia communications.

- Traffic monitoring
- Traffic shaping
- QoS

4.3 Industry Centric SBC Definition Approach

A slightly different, more IP multimedia industry centric approach [Tel03] defining the role of session border controller and the problems it addresses starts with the definition of the concepts of session, border and control in the context of IP multimedia.

> *Session*: Any real-time, interactive voice video or multimedia communication using layer 5 IP signalling protocols such as SIP, H.323, MGCP or Megaco/H.248
>
> *Border*: Any IP-IP network border between two service providers or between a service provider and its end user customer/subscriber.
>
> *Control*: Functions spanning security, service assurance and law enforcement requirements.

This approach focuses on the above control functions: security, service assurance and law enforcement requirements

4.3.1 Security

The security functions aim to protect service infrastructure and customer supplier relationship from attack. IP networks suffer from a lack of trust on the network layer. A service provider must allow authorized users into its network and concurrently shield the service infrastructure from Denial of Service attacks.

The IP multimedia service infrastructure may consist of a large number of devices like softswitches, SIP proxies, H.323 gatekeepers, media gateways, application servers, etc. SBCs can be used on the infrastructure border for protection by only allowing access and traffic from authorized users to the service platform. A SBC uses provider’s signalling infrastructure to control network access based on layer 5 signalling messages instead of for example the layer 3 IP addresses. When communication is authorized for example by successful SIP registration, the SBC allows the media streams in and out by opening and closing firewall pinholes.

Service providers and carriers may also want to hide the actual implementation of their service platform and network topology from the outside world for security and business reasons. SBCs can implement topology hiding so that all traffic to and from the service platform appears to flow via the SBC making the various network elements of the internal infrastructure invisible from the external networks.

SBCs are used to protect the service infrastructure from overloading, by limiting the rate of incoming signalling messages to a configured value. In the case a softswitch for example, that is able to handle *n* calls per second, before overloading, the SBC in front of the platform can begin to gracefully reject new requests when a set threshold is exceeded.

Virtually all enterprises and many consumers use firewalls to protect their premise-based equipment, like workstations and servers. Firewalls however present a problem to IP multimedia services. Protocols like SIP, H.323 do not work across a firewall or NAT by default. SBCs offer various methods for NAT and firewall traversal. Some methods do not require any new customer-premise equipment or configuration. This helps the enterprises or consumers maintain a secure firewall configuration, while getting access to the services.

4.3.2 Service Assurance, Revenues, Profit

In most IP networks oversubscription exists not only on the customer access link, but also in many places of the network, like between DSL access multiplexers and the edge routers. Oversubscription is natural to packet transmission, but easily results in QoS parameters delay, jitter and packet loss, not suitable for real-time interactive IP multimedia applications. These applications require network QoS parameters to be bounded to some application specific values. Too high delay, jitter or packet loss results in low quality sound, or video or loss of interactivity in the service.

By utilizing SBCs it possible to change the way real-time traffic flows in the network to an overlay topology that is easier to control in terms of QoS and traffic engineering for the multimedia applications.

SBCs offer also admission control policies implemented on the signalling level to control the number of real-time media streams directed to a particular network destination. This makes it possible, not to exceed the network capacity – real-time flows cannot be oversubscribed without affecting QoS parameters, like packet loss.

SBCs offer session accounting and call detail record (CDR) generation. This information can be used for capacity planning and billing purposes. The CDR information can include QoS information in addition to the more traditional CDR information, like calling party, called party, call duration, time of day, etc.

4.3.3 Law Enforcement

In addition to the categories above, SBCs attempt to solve a group of requirements emerging from national legal and regulatory demands. These requirements are dependent on local legislation and regulation of national authorities. Typical requirements have to do with assisting the authorities in the form of legal intercept and delivery of call logs. The combined signalling and media routing features of SBCs can be used to implement transparent duplication and routing of media streams to the authority’s systems. The access control and network management performed enables the generation of call logs with detailed information that is required for law enforcement purposes in criminal investigation process in most countries.

Also there are requirements for handling emergency calls to emergency numbers, like 112 in the European Union and 911 in the United States. SBCs can help in emergency traffic delivery in congested networks.

4.4 SBC Deployment Scenarios

This chapter describes some scenarios, where SBCs are deployed in production networks today. This section is divided in two parts. The first takes a look at carrier and service provider deployments. The second focuses on the case of SBC in the enterprise. The scenarios presented here are the most typical ones found on common network borders, but the application of SBCs are not limited to these cases.

4.4.1 Carrier and Service Provider

Peering and Protocol Interworking Scenario

This case illustrates PSTN / PLMN origination, termination and IP transit in a carrier-to-carrier IP interconnect scenario.

In this example there are three operators A, B and C involved. Each of them has connectivity to the PSTN or PLMN and to IP network. Operators A and B have a SIP based infrastructure. Operator B has H.323 based infrastructure. Switching function is represented in Figure 6 by softswitches (SS) and gatekeeper (GK).

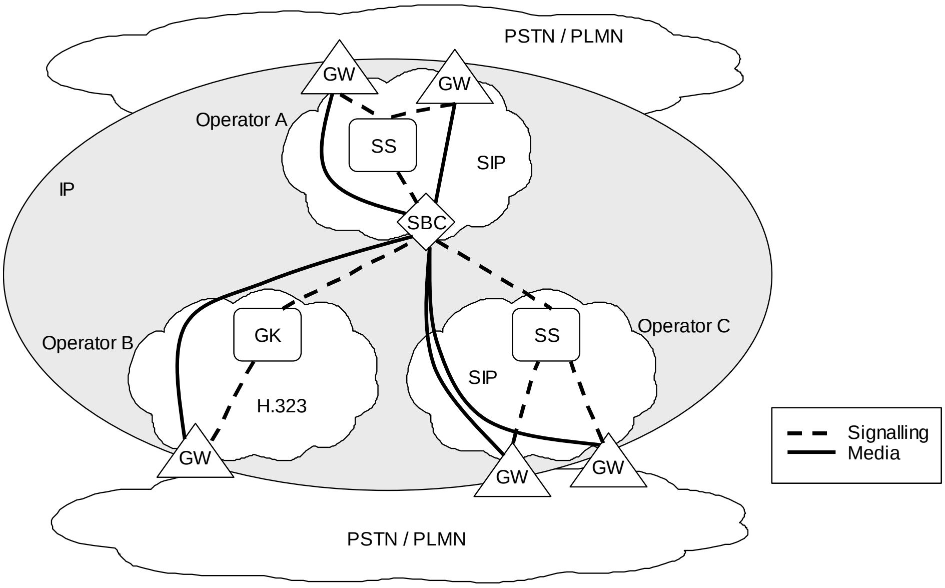

Figure 6 Three operator peering scenario

In Figure 6 operator A is peering with two other operators, operator B and operator C. It is using a SBC, that supports SIP and H.323 and because of this it has connectivity to both B and C.

Operators B and C do not have a SBC in their networks. They cannot exchange traffic directly with each other because they are using different protocols in their infrastructure, and have no network element to perform protocol interworking.

Operator A could operate as a peering point and a transit operator for B and C, as it can support for SIP and H.323.

In the case of operator A, all signalling and media streams are terminated in the SBC on the border of operator A IP infrastructure. The internal topology, the IP addresses or the number of distinct media gateways or softswitches is not visible outside. Traffic is allowed to flow in and out of A’s network under the control of SIP signalling by the softswitch. This protects the infrastructure from unauthorized use and DoS attacks. Only authenticated and authorized sessions from B and C are allowed in.

In the case of operators B and C, their internal infrastructures are visible outside of their network domains. Modifications to the internal topology, like routing, IP addressing and number of network elements are visible to the outside world.

Even though operators B and C do not utilize SBCs in their network, in a practical implementation they might be using static access control lists (ACL) in firewalls or routers, to limit access to the needed networks, such as the IP address space of operator A. This offers some level of protection, but leaves the system open for distributed denial of service attacks utilizing IP address spoofing. Keeping the ACLs up-to-date can also present a management challenge.

Data Centre Scenario

This scenario applies to service providers offering IP multimedia communication services, like voice services, multimedia conferencing or collaboration services, hosted IP PBX, etc. from their data centres.

In this example there are four parties involved: The IP multimedia application service provider, two enterprises, Enterprise A and Enterprise B, using different kind of business communication services and a residential customer.

The Enterprise A is using a business communication service sometimes called “IP Centrex”. This means, that the basic telephony service along with other business applications, like conferencing, voicemail / unified messaging or presence services are provided by the service provider from the network. The application servers used in the implementation are located in the providers hosting facility. Enterprise B is running its own enterprise communication service, sometimes referred to as IP PBX. It uses services from the service operator to connect the enterprise infrastructure to the rest of the world in order to e.g. make and receive calls from the mobile networks and the PSTN. This is analogous to using E1 based 30B+D ISDN TDM subscriptions from traditional service providers. The residential customer is using IP communication services with a consumer oriented feature set, like voice and video calls, instant messaging and presence.

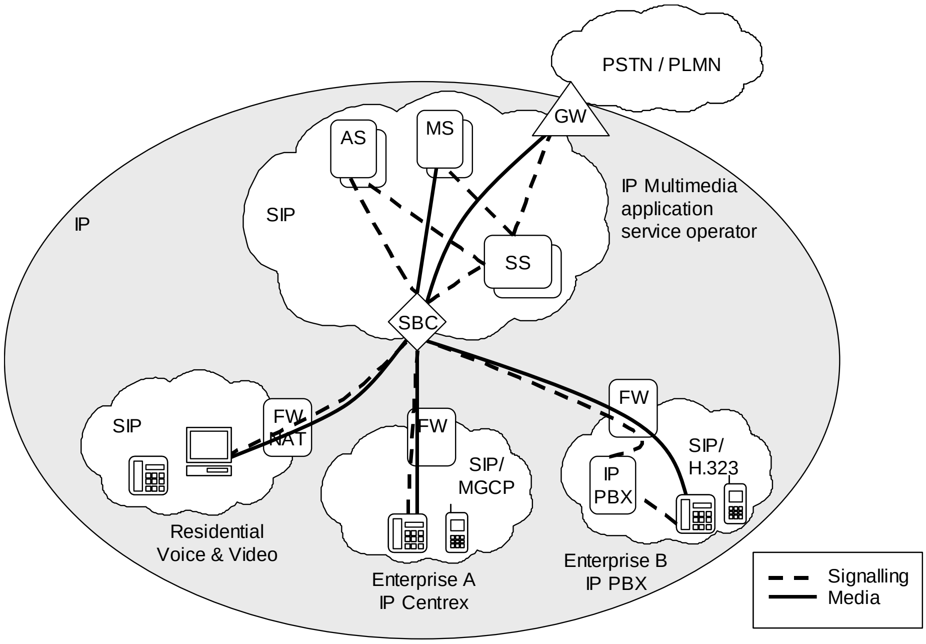

Figure 7 Service provider offering IP multimedia services

In this scenario shown in Figure 7, the SBC located at the edge of the operator’s service production network has two main roles: One for the service operator and one for the customer side of the network.

The main role for the operator is to implement perimeter defence. This means protecting the service production infrastructure from external network treats.

The second main role is to enable simple and secure access between the endpoints located in the three customer networks and the service provider’s platform. Issues in this area include manageable customer firewall and NAT traversal.

In this scenario all IP multimedia traffic between the customers and between a customer and the service provider flows through the SBC. This makes customer firewall rule configuration simple. The SBC can be configured as the source and destination of all IP multimedia sessions in the customer firewall. It is safe to assume that the SBC can act as a trusted node for most customers, because in order for a session to be allowed by the SBC, it has to be authenticated and authorized by the service provider’s signalling infrastructure. This is analogous to authentication of other subscription-based services found in mobile and PSTN networks.

In a case without SBC the customer firewall would have to accept traffic flows from all the IP addresses, the customer wants to communicate with. This means either opening up the firewall for all IP addresses of listing all the trusted addresses or networks. The first option is insecure and the second unmanageable for more than a handful of endpoint addresses, not to mention the impact of endpoints behind non-static IP addresses.

This scenario description focuses on security issues. The SBC can perform other functions described later in Chapter 5 of this thesis, but they are not elaborated here, as the above is sufficient in the scope of this chapter.

4.4.2 Enterprise

The previous scenarios focused on the deployment of SBCs from service provider point of view. Enterprises use SBCs to address security issues and problems created by the use of private IP addresses and NAT. The following deployment example is focused on the enterprise.

In this example, shown in Figure 8, there are three parties involved. The Enterprise A is central in this case. It utilizes a SBC for direct peering with another enterprise, Enterprise B. The SBC allows control over how much topology information is visible between the peering organisations and provides protocol interworking.

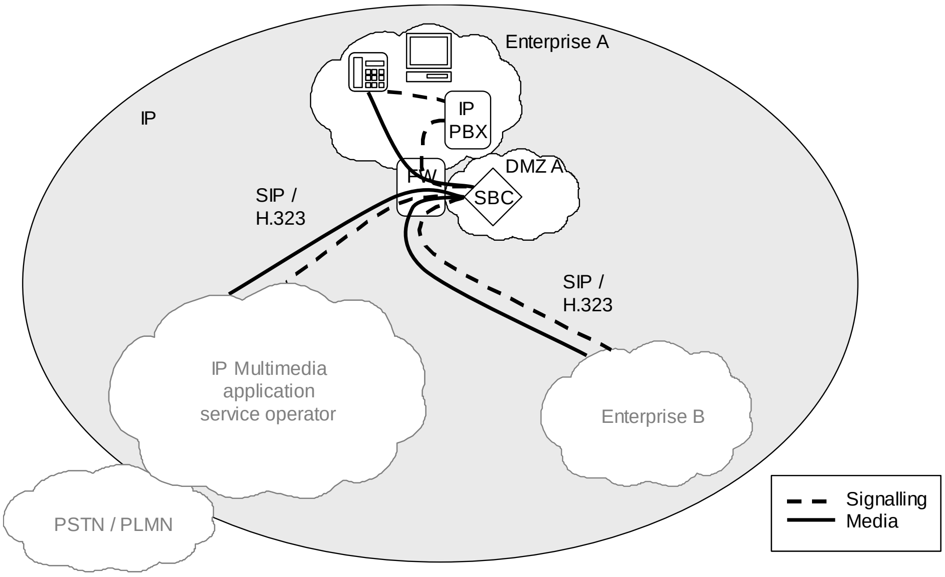

Figure 8 SBC deployed at the border of an enterprise network

5 SBC Functionality and Relationship to the Standards

This chapter reports the practical work done. The chosen method is identifying, analysing and comparing the functions of SBC to the functions specified in the standards. This is the main goal of the thesis.

This chapter takes a look at the elementary functions performed by a SBC. Functions are described briefly along with the implementation. Next relevant standards based implementations are described.

Also non-standard functionality is handled where identified. Analysis and interpretation of the SBC vs. standards will be presented.

The secondary goal of this thesis, which is to study the role of SBC in communication infrastructures of different standard making organizations, is done in the second half of this chapter.

5.1 SBC Architecture

The session border controller and its functions are not defined in any single standard and the functions vary from vendor to vendor. The architecture descriptions available in vendor product literature vary in the level of detail and focus of different aspects. This section presents some rough architecture descriptions available form the vendors. Also one high level model discussed in an IETF SIPPING working group draft is presented. These models are summarised and enhanced to formulate a generic reference model for the purposes of this study. This model is presented to enable viewing the SBC in a vendor independent manner and from several points of view.

5.1.1 SBC Architecture 1

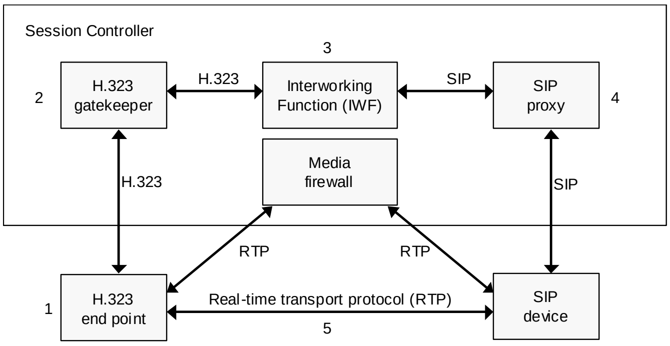

Figure 9 SBC architecture

The architecture description [Com04] in Figure 9 focuses on protocol interworking between H.323 / SIP and NAT/Firewall traversal. The two media paths in Figure 9 represent two scenarios. The direct RTP media path is used, when the H.323 and the SIP endpoint can communicate directly. The media paths between the endpoints and the SBC media firewall represent a NAT/firewall traversal scenario, where the endpoints can not reach each other due to firewall policy or NAT being used, but can be reached by the SBC NAT/firewall traversal.

The H.323 gatekeeper implements the functionality to interface with H.323 end points. The H.323 signalling takes place between the H.323 end point and the gatekeeper module in the SBC.

The SIP proxy module implements the functionality to interface with SIP end points. SIP signalling takes place between the SIP end point and the SIP proxy module in the SBC.

The H.323 and SIP interworking function (IWF) translates between the protocols and thus provides routing services between H.323 and SIP devices. When calls are placed between an H.323 and a SIP device, the SBC views each call as two legs: an ingress leg terminating on the IWF and an egress leg the IWF generates based on the protocol used by the destination.

The media firewall provides security and controls access to a provider’s network.

5.1.2 SBC Architecture 2

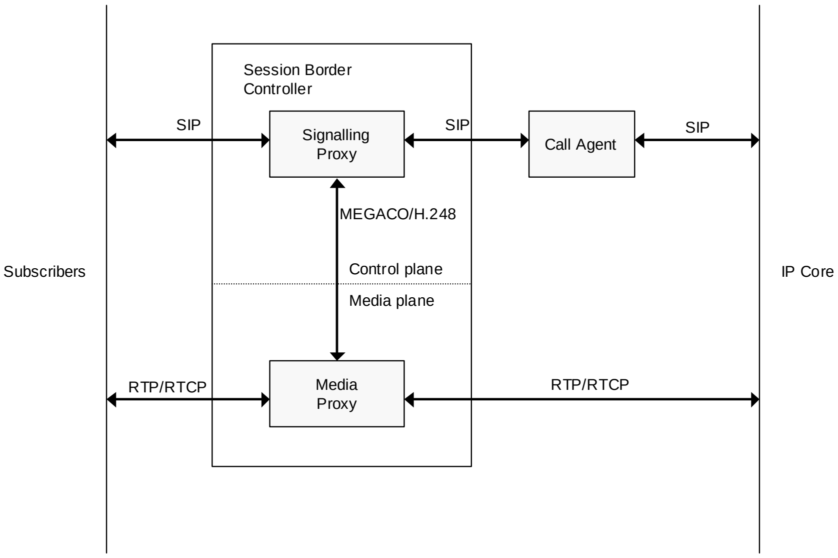

Figure 10 SBC architecture

This architecture description [New05] focuses on NAT traversal and isolation between the service provider network and the client side access network.

In Figure 10, the Signalling Proxy acts as a SIP B2BUA (Back-to-Back User Agent). It is configured as a transit point for SIP signalling messages between the client (User Agent) and the Call Agent (and vice versa). In this way, it acts as a proxy for both client and server. All signalling messages pass through it.

The Media Proxy operates under the control of the Signalling Proxy to provide a transit point for RTP and RTCP media streams between User Agents. All media is directed to the Media Proxy. The Media Proxy can also perform NAPT (Network Address and Port Translation).

The Signalling Proxy and Media Proxy exchange information using an internal Megaco/H.248 protocol.

5.1.3 SBC Architecture 3

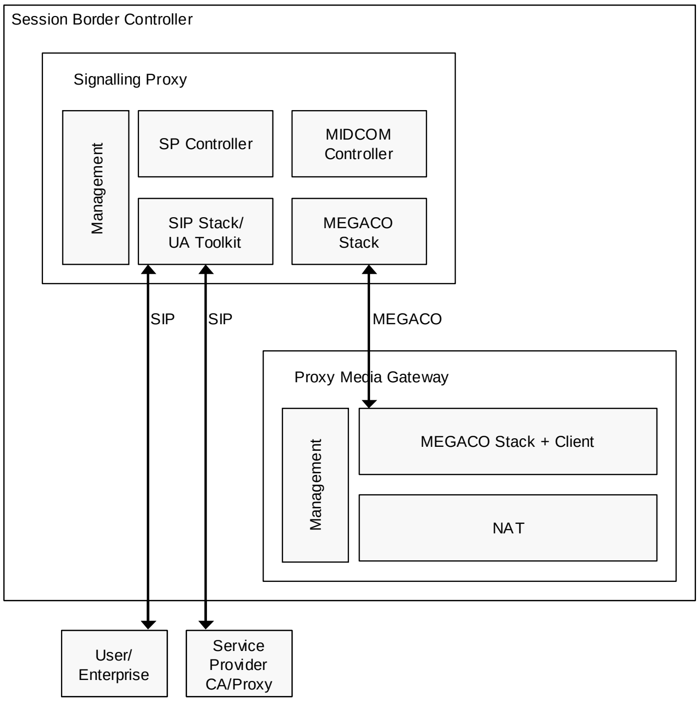

Figure 11 SBC architecture

Figure 11 describes a proposal for SBC architecture [Fle05]. It resembles the previous one, but has more information on the internal structure of the Signalling Proxy and the Media Proxy. Also this architecture decomposes the signalling and media planes into different entities, suggesting a possibility for a distributed architecture.

The SIP stack provides the SBC with the basic encoding-decoding capability to parse the SIP messages. A UA Toolkit, positioned on top of the basic SIP stack, facilitates call object level operation by the application (here, the SP Controller). The UA Toolkit relieves the application of functions such as realization of basic SIP procedures and semantic validation, in addition to syntactic validation of SIP messages and parameters.

The SP Controller acts as a routing entity for the SIP messages exchanged between the Call Agent and the User Agent. The SP Controller is primary responsible for channelling SIP messages received from the Call Agent to the SIP user and vice versa.

The MEGACO stack on the signalling proxy side is the interface to the media proxy side. The signalling proxy commands the media proxy to add, modify, or subtract RTP/RTCP sessions.

The MIDCOM Controller analyzes the SDP payload in the SIP message and sends corresponding MEGACO control commands to the Proxy Media Gateway via MEGACO stack.

The Proxy Media Gateway acts as a 'Middlebox' as defined in the MIDCOM architecture. The Proxy Media Gateway analyzes the RTP/RTCP ports and IP Address sent by the Signalling Proxy, opens/closes the corresponding ports, and then returns a new SDP back to the SIP proxy containing the addresses and ports used by the Proxy Media Gateway for the current session.

5.1.4 SBC Architecture 4

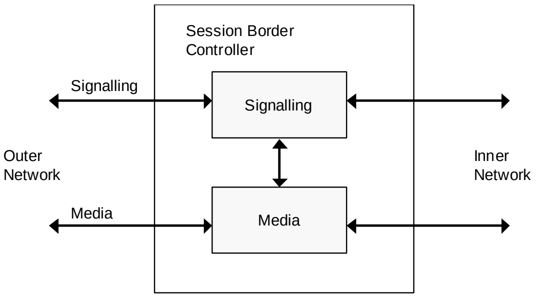

Figure 12 SBC architecture

The logical architecture in Figure 12 is used in an IETF SIPPING working group draft [Cam05] document discussing the functions of current SBCs. This architecture resembles the two previous ones described, but has less detail. The logical structure is the same as in 2 and 3.

5.1.5 SBC Reference architecture model

As can be noticed from the above models, the architecture descriptions available in vendor product literature and the IETF draft documents vary in the level of detail and focus on different aspects. The diversity of logical architecture models makes analysing the SBC difficult.

In order to have a single point of reference for the purposes of this thesis, the following logical architecture model is defined. Its goal is to describe sufficient building blocks for approaching the SBC in a generic way. This model does not describe directly any practical real-world implementation. It is presented here in order to help the reader visualize and understand how the functions performed by a SBC might be achieved. The model is a generalization based on models from the following sources [Cam05], [Tel03], [New05], and [Fle05].

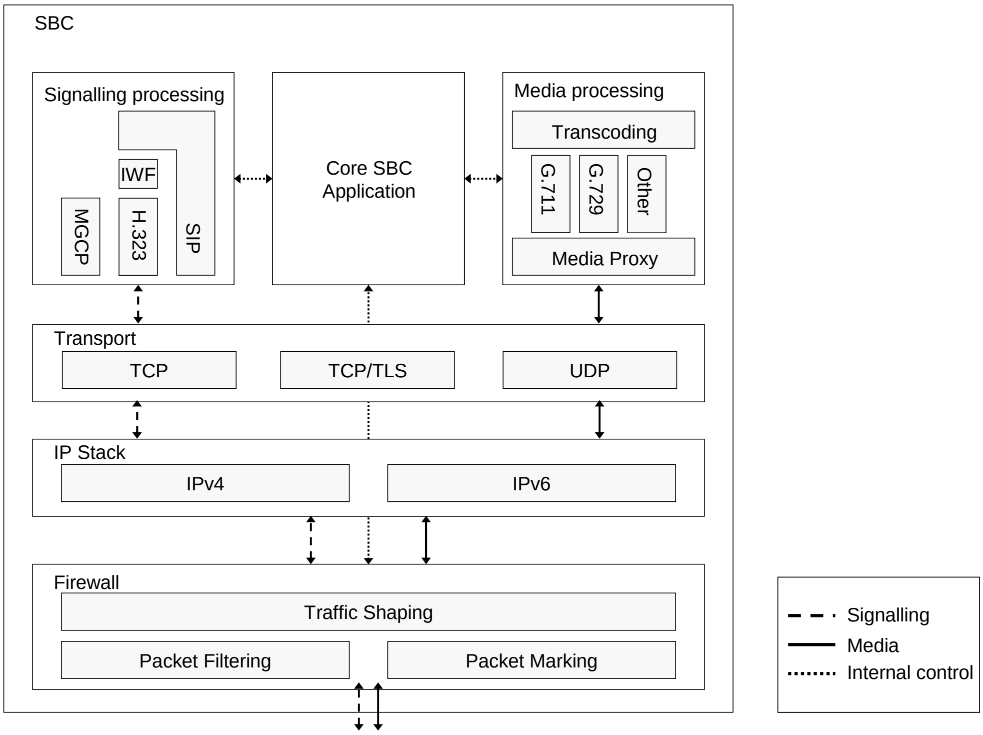

Figure 13 SBC reference model

The dotted lines in Figure 13 represent a control relationship between Core SBC application and the functional modules. The dashed line represents signalling and the solid lines media flows.

In typical cases, like the ones described in deployment scenarios, SBCs are located at the border of two networks. The network interfaces however are not included in Figure 13 describing this model.

Signalling Processing

This module handles the signalling protocol processing for the communication sessions, in which the SBC participates. It contains signalling stack implementations for all the supported protocols, like SIP and H.323.

This model assumes, that all non-SIP signalling protocols are first mapped to SIP via a protocol specific interworking function (IWF) before further processing. By making this assumption it is possible to limit the analysis of the SBC functions to SIP only, while maintaining the multi-protocol nature of most SBC implementations.

Media Processing

This module handles media processing related to the communication sessions flowing through the SBC. The Media proxy implements termination and regeneration of media streams. The transcoding module implements interoperation between different media coding formats. This is done by first decoding the incoming media stream using its native format and then re-encoding it using another format.

IP Stack

IP stack contains the implementation of the TCP/IP protocol. A dual stack is used for interfacing to both IPv4 and IPv6 networks.

Firewall

The firewall module implements traffic management functions by allowing and preventing communication using packet filtering performed under control of the Core SBC application. In this model traffic shaping and marking are placed inside the firewall module. Traffic shaping performs rate limiting and inter-packet delay normalizing for ingress and egress traffic. Packet marking performs QoS marking of egress traffic, to enable proper treatment in the external transport network.

SBC Core Application

The SBC Core Application is the most complex part of the model. SBC devices can operate from the SIP point of view as proxy, B2BUA, or a hybrid of both proxy and B2BUA. This functionality is implemented in this model by the SBC core application. This module is also responsible for coordinating the overall operation of other modules: Signalling processing, Media Processing, IP Stack and the Firewall.

5.2 Session Border Controller Functions and Implementation

This section and the subsections deal with the main goal of this thesis: What SBC functionality is standard behaviour and what is non-standard? The functions are first identified and then each function is described along with motivation why the function is performed.

The IETF draft [Cam05] lists the following SBC functions. SBC vendor material lists additional functions and both are presented in Table 3. A summary of vendor material used is presented in Appendix A.

SBC functions by IETF	**Additional functions in vendor material**
Access control	Call Admission Control (CAC)
Topology hiding	DoS detection and prevention
Traffic monitoring and shaping and QoS marking	Overload prevention
Protocol repair	Media Transcoding
Protocol/profile interworking	Law enforcement, emergency traffic
Transport protocol interworking	
NAT traversal	

Table 3 SBC functions

The functions from both sources above are discussed in this thesis. However, as the SBC concept is not specified by any universally accepted definition, the list of functions is not complete. Devices under the name of SBC, or its variations might perform functions, not listed above.

5.2.1 Access Control

The access control function makes it possible to control gaining access to the services provided by the service platform. The decision of granting or denying access can be based on IP addresses or address ranges, like traditional firewalls, and this kind of static filtering is implemented as an administrative task in SBC configuration. In addition to static firewall configuration, SBCs typically implement access control based on signalling. This means, that access to the platform behind the SBC may be controlled and granted only if the endpoint that is the source of the signalling messages can successfully authenticate itself. The authentication mechanism is specific to the signalling protocol used. SIP, H.323 and MGCP protocol specifications specify the methods for each protocol.

5.2.1.1 Access Control Approaches

Access control by filtering IP traffic can be applied only to the signalling, or to both the signalling and the media. If it is applied only to the signalling, then the SBC can be thought to operate as a proxy server. On the other hand, if access control is applied to both the signalling and media, then the SBC can be thought to operate as B2BUA and media proxy. A key part of media-layer access control is that only media for authorized sessions is allowed to pass through the SBC.

If the access control is applied to both signalling and media, then firewall pinholes are dynamically opened and closed for media and authorized signalling after authenticating with an external signalling node. This helps to limit the traffic entering the service platform to flows generated by authenticated endpoints, thus limiting the impact of denials of service (DoS) and other network based attacks

In both of the cases, proxy and B2BUA, the SBC needs to handle every single signalling message. This function has scalability implications. In addition, the SBC is a single point of failure from the architectural point of view. Many current SBCs, however, have redundant configuration, which prevents the loss of calls/sessions in the event of a failure.

The nodes used as the sources of access control information can be SIP proxies, registrars or H.323 gatekeepers in the service platform. They have to be trusted by the service provider to contain correct information and configuration [Jun05].

The signalling based access control function can be achieved by co-operation of the firewall packet filter module, the signalling processing module and the core SBC application.

The following Figure 14 shows a successful registration example with SIP [Joh04], [Cam05].

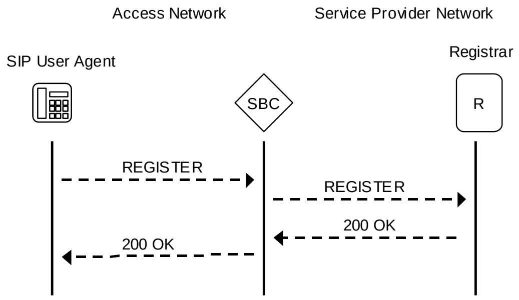

Figure 14 User Agent registration

1. SIP User Agent sends REGISTER request to the SBC
2. The SBC modifies the REGISTER request by inserting itself to the path header field, or modifying the original UA Contact header field to point at the SBC instead of the original. Then the SBC sends the modified REGISTER to the SIP registration server, also known as a registrar.
3. The registrar responds to the SBC with SIP 200 OK response to indicate successful authentication
4. The SBC responds to the User Agent with 200 OK response

Before a valid authentication, the UA located in the access network has very limited access to the service provider service platform network. It can only access the access network side of the SBC. Access to the rest of the platform is blocked without valid authentication. Figure 15 below illustrates, how the UA, SBC and registrar and other SIP nodes in this example are located.

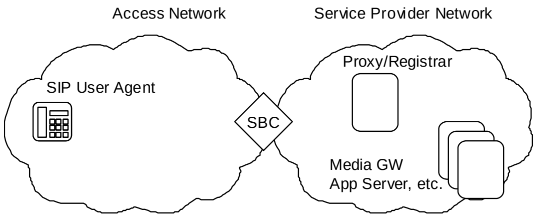

Figure 15 Network topology

The following call flow in Figure 16 illustrates the creation and teardown of a SIP session with SBC in the path. A user agent A initiates the session creation and teardown.

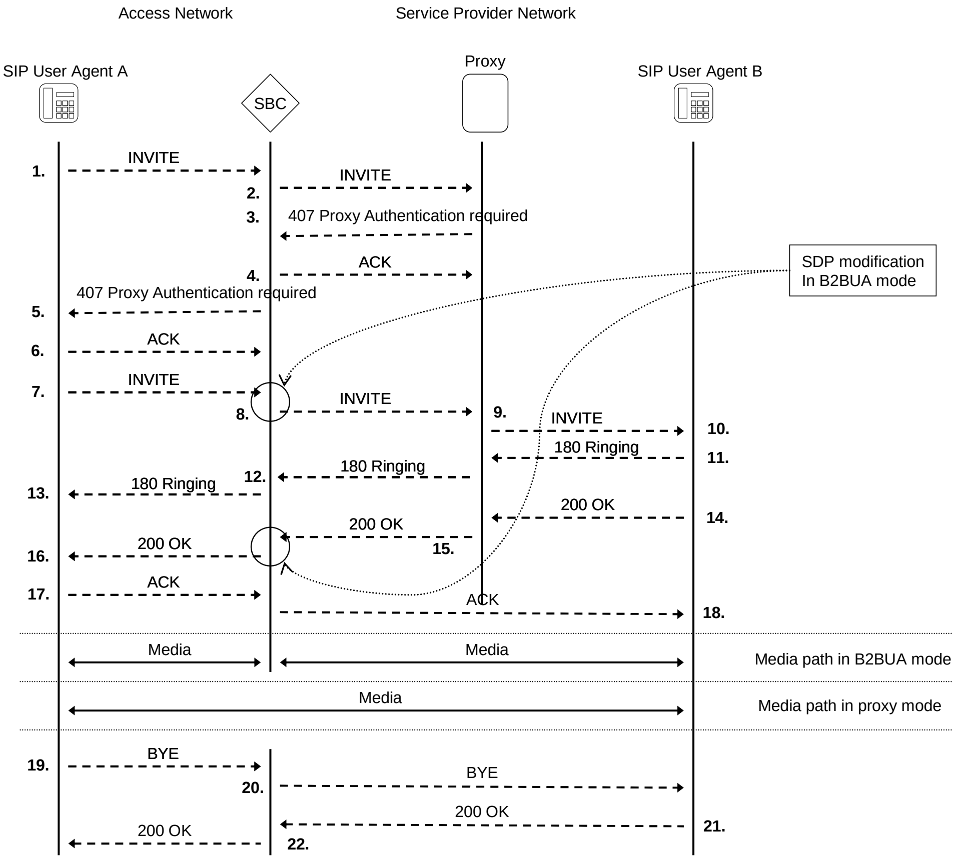

Figure 16 SIP session with SBC in the path

The following list explains what happens in Figure 16.

1. The User Agent A sends an INVITE to the SBC without Proxy-Authorization header field
2. The SBC forwards the INVITE to the proxy server.
3. The proxy requires authentication, and responds with 407 Proxy Authentication Required response containing challenge information
4. The SBC receives the response and sends ACK to the proxy
5. The SBC forwards the with 407 Proxy Authentication Required response to the User Agent A
6. The User Agent A send ACK to the SBC
7. The User Agent sends an new INVITE carrying the credentials to the SBC

8. The SBC forwards the INVITE to the proxy server.
9. The proxy receives the INVITE with appropriate credentials
10. The proxy processes the entry matching its own realm, leaving the remaining entries intact and forwards the INVITE to the User Agent B
11. The User Agent B responds with a 180 Ringing response to the proxy
12. The proxy forwards the 180 Ringing response to the SBC
13. The SBC forwards the 180 Ringing response to the User Agent A
14. When the User Agent B answers the incoming call a 200 OK response is sent to the proxy
15. The proxy forwards the 200 OK response to the SBC
16. The SBC forwards the 200 OK response to the User Agent A
17. The User Agent A responds with ACK to the SBC
18. The SBC forwards the ACK to the User Agent B

At this time both way media path is established. The call flow in Figure 16 shows two media path cases. The first case illustrates operating in B2BUA with media proxy mode and the second proxy mode.

In B2BUA mode, the media path setup is not end-to-end – the SBC terminates both media streams and bridges them. In order to terminate the media streams, SBC modifies the SDP carried in the SIP messages to point to itself instead of the original user agent. The SDP is modified before forwarding it to the direction of the destination user agent. The modification points are illustrated in the Figure 16.

In the SBC reference model the SDP information is passed from the Signalling Processing to the SBC Core Application which implements the B2BUA. The B2BUA makes the SDP modifications and passes the information back to the Signalling Processing to establish the other SIP session. In addition to the SDP modification, the Media Proxy function is configured for bridging of the two individual media streams according to the SDP information.

In proxy mode the media path setup is end-to-end between user agents and no SDP rewriting is done. The SBC can either be in the media path, or not. In a case where the

SBC is in the path, access control of the media can be done. In this case the SBC merely forwards the authorized RTP packets carrying the media stream as a traditional firewall would and drops the packets that have not been authorized.

In the SBC reference model SDP information is passed to the SBC core application, which in this case implements proxy functionality. In this case however, no signalling modification is performed. The SDP information is used to open firewall pinholes on the Firewall module in order to allow the RTP packet to be forwarded.

The User Agent A initiates the teardown of the session.

19. User Agent A sends BYE to the SBC
20. The SBC forward the BYE message to the User Agent B
21. The User Agent B responds to the SBC with 200 OK
22. The SBC forwards the 200 OK response to User Agent A

When the session ends, the SBC removes the media-bridging configuration of B2BUA mode or the firewall pinholes created in proxy mode. From an access control point of view, the access of User Agent A through the SBC is blocked until a new authorized session is granted.

5.2.1.2 Standards Based and Non-Standard Approaches

The implementation of access control function in current SBC devices can be vendor specific or standards based. The proxy based approach where no SDP modification takes place can be considered standards conforming, if the SBC acts as a proxy defined in [Ros02]. If however the SBC operates in the role of SIP proxy, but modifies SDP information, as is the case in the B2BUA mode, the operation is non-standard. SDP modification is not allowed for proxies. If a SBC is transparent to user agents, it should not modify SDP.

Some SBCs utilize the MIDCOM [Sri02] approach for controlling an external firewall. MIDCOM stands for Middlebox Communication, and is being developed by the IETF Middlebox Communication Working Group. MIDCOM seeks to enable trusted third parties make policy decisions on behalf of the various entities participating in an application's operation. The objective of the MIDCOM approach to enable complex applications such as IP multimedia, through the middleboxes, seamlessly using a trusted third party.

The trusted third parties in the case of IP multimedia are the trusted signalling nodes e.g. SIP proxies or registrar servers. The entities participating in an applications operation can be e.g. firewalls protecting the service platform of a service provider.

The concept of a Middlebox is defined in [Sri02] as follows:

A Middlebox is a network intermediate device that implements one or more of the middlebox services. A NAT middlebox is a middlebox implementing NAT service. A firewall middlebox is a middlebox implementing firewall service. Traditional middleboxes embed application intelligence within the device to support specific application traversal. Middleboxes supporting the MIDCOM protocol will be able to externalize application intelligence into MIDCOM agents. In reality, some of the middleboxes may continue to embed application intelligence for certain applications and depend on MIDCOM protocol and MIDCOM agents for the support of remaining applications.

5.2.2 Topology Hiding

Topology hiding means hiding information related to internal topology of a network or service platform domain when observed from outside the domain. In order to hide the internal topology, all IP packets emerging from that domain must have a source address that belongs to the network element implementing topology hiding. In addition to the IP addresses in packet headers, signalling protocols carry topology information inside the actual signalling messages. In order to keep the topology hidden, this address information must also be modified to refer to the border element at the edge of the domain and not the original source. [Jun05], [Acm05].

5.2.2.1 Topology Hiding Approaches

To implement topology hiding the SBC operates as B2BUA and media proxy terminating all signalling and media streams on both sides of the border. As the session is fully terminated and regenerated on both sides, it is possible to control all information transferred in signalling. In the case of SIP the fields like Contact, Via, and Record-Route contain topology information. The actual IP packets generated have the address of the SBC as well. The same is true for the media streams that are terminated and regenerated by the SBC.

In the SBC reference model, the topology hiding could be implemented in a similar way to the B2BUA scenario of access control function described previously in 6.2.1. In this case, not only the SDP carried in SIP signalling is modified by the SBC Core Application, but also all address and routing related information carried in the SIP header fields like Contact, Via, and Record-Route.

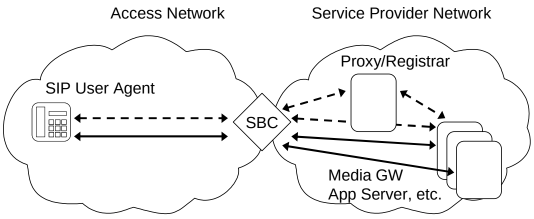

Figure 17 Topology hiding

Topology hiding can result in completely rewritten SIP signalling messages. Figure 17 illustrates the signalling and media flows on both sides of the SBC. Solid lines are media streams and the dashed lines signalling.

The next example in Figure 18 shows two SIP INVITE messages. The messages are actually the “same” INVITE before and after processing by a SBC performing topology hiding. The first message is the original INVITE generated by the SIP service platform and the second one has been processed by a SBC with topology hiding function.

It can be observed, that the two invites look quite different. There are many modifications in the message. One of the most apparent changes is that header field names have been modified to compact SIP representation e.g. Via is converted to v, Call-ID to I, etc. The use of compact SIP is not related to topology hiding, but is performed for other reasons. The modifications directly related to topology hiding are in the following parts of the message: modification of SIP URI in INVITE, removal of Record-Route header, modification of Via header, modification of Contact header, modification of SDP addresses.

No addresses of the original service platform are present in the message headers or SDP after topology hiding. Figure 18 illustrates this.

```
INVITE sip:80.222.48.106:5060;transport=udp;ua=bcdfcb8a5c16502421cc0e9d941e53be SIP/2.0
Max-Forwards: 10
Record-Route: <sip:395231@69.90.155.70;ftag=94919237389882988;lr=on>
Via: SIP/2.0/UDP 69.90.155.70;branch=z9hG4bKcc9c.0855ddd1.0
Via: SIP/2.0/UDP 69.90.168.5:5060
Call-ID: 484992572@conf1.conference.libretel.com
CSeq: 2 INVITE
From: "fwdusers@pulver.com" <sip:513@fwd.pulver.com>;tag=94919237389882988
To: sip:395231@fwd.pulver.com
Contact: sip:69.90.168.5:5060
User-Agent: eDial Server
Proxy-Authorization: Digest username="513", realm="fwd.pulver.com", algorithm=MD5, uri="sip:395231@fwd.pulver.com",
               nonce="42ffb9b9ddf1c6425d8f06933e425762aea88ca0", response="ed6811985d2642fbb1c9045e0984e931"
Content-Length: 208
Content-type: application/sdp

v=0
o=513 94919237389882988 1 IN IP4 69.90.168.5
s=phone-call
c=IN IP4 69.90.168.5
b=CT:1000
t=0 0
m=audio 12004 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

```
INVITE sip:395231@192.168.0.37:5060 SIP/2.0
v: SIP/2.0/UDP 80.222.48.106:5060;branch=z9hG4bK-bdffbc418aaa30fb0854fb54b72b0155
f: "fwdusers@pulver.com" <sip:513@fwd.pulver.com>;tag=94919237389882988
t: <sip:395231@fwd.pulver.com>
i: 484992572@conf1.conference.libretel.com
CSeq: 2 INVITE
Max-Forwards: 9
m: <sip:e07d8890de5d57c02e0f9b854edff3fd@80.222.48.106:5060;transport=udp>
Proxy-Authorization: Digest username="513", realm="fwd.pulver.com", algorithm=MD5, uri="sip:395231@fwd.pulver.com",
               nonce="42ffb9b9ddf1c6425d8f06933e425762aea88ca0", response="ed6811985d2642fbb1c9045e0984e931"
User-Agent: eDial Server
c: application/sdp
l: 204

v=0
o=513 586782316 1 IN IP4 80.222.48.106
s=phone-call
c=IN IP4 80.222.48.106
b=CT:1000
t=0 0
m=audio 49186 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Figure 18 SIP INVITE before and after topology hiding

Without topology hiding many of the endpoint IP addresses in the service provider network would be visible to the user agent located in the access network. Some addresses might be visible directly as session and media termination addresses. Some addresses could be seen by looking at the SIP header fields received by the user agent. Topology hiding limits the visibility of internal structure of the service platform to the addresses of the SBC. This is desired by some service providers in order to protect the service platform from DoS attacks, or for commercial reasons.

5.2.2.2 Standards Based and Non-Standard Approaches

Topology hiding function is based on the B2BUA concept defined in SIP [Ros02]. The SIP specification does not define the topology hiding functionality, but only the concept of B2BUA as being a concatenation of a UAC and UAS. Another RFC in the category

Best Current Practice [Joh03] describes a session through a SIP application layer gateway (ALG). This section mentions the use of B2BUA as part of an anonymizer service, in which all identifying information of the calling party would be removed. This is roughly equal to SBC topology hiding functionality, but the IETF material does not specify what the anonymizer service is allowed or not to do.

A 3GPP technical specification IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP) [3GP05] describes a function called Topology Hiding Inter-network Gateway (THIG). This function is part of the I-CSCF in IMS. The goal of THIG is similar to the topology hiding functionality available in SBC devices in the sense, that it hides the configuration, capacity, and topology of the network from the outside. No SBC vendor material encountered explicitly state support for IMS THIG.

Session border controller implements topology hiding by removing original header fields and replacing them with addresses and pointing to the SBC itself, as described earlier. THIG method on the other hand is based on utilizing header encryption. This method is described in [3GP05]. Header fields which are applicable for THIG are encrypted as encryptedtoken@mydomain and inserted in the message with the following extra parameter: tokenized-by=mydomain.

5.2.3 NAT and Firewall Traversal

The NAT and firewall traversal is one of the most complex areas of SBC functionality. The background of the complexity comes from the number of different possibilities in NAT and firewall configurations, different deployment models of the service platform and variation in UA configuration and functionality.

The ultimate goal of SBC NAT and firewall traversal is to enable two users to communicate regardless of network topology or configuration. Issues in this area concern the presence and type of NAT, firewalls and the firewall policies. Communication should be achieved without major reconfiguration or upgrades of network elements or violation of existing security policies implemented by firewalls.

IP multimedia protocols must carry IP addresses, domain names and ports in their signalling messages to describe the sessions they are controlling [Ros00]. The port numbers used are often dynamic, like the UDP ports used to carry RTP to establish streams for audio and video. There are two core issues in NAT and firewall traversal. The first is to enable the signalling to pass through. The second is getting the actual media session that the signalling controls through.

The case of getting signalling through is affected by firewall type, policy and NAT. IP multimedia signalling takes place over TCP or UDP depending on the protocol. SIP can use TCP or UDP [Ros02]; H.323 uses TCP for signalling using H.225 [Kum01]. MGCP uses UDP [And03].

The media session traversal of NAT and firewall is affected by the same things as in the signalling case with the exception that media transport typically takes place over UDP and not TCP. This is true regardless of signalling protocol used.

The connection oriented nature of TCP and the connectionless nature of UDP require a different approach for traversal. In general it is easier to deal with TCP session opened from inside the firewall to the public network to well known signalling nodes, than it is to the media streams, which may originate from any of the clients that are the other party of communication. These clients are distributed around the public IP address space.

Problems related to NAT and firewall transfer emerge from the following issues [Stu04]:

- Dynamic port allocation
- Embedding transport addresses in the message body
- End user private IP addresses
- Sessions initiated from the public network to a private network

5.2.3.1 Firewall Policies

A firewall policy can be implemented in an infinite number of ways. From IP multimedia signalling and media transport point of view, the policy can vary from very permissive to very restrictive. In the permissive case no special attention or action is required to traverse the firewall. In the restrictive case, firewall traversal might be completely impossible. Most practical firewall policies can be positioned in between the permissive and restrictive end of the spectrum.

Typical firewall rules that the policies are built from allow or deny traffic based on IP protocols like TCP or UDP, IP addresses, protocol port numbers and the direction of communication. Timers are also used to close bi-directional communication sessions, which have been idle for a time specified in the policy.

For the purposes of this thesis, the following assumptions on generic firewall policy are made.

To be able to establish IP multimedia sessions across a firewall, the following minimum requirements must hold:

1. If TCP based signalling is used, it must be possible to establish a TCP session for signalling across the firewall by a client located in the internal network to a well known TCP port in the signalling node located in the external network.

2. It must be possible to deliver UDP packets across the firewall by a client located in the internal network to a well known UDP port in a node located in the external network. Once the client has delivered a UDP packet to the node in the external network, it is possible for that node to deliver UDP packets through the firewall to the client at least using the source port and IP address of the client as the destination.

If these assumptions are correct in the network under study, it is possible to implement NAT traversal without modifying the firewall rules using a reachable relay host.

If TCP session establishment, as specified in assumption 1, is not possible, no IP multimedia signalling sessions using TCP can be established. If no bi-directional UDP traffic as specified in assumption 2 is possible, no bi-directional signalling or media transport flows can be established across the firewall.

Internal network is the network behind the firewall and external the public untrusted network. A firewall policy specifies how traffic may flow between the internal and the external network.

5.2.3.2 Different Types of NAT

NAT implementations and configurations found in routers and firewalls vary. The following treatment of UDP has been observed in implementations [Ros03]. The four types of NAT observed are:

Full Cone: A full cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, any external host can send a packet to the internal host, by sending a packet to the mapped external address.

Restricted Cone: A restricted cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a full cone NAT, an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

Port Restricted Cone: A port restricted cone NAT is like a restricted cone NAT, but the restriction includes port numbers. Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.

Symmetric: A symmetric NAT is one where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

Determining the type of NAT is important, because in order to achieve NAT traversal different approach may be needed in different cases.

5.2.3.3 NAT and Firewall Traversal Approaches in SBCs

Different SBC implementations approach the NAT and firewall traversal problem using standard and proprietary methods. Several standard methods exist in the form of RFC or an Internet Draft. Some methods, like STUN [Ros03] offer a solution for a limited number of network configurations. Some like TURN [Ros03a] and ICE [Ros03b] provide more complete solutions that work in nearly all cases, but introduce complexity. Using STUN, TURN or ICE the standard way, explicitly requires that the client application implements support for it. As these methods are not a part of any of the multimedia signalling protocols, their support by client applications or devices varies greatly. SBCs utilize these methods in different ways and in different combinations to provide the traversal functionality.

In addition to the standards, a SBC typically implements NAT traversal in a proprietary way too. One reason for this is the lack of client support for the standard methods. This is important, as typical service providers of IP multimedia services would like to be able to reach as many users as possible using different client applications and devices many lacking support for the traversal methods. Another reason behind using proprietary solutions is the immaturity of the standards. STUN is defined by a RFC, but TURN and ICE are still evolving Internet Drafts.

The following sections describe the standard and proprietary methods used for NAT and firewall traversal by SBC devices.

5.2.3.4 Standard Approaches to Firewall and NAT Traversal

The following sections presents the standards based approaches for NAT and firewall traversal, that are found in SBCs

STUN

Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (STUN) is specified by RFC 3489. It describes STUN in the following way:

Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) is a lightweight protocol that allows applications to discover the presence and types of NATs and firewalls between them and the public Internet. It also provides the ability for applications to determine the public Internet Protocol (IP) addresses allocated to them by the NAT. STUN works with many existing NATs, and does not require any special behaviour from them. As a result, it allows a wide variety of applications to work through existing NAT infrastructure.

Using STUN allows the client to discover the presence and the type of NAT in the network between the client and the STUN server. A client can discover the mapping between its private IP address and port and the public IP address and port that is visible in the public Internet. STUN requires that support for it is implemented in the client device or application.

Typically, service provides operate a STUN server in the public Internet. These servers can be located by clients using DNS SRV records by querying the provider’s domain for a service name “stun”.

When the application or device starts, the embedded STUN client sends a STUN Shared Secret Request to its server. The response from the server contains a username and password to be used in subsequent client-server communication. This initial negotiation of shared secrets is done using TLS over TCP and the server should present a site certificate that the client can use to verify, that it has connected to the intended STUN server. After obtaining the shared secret form the server the client sends a Binding Request to the server.

The discovery process of STUN consists of three tests and is executed according to the flowchart in specification [Ros03]. From the results gathered in the discovery process, the client can determine the following information on the type of NAT or firewall in the network between the client and the server:

- On the open Internet
- Symmetric UDP Firewall – Firewall that allows UDP out, and responses have to come back to the source of the request (like a symmetric NAT, but no translation)
- Full-cone NAT
- Restricted cone or restricted port cone NAT
- Symmetric NAT
- Firewall that blocks UDP

After determining the network environment, the STUN client can obtain NAT/firewall bindings for use with signalling and media transport for the IP multimedia application. The Binding Request packets must be sent from the same IP address that the client uses for the IP multimedia application, because the port and address mapping between that particular private IP address and public IP address is what is required for NAT traversal.

After receiving a Binding Request, the server sends Binding Response to the client. This response contains the public IP address and port number from which the Binding Request was received. By combining the public IP address and port information received in the Binding Response with the private IP address and port the client used when sending the Binding requests a mapping between private and public addresses can be made.

The IP addresses and ports obtained with STUN can then be used by other protocols for getting UDP flows across NATs and firewalls. In this case, IP multimedia protocols, like SIP, MGCP and H.323.

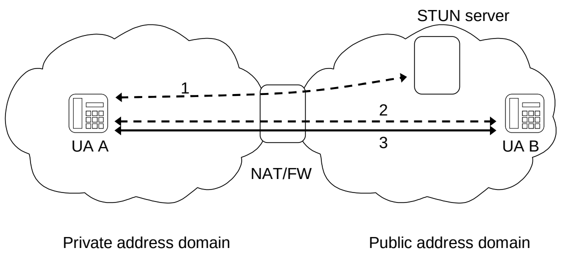

Figure 19 A user agent establishing communications using STUN

In Figure 19, the User Agent A (UA A) is equipped with STUN support and resides behind NAT. In order to communicate with UA B, it needs an address, which is reachable by UA B. In order to do this, UA A uses STUN to obtain a public IP address and port that it can receive packets from. Then UA A uses this address and port in the signalling to tell UA B, residing in the public address domain, where to send its media stream intended for reception by UA A (2). The media stream is established (3).

However the binding acquisition of STUN does not work for all NAT/firewall types. It will work for any application for full cone NAT only. For restricted cone and port restricted cone NAT, it will work for some applications depending on the application. For symmetric NAT, the binding acquisition will not work at all. Also, if there is a firewall configured to block UDP, STUN is of no use. IP multimedia applications typically depend on RTP/UDP for media transport, and blocking UDP completely prevents media streams form flowing between users.

TURN

Traversal Using Relay NAT (TURN) is specified by an Internet-Draft [Ros03a]. It describes TURN in the following way:

Traversal Using Relay NAT (TURN) is a protocol that allows for an element behind a NAT or firewall to receive incoming data over TCP or UDP connections. It is most useful for elements behind symmetric NATs or firewalls that wish to be on the receiving end of a connection to a single peer.

Using TURN enables a client to obtain a transport IP address and port from the public IP address space, while residing in a private IP address space behind NAT. The TURN server acts as a relay between two clients. For some NAT topologies such as a client behind a symmetric NAT or communication between two user agents both behind port restricted cone NAT, using a relay located in the public Internet is the only approach that allows communication to take place.

TURN resembles the STUN protocol in many aspects: It uses same message syntax as STUN, although it defines additional messages. It can use a similar DNS SRV record based discovery mechanism as STUN. The method of negotiating shared secrets for authorized request and response delivery is identical to STUN. TURN can be thought to compliment STUN in order to create a complete solution that will work with all types of NAT.

Although TURN will almost always provide connectivity to a client, it comes at a cost to the provider of the TURN server. As turn operates as a relay for media streams, all the active streams are routed through the TURN server. Without careful network design, this may lead to sub-optimal routing and performance problems. It is therefore desirable to prefer other methods like STUN or direct connectivity between clients over TURN and use it only as a last resort.

When an application starts, it first discovers the address of the TURN server. This can be preconfigured or discovered using DNS SRV records. TURN uses a similar mechanism for mutual authentication and integrity checks for both requests and responses, as STUN. Once the address of the TURN server is known the client acquires a shared secret to use and sends a TURN Allocate request to the TURN server. In response to this request the TURN server returns a public IP transport address. Packets sent to this public IP address are relayed to the TURN client by the TURN server. However, the TURN server will not relay any packets to the client until the client sends a packet through the TURN server towards a correspondent. To do that, a client sends a TURN SEND command, which includes a data packet and a destination IP address and port. The data packet encapsulated in the message will be forwarded to the correspondent’s IP address/port and permission is added for that destination. After this inbound and outbound packets are permitted between the IP addresses / ports of the correspondent and the TURN server. Only the first data packet to the correspondent is sent encapsulated in the TURN SEND message. Once the correspondent’s address and permission is established in the TURN server, rest of the communication between the client and the server takes place using regular unencaptulated UDP/RTP packets.

As with STUN the IP addresses and ports obtained with TURN are then used by other protocols for getting UDP flows across NATs and firewalls.

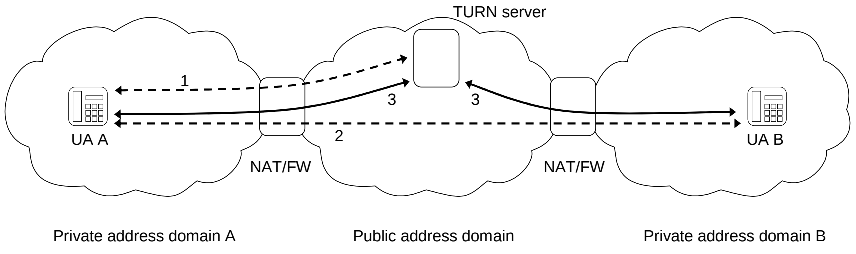

Figure 20 A user agent establishing communications using TURN

In Figure 20 above, the User Agent A (UA A) is equipped with TURN support and resides behind NAT. UA B is also located behind NAT, but in another private address domain than UA A. In order to communicate with each other UA A needs to know the public address, that UA B can use to receive packets sent by UA A and vice versa. In this example UA A uses TURN to obtain a public transport IP address and port from a TURN server (1). Then UA A uses this address and port in the signalling to tell UA B where to send its media stream intended for reception by UA A (2). The media stream (3) in the TURN case is not established directly between the user agents as with STUN, but is relayed by the TURN server.

ICE

Interactive Connectivity Establishment (ICE) is specified by an Internet-Draft [Ros03b]. It describes a methodology for NAT traversal for multimedia session signalling protocols, such as SIP:

ICE makes use of existing protocols, such as Simple Traversal of UDP Through NAT (STUN) and Traversal Using Relay NAT (TURN). ICE makes use of STUN in peer-to-peer cooperative fashion, allowing participants to discover, create and verify mutual connectivity.

ICE is a methodology to determine the best way of establishing connectivity through NAT. ICE makes use of existing NAT traversal methods, but uses them in a coordinated fashion in order to avoid many of the pitfalls of just using one of the methods alone.

STUN and TURN are used by default, but it is possible to use other methods via extensions. ICE requires that additional capabilities are implemented in the multimedia signalling protocols. This means modifying current implementations.

For those protocols which make use of the Session Description Protocol (SDP), the ICE specification defines the necessary extensions. Protocols such as SIP and MGCP fall into this category. Other protocols, like H.323 must define their own mechanisms.

The key assumption made by ICE is, that it cannot know in advance how to communicate with any peer. This means, that in the beginning nothing is assumed about the presence of NAT or the four different types of NAT between the client itself and the other client that it is connecting to. In ICE, the client initiating the session (e.g. calling party) is called the Initiator. The client receiving the session request is called the Responder.

Before establishing a sessions, the initiator obtains as many IP address / port combinations, that might be potential points of contact to receive packets from other clients, the responders. Any protocol or method that provides these points of contact can be used. These include using STUN and TURN and even a VPN. Also the local interface addresses are used. For hosts with IPv4 and IPv6 dual stack, local interface addresses from both stacks will be used. The only requirement is that, one of all the addresses has to work for any responder it might communicate with. This address is used as a last resort or used with a peer that does not support ICE. Such an address, that always works is a transport address obtained from a TURN server residing in the public Internet address space.

After gathering the addresses, the initiator runs a STUN server on each address it obtained. The potential transport addresses are advertised to the responder via ICE. The responder then sends STUN connectivity checks to all the addresses. Information is gathered from the tests and eventually one of the points of contacts selected according to local preference.

The ICE method is complex, but will result in the selection of an optimal transport path for the media. ICE addresses only the method of determining best media path. NAT traversal of signalling is not in the scope of ICE.

MIDCOM

Some SBCs use MIDCOM approach for NAT traversal in a decomposed implementation. It resembles the MIDCOM approach of access control. In the case of NAT and firewall traversal, the middleboxes are NAT devices and firewalls. The corresponding middlebox services in the case of NAT are the translation of IP addresses and ports, and in the case of firewall, the filtering and policing of traffic. The MIDCOM agent is an entity that is tightly integrated with the application, like a softswitch or a SIP proxy. The agent combines application awareness (information on the application’s requirements for NAT traversal for example) with the knowledge of the middlebox service or function. This combination enables the agent to guide the operation of the middlebox to enable the applications IP packets to traverse the middlebox.

The agents communicate with the middleboxes using middlebox communication (MIDCOM) protocol.

The protocol between a MIDCOM agent and a middlebox allows the MIDCOM agent to control how the middlebox performs its services. On the other hand the protocol allows the middlebox to offload application specific processing to the MIDCOM agent. As the agent is integrated with the application itself, it is much easier to keep the function of the agent up to date with the application, than it would be in a case when the awareness of the application would reside in the middlebox itself. In a nutshell, the MIDCOM protocol allows the middlebox to perform its operation with the aid of MIDCOM agents, without resorting to embedding application intelligence.

The main motivation behind architecting this protocol is to enable complex applications through middleboxes, seamlessly using a trusted third party, i.e., a MIDCOM agent and without changing the application logic each time new applications with new requirements emerge.

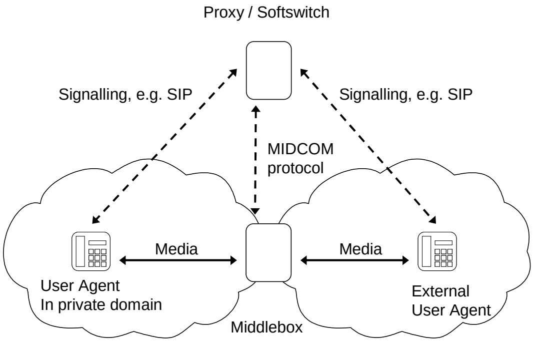

Figure 21 MIDCOM framework illustration with In-Path SIP Proxy

In Figure 21 the dashed arrow between the user agents and the proxy / softswitch both in the private domain and the external domain refer to a signalling exchange between the endpoints and the switching node. The arrow between the proxy /softswitch and the middlebox refer to MIDCOM communication. The arrows between each user agent and the middlebox represent RTP/RTCP media traffic.

The Figure 21 could represent a practical example of a case, where the application in the softswitch has detailed information, such as IP addresses and ports, on the media streams that are needed for the application to function. The MIDCOM agent residing within the application passes this address information to the middlebox (e.g. a NAT firewall) using the MIDCOM protocol. The middlebox uses the information delivered by the agent (a trusted third party) to enable the media streams required to traverse the NAT firewall.

Midcom must be considered work in progress, because as of writing this paper, the MIDCOM protocol is yet to be devised. Some protocols have been evaluated [Bar05] to be used as the MIDCOM protocol. The protocols are:

- SNMP
- RSIP
- Megaco [Cue00]
- Diameter
- COPS

Early MIDCOM implementations in SBCs [Jun05a], [New05a], [Fle05] use at least Megaco / H.248 as the MIDCOM protocol.

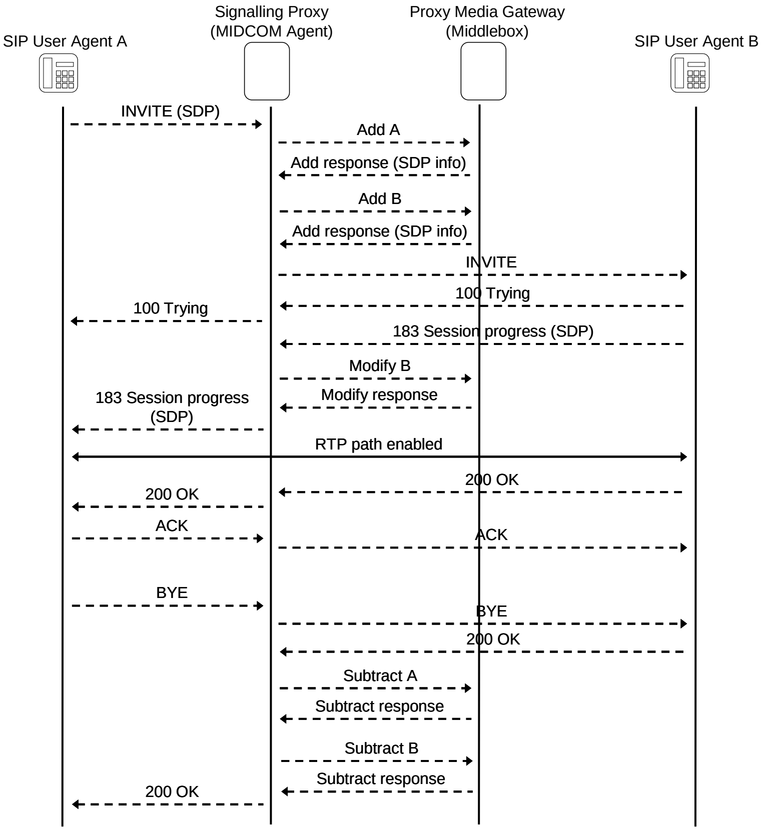

Figure 22 A Sample call flow of MIDCOM using Megaco [Fle05]

The diagram in Figure 22 shows message flows of two SIP user agents communicating via a SIP proxy. The SIP proxy is part of a signalling proxy, which also includes a MIDCOM agent. The MIDCOM agent is connected to a proxy media gateway and acts as a MIDCOM middlebox. The protocol between the agent and middlebox is Megaco. The SIP endpoints (user agents and proxy) communicate using SIP as usual. Ignoring the Megaco messages in the diagram would result in a typical SIP session setup and teardown with a proxy involved.

The MIDCOM part of the call flow describes the exchange of information enabling the media flows to traverse the middlebox under the control of the SIP proxy. If the middlebox in question is a NAT device, the Add messages create NAT mappings across the middlebox. In the case the middlebox is a firewall, firewall pinholes are opened between the two SIP user agents.

Modify messages are used to change the mappings or pinholes created with Add. This is necessary, as all the addresses and ports to be used in an established media session are not initially known, but more SDP information becomes available as the session setup progresses.

As the session is torn down, Subtract messages are used to remove the mappings or pinholes created during session setup.

5.2.3.5 Non-Standard Approaches to Firewall and NAT Traversal

The non-standard approaches to NAT traversal in SBCs is based on modifying the signalling in the SBC application. A SBC operates by taking the initial IP addresses and ports provided by the client application for signalling and media, and modifying them before delivering them to the other clients. While doing this, the SBC application also creates internal mappings for media streams to match the modified signalling. As a result, the media streams between clients are forced to flow through the SBC [Ros05].

This method is very useful for NAT and firewall traversal. In order to communicate with the rest of the world, the clients behind NAT and firewall devices only need to be able to establish connections with the SBC. The rest of the world can reach the SBC directly, as it has at least one public IP address.

This kind of operation is possible, when certain assumptions about the behaviour hold:

- The client sends and receives its media traffic from the same IP address and port
- When using UDP based signalling, the client sends and receives its signalling traffic from the same IP address and port

SIP signalling, for example, can take place over TCP or UDP. For TCP, the port for sending and receiving traffic is always the same. It follows form the connection oriented nature of TCP and is specified in the TCP RFC. For the connectionless nature of UDP, such a requirement on using the same port for sending and receiving data does not exist and is not required by the SIP specification [Ros02] either. However, using the UDP ports in a symmetric way is common practice [Ros05].

This being common practice makes the proprietary NAT traversal method very effective, because no explicit support for NAT traversal is required in the client or other existing network element, like NAT routers, firewalls.

Non-standard NAT traversal was examined using a test setup with a user agent located behind a NAT firewall. The user agent connects to an IP telephony service provider operating in the Internet. NAT and firewall traversal is implemented with a SBC. This setup is described in detail in Appendix B.

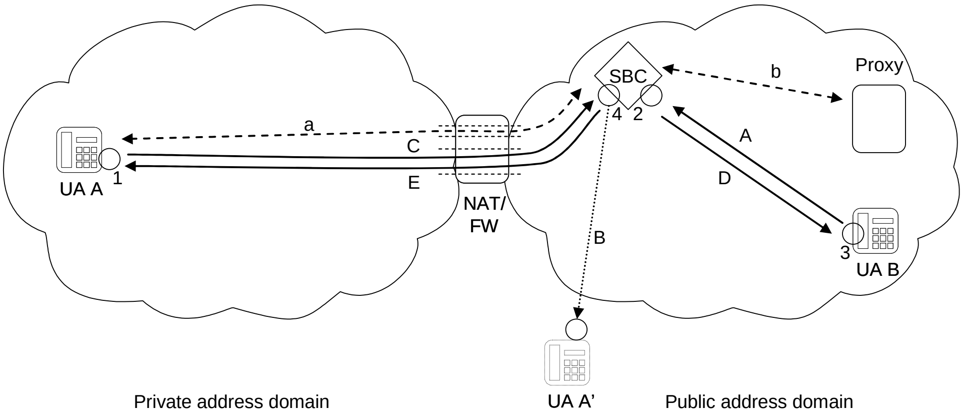

Figure 23 Proprietary SBC NAT traversal scenario

Figure 23 describes a NAT traversal scenario with SIP, where one user agent is located in a private network and the other in the public Internet. The NAT traversal function is implemented using a non-standard method by a SBC located in the public Internet address domain. The Figure 23 shows by numbers 1-4 the chronological order in which the media transport addresses for receiving media by the network elements become available via SDP. The order in which the actual media streams start flowing is marked in the diagram using letters A-E. Letters a-b marks the SIP signalling.

The signalling takes place using SIP over UDP, but in a symmetrical manner, i.e. source and destination port is the default SIP port 5060 in all network nodes. This symmetrical operation enables the signalling to traverse through the stateful NAT firewall. Firewall state is kept open with periodic re-registrations. This avoids closing of the pinhole when the user agent is idle for a long time.

Below in Figure 24 is an extract of a call flow generated from network traffic captures of the test setup.

```
    UA A                    NAT Firewall                                SBC                                     Proxy
1.  |>F13 INVITE (sdp)-------------------------------------------------->|                                        |
    |                           |>F14 INVITE (sdp)---------------------->|                                        |
    |                           |                                     2. |>F15 INVITE (sdp)---------------------->|
    |                           |                                     3. |< Proxy Authentication Required 407 F16<|
    |                           |< Proxy Authentication Required 407 F17<|                                        |
    |<---------------------------- Proxy Authentication Required 407 F18<|                                        |
    |>F19 ACK ---------------------------------------------------------->|                                        |
    |                           |>F20 ACK ------------------------------>|                                        |
    |>F21 INVITE (sdp)-------------------------------------------------->|                                        |
    |                           |>F22 INVITE (sdp)---------------------->|                                        |
    |                           |                                        |>F23 ACK ------------------------------>|
    |                           |                                        |>F24 INVITE (sdp)---------------------->|
    |                           |                                     4. |<- your call is important to us 100 F25<|
    |                           |                                        |<---------------------- Ringing 180 F26<|
    |                           |<- your call is important to us 100 F27<|                                        |
    |<------------------- trying -- your call is important to us 100 F28<|                                        |
    |                           |<---------------------- Ringing 180 F29<|                                        |
    |<-------------------------------------------------- Ringing 180 F30<|                                        |
    |                           |<---------------------- Ringing 180 F31<|                                        |
    |<-------------------------------------------------- Ringing 180 F32<|                                        |
    |                           |<---------------------- Ringing 180 F33<|                                        |
    |<-------------------------------------------------- Ringing 180 F34<|                                        |
    |                           |<---------------------- Ringing 180 F35<|                                        |
    |<-------------------------------------------------- Ringing 180 F36<|                                        |
    |>F37 (sip incomplete) >>>------------------------------------------>|                                        |
    |                           |>F38 (sip incomplete) >>>-------------->|                                    5.  |
    |                           |                                        |<----------------------(sdp) OK 200 F39<|
    |                        7. |<----------------------(sdp) OK 200 F40<| 6.                                     |
8.  |<--------------------------------------------------(sdp) OK 200 F41<|                                        |
    |>F42 ACK ---------------------------------------------------------->|                                        |
    |                           |>F43 ACK ------------------------------>|                                        |
    |                           |                                        |>F44 ACK ------------------------------>|
```

Figure 24 Call flow of a proprietary SBC NAT traversal scenario

The actual network traces that this call flow is based on are not presented in this document as they contain a lot of data. Full traces and call flows are available from the author by request.

The following list describes how the proprietary NAT traversal takes place. The numbers at the beginning of each paragraph refer to Figure 24.

1. The UA A sends a SIP invite F13 to the SBC. This invite contains SDP information with address and port, A1:P1, describing where UA A wants to receive media. This is marked by (1) in Figure 23. After receiving F13, the SBC knows the transport address, that is used by UA A in the private address domain to receive media.

2. The SBC sends an invite with F15 to the proxy. This invite is based on information from F13, but has been modified by the SBC. The modification essential for NAT traversal is in the SDP. Instead of containing the transport address A1:P1 belonging to UA A, the SDP has a transport address A2:P2 that belongs to the SBC, marked by (2) in Figure 23.

3. The proxy responds with Proxy Authentication Required 407 F16. This happens because authentication is required and has not taken place yet. The message exchange from F16 to F21 takes care of the authentication and finally the SBC sends invite F24, which is similar, to F15 but contains valid credentials and has identical SDP with F15.

4. Messages from F25 to F36 contain progress information originally sent by the proxy and then relayed by the SBC through the NAT firewall to the user agent. In addition to just relaying, the SBC repeats the Ringing 180 message several times while waiting for the OK 200 F39 from the proxy.

5. The OK 200 F39 message from the proxy is a response to F24 and contains SDP describing the address and port A3:P3 that belongs to the UA B, marked by (3) in Figure 23. This is the address used by UA B to receive media. After receiving F39 the SBC knows where to send media intended for reception by UA B. Virtually at the same time as the SBC receives F39 from the proxy, it starts receiving a media stream to (2) form UA B. The stream is marked with (A) in Figure 23.

6. After receiving F39 the SBC sends OK 200 F40 to the UA A. This message is based on information from F39, but has been modified by the SBC in a similar way, as earlier with F13 and F14 travelling in the opposite direction. The F40 SDP contains address and port A4:P4 belonging to the SBC instead of the original A3:P3 in F39. A4:P4 is marked in Figure 23 by (4). Instantly after sending F40 the SBC starts relaying the media stream of UA B. The destination of this stream is the original private A1:P1 that contains the private address of UA A. As the SBC resides in the public address domain and sends traffic with a non-routable private destination address, the media stream does not reach the correct recipient. This phase is presented in Figure 23 by a pseudo user agent UA A’ and the media stream (B). This user agent does not really exist in the network.

7. The reason for the SBC using a wrong destination address is that the correct mapping between the private address and port of UA A (A1:P1) and its public presentation after the NAT firewall is not known by the SBC at this point of time.

8. When the UA A receives OK 200 F41 sent by the SBC through the NAT firewall, it responds with ACK F42 and starts sending media to the SBC as described in SDP of F41. This SDP contains the address A4:P4. This is media stream (C) in Figure 23.

This concludes the exchange of SDP information. The SBC is currently sending a media stream to the pseudo user agent UA A’ using A4:P4 as the source and the wrong, private A1:P1 as the destination. When the packets of the media stream (C) through the NAT firewall reach the SBC, they have the public presentation of the private address A1:P1 as the source address. From this information the SBC learns the mapping between public and private presentations of the media stream. The SBC then modifies the stream (B) destination from private A1:P1 to the public presentation of A1:P1. This is the stream (E) in Figure 23.

The stream (D) from the SBC is a relayed version of the stream (C), and is generated as soon as the packets of (C) are received by the SBC.

At this point there is bi-directional media flowing between UA A and UA B. The Media stream traverses the NAT firewall and is being relayed by the SBC.

In addition to NAT traversal of the media streams SBCs can perform NAT traversal for UDP SIP signalling in situations where a NAT is present between a user agent and the registrar of the domain. NAT bindings and firewall pinholes are typically valid for relatively short periods if the connection is idle i.e. there are no UDP datagrams travelling across the NAT of firewall between an IP address/UDP port pair. SBCs can be used to keep the binding alive by forcing the sending of REGISTER messages with a period shorter than the expiry time of the NAT or firewall.

When the registrar receives a REGISTER request from the user agent and responds with a 200 (OK) response, the SBC modifies the response decreasing the validity time of the registration so that the registration expires sooner. This forces the user agent to send a new REGISTER to refresh the registration sooner that it would have done on receiving the original response from the registrar. The REGISTER requests sent by the user agent refresh the binding of the NAT before the binding expires [Cam05a].

Although the proprietary method of NAT traversal enables operation through NAT it requires the SBC to behave like a B2BUA and has some adverse effects. These will be discussed later in this document.

5.2.4 Traffic Monitoring

Operators and service providers are usually interested in the properties of network traffic related to the services and applications they offer to the customers. In order to gain detailed information a SBC can be used for traffic monitoring of IP multimedia services. This is achieved either by having a SBC perform the monitoring function itself when signalling and media is routed via the SBC, or configuring the SBC in a way that enables a third party to perform the task. The latter can be done by having a SBC force the required flows through a network element performing traffic monitoring. [Cam05]

When both signalling and media are routed via a SBC, it is possible to associate each media flow with signalling. This enables collecting detailed information per session basis. Information about the state of the network can be obtained by monitoring the RTP and RTCP flows. QoS parameters, like packet loss, jitter and delay, can be measured [Acm05] for each session or call. SBCs can also generate call detail records of the sessions. CDR data is used e.g. for billing and capacity planning.

The data collected can be used for various purposes, including network capacity planning and network management. Information needed for SLA assurance with peering operators and end users can be obtained [Acm05].

5.2.4.1 Traffic Monitoring Approaches in SBCs

The SBC product literature and vendor whitepapers publicly available do not include many details on how, or according to which standards the monitoring function is implemented. Some vendors [Sno05] mention SNMP as a method for accessing the monitoring data.

5.2.5 Traffic Shaping

In addition to monitoring many operators and service providers may want to control and shape the traffic according to traffic agreements and the capacity of the network. [Cam05].

As with monitoring, when a SBC is in the path of both signalling and media, it is possible to identify and associate each media flow with the signalling controlling it. It can be verified, that the actual media stream is what has been indicated in signalling. This is done by looking at the media coding information in the signalling, and comparing it with the properties of the associated media stream.

If differences are observed, it is possible to terminate the session by stopping relaying the media stream and sending termination signals to the parties of the conversation. This can help to maintain good QoS, by blocking denial of service attacks attempting to flood the network by sending a much higher bandwidth media stream that was agreed by the parties with signalling. Another reason for forcefully terminating a session is running out of credits, while using a pre-paid service [Sno05].

5.2.5.1 Traffic Shaping Approaches in SBCs

The SBC product literature and vendor whitepapers publicly available do not mention any standards related to the traffic shaping implementation or behaviour. The internal implementations are not disclosed either.

5.2.6 QoS Marking

IP multimedia has real time requirements and therefore it is beneficial for overall service quality to use QoS mechanisms, like different traffic classes and priorisation of different kinds of traffic, while transported in the IP network.

As with monitoring and shaping, when a SBC is in the path of both signalling and media, it is possible to identify and associate each media flow with the signalling. This enables the marking of signalling and media packets sent by the SBC with QoS information [Cam05].

The QoS marking functionality can be used to implement and enforce the QoS policy for an IP multimedia service. Explicit marking can be used. This means that the original QoS markings of incoming traffic are not trusted, and that the signalling and media flows are explicitly marked as required by the transport network. This enables intended treatment of different traffic types, like voice, video and instant messaging.

5.2.6.1 QoS Marking Approaches in SBCs

The SBC vendor literature mentions compatibility for QoS marking with the following methods: DiffServ, MPLS, RSVP, IEEE 802.1p. The most commonly supported method is DiffServ. Below are brief descriptions of each one.

DiffServ

DiffServ uses DS bits in the IP packet header to recognize the need for QoS on a particular packet-by-packet basis. DiffServ, as defined in several RFCs [Nic98], [Bla98], uses the Type of Service (TOS) field within the IP header to mark and prioritize traffic. DiffServ defines a common understanding about the use and interpretation of this field.

MPLS

The MPLS QoS marking support by SBC means MPLS support for DiffServ [Fau02]. This RFC specifies methods for transporting DiffServ QoS information in MPLS header EXP bits (E-LSP) or alternatively using labels to indicate traffic classes (L-LSP).

RSVP

The RSVP [Bra97] protocol is used by a host to request specific qualities of service from the network for particular application data streams or flows. RSVP is also used by routers to deliver quality-of-service (QoS) requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service.

IEEE 802.1p

The IEEE standard 802.1p [IEE98] specifies a priority scheme for the layer 2 switching in a switched LAN. It adds 16 bits to the Layer 2 header, including three bits that can be used to classify priority. DiffServ can be mapped to IEEE 802.1p when packets flow from a layer 3 network into a network, such as a LAN, where switching takes place on layer 2.

No non-standard methods for QoS marking where encountered in SBC literature. This is not surprising considering that if non-standard methods were used for QoS marking, it would limit the usefulness of the QoS information to the network elements implementing the proprietary scheme.

5.2.7 Signalling Interoperation and Protocol Repair

Protocol details in IP multimedia platforms vary from one implementation to another. Vendors have implemented protocols like SIP and H.323 in various ways for different reasons, such as to gain efficiencies or advantages over competitors. In addition, it is possible to interpret standard protocol specifications differently or implement a standard’s features before they have been officially included in a standard. [Com04].

Protocol repair means dealing with protocol messages generated by not-fully-standard clients in a graceful way Also, new versions of protocols become available in network elements, and can result in an environment with multiple potentially incompatible versions coexisting simultaneously [Cam05].

The variance in implementations leads to interoperability problems even between systems using the same protocol. Calls may be rejected or other unexpected behaviour may occur, because the systems use different interpretations or versions of a standard.

SBCs can be used to create an additional abstraction layer by normalizing vendor specific protocol implementations to one selected version. This abstraction can be very appealing to carriers wanting to deploy new technologies at the edge and access, while changing and developing the interface towards the core network at a slower pace. By creating this kind of an abstraction layer, peering and interconnection with other carriers or service providers becomes less complex, as the internal interface can remain unchanged and interoperability issues with other parties are dealt with on the other side of the layer [Rod04].

5.2.7.1 Signalling Interoperation and Protocol Repair Approaches in SBCs

The publicly available literature studied des not mention any standards related to IP multimedia signalling interoperation and protocol repair.

The implementations of protocol repair found in SBCs vary. One approach is to implement the signalling processing as a proxy (This is not a reference to the SIP Proxy nomenclature) that is liberal in what it receives and strict in what it sends [Cam05]. At the other end of the spectrum are implementations, which fully terminate and regenerate signalling [Jun05b]. The mode of operation, where signalling is first terminated on one side and regenerated on the other can be modelled by an entity consisting of two endpoints connected back-to-back e.g. SIP B2BUA.

5.2.8 Protocol Interworking

Protocol interworking means providing functionality that enables two different systems using different signalling protocols to work together with the aid of an interworking function (IWF).

Protocol interworking is needed for example in peering between operators using different protocols in their networks. Both H.323 and SIP implementations are present in the networks of long distance and international carriers, who utilize VoIP to carry PSTN and PLMN traffic. In order to exchange traffic between carriers using different signalling protocols, conversion is required.

The need for protocol interworking is not limited to carrier to carrier peering. SIP is frequently used in systems providing communication services for consumers and enterprise. On the other hand, H.323 based video conferencing is very common and widely deployed in the enterprise environment. Protocol interworking is needed to connect the two. Figure 25 presents architecture for implementing IWF.

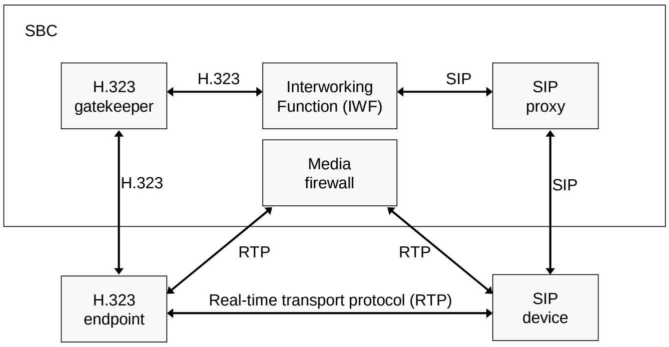

Figure 25 Architecture of a SBC with SIP-H.323 IWF [Com04]

In addition to providing interworking between two completely different protocols such as SIP and H.323, protocol interworking may be required even between two systems using the same signalling protocol as the basis of their operation.

This is the case in connecting the SIP services found in the Internet with the SIP based IMS services specified by 3GPP and ETSI TISPAN. This incompatibility within the same protocol is the result of different SIP profiles used by the two [Cam05]. The IMS specification defines profiles of IETF RFCs for 3GPP usage [3GP05].

SBCs are used to provide interoperation between different signalling protocols such as SIP and H.323 [Com04]. The SIP-H.323 interworking has existed since the first SBCs.

5.2.8.1 Protocol Interworking Approaches in SBCs

The interworking implementations in SBC are similar to the SIP B2BUA approach found in protocol repair function, but in this case the other endpoint may be other than a SIP user agent, such as H.323 gateway. The signalling is terminated by the IWF and then regenerated using another signalling protocol or protocol profile. Converting between signalling protocols to achieve interworking is a complex process and interworking functions require quite a lot of processing resources.

5.2.8.2 SIP-H.323 Interworking

The requirements for SIP-H.323 IWF have been defined by the IETF in RFC 4123 [Sch05]. This RFC was published in July 2005, but the SBC literature studied does not specify conformance to this specification or any of the draft versions [Agr01] through [Sch04].

The RFC 4123 states that a SIP-H.323 IWF contains functions from the following list among others:

1. Mapping of the call setup and teardown sequences
2. Registering H.323 and SIP endpoints with SIP registrars and H.323 gatekeepers
3. Resolving H.323 and SIP addresses
4. Maintaining the H.323 and SIP state machines
5. Negotiating terminal capabilities
6. Opening and closing media channels
7. Mapping media-coding algorithms for H.323 and SIP networks
8. Reserving and releasing call-related resources
9. Processing of mid-call signalling messages
10. Handling of services and features

The functions seem fair and practical and it is very possible, that many SBC IWF implementations perform them.

The requirement specification states that IWF should not process media and assumes media exchange to take place directly between endpoints. If a particular service requires, the IWF is allowed to handle media. The IWF then simply forwards media packets without modification from one network to the other.

SBCs frequently participate in media processing to perform other functions such as NAT traversal or topology hiding. It is allowed although not encouraged by the specification.

Figure 26 shows common IWF configurations or deployment scenarios listed by the specification. They are similar to the SBC deployment scenarios. The first four ones describe scenarios where endpoints using different signalling protocols communicate. The last two ones are examples of a situation, where a network used as a transit network uses a different protocol than the access networks.

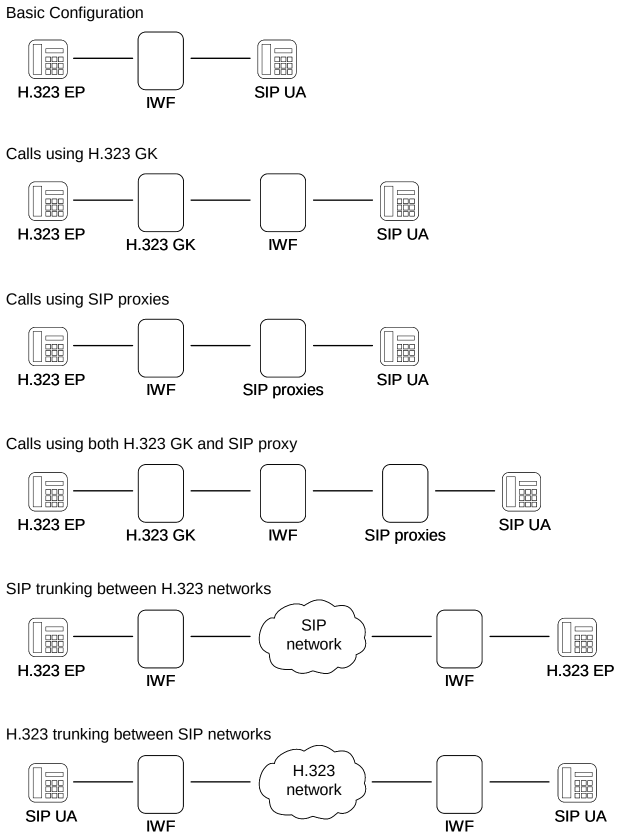

Figure 26 Common SIP-H.323 interworking scenarios [Sch05]

While the basic functionality described in SBC literature matches that of the IWF requirements no compliance to RFC 4123 is mentioned. The IWF requirement specification covers the following main areas (sub categories not presented here):

- Pre-Call requirements
- General interworking requirements
- Transport
- Mapping between SIP and H.323
- Security considerations

The material available from vendors does not describe details of the implementations. As a result, it is not possible to analyze how the requirements in any of the areas handled are met by SBC implementations.

5.2.8.3 Generic SIP and IMS Interworking

As of writing this thesis, there is emerging support for TISPAN based IMS services by some SBCs [New05b], [Acm05a]. The standardization of extended IMS architecture by ETSI TISPAN is work in progress.

TISPAN NGN Functional Architecture Release 1 [ETS05] specifies an Interworking Function (IWF). The IWF performs the interworking between protocols used within TISPAN NGN service control subsystems and other IP-based protocols e.g. between the SIP profile used in the IMS and other SIP profiles or IP-based protocols such as the H.323 protocol.

The IWF has two interfaces between which it performs it function. For protocols used in TISPAN, it interfaces to another TISPAN defined functional element called The Interconnection Border Control Function (IBCF) shown in Figure 27. For non TISPAN protocols, such as SIP version not compatible with SIP in IMS the IWF interfaces to the functional elements in other IP networks [ETS05]. IMS is not in the central focus of this thesis, but SBCs may be used to implement IWF and IBCF. These functions are marked by the dashed line in Figure 27.

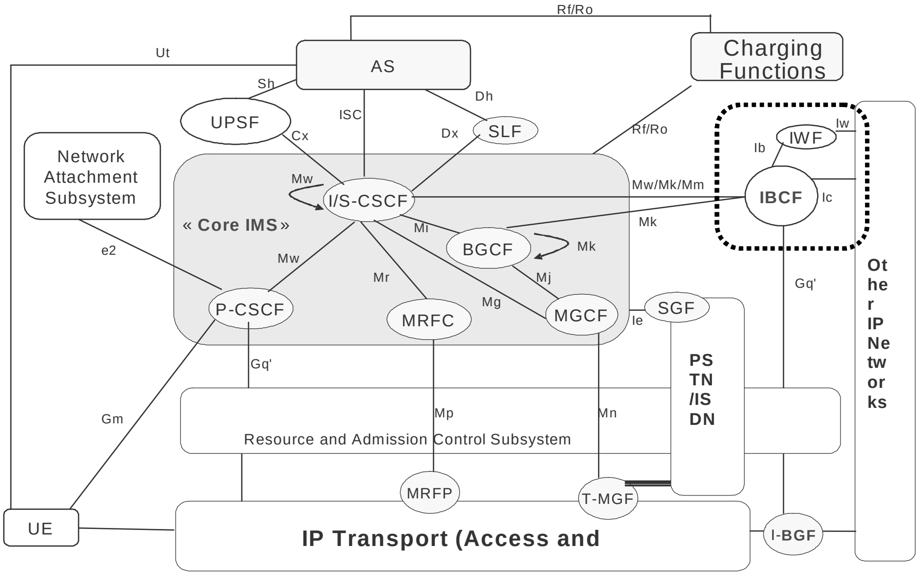

Figure 27 TISPAN NGN Architecture

The TISPAN specifications do not yet define any internal functions or requirements of the IWF, thus no comparison between it and the functions found in SBCs can be made.

5.2.9 IPv4/IPv6 Interworking

IPv4/IPv6 interworking means providing functionality that enables endpoints located in networks using different versions (IPv4 and IPv6) of the Internet Protocol to communicate. Direct communication between an endpoint using IPv4 and another using IPv6 is not possible, as the protocol versions are not compatible. There is expected to be a long transition period during which it will be necessary for IPv4 and IPv6 nodes to coexist and communicate [Tsi00]. A set of IPv4-to-IPv6 transition and coexistence mechanisms will be required during this transition period.

IPv6 networks are emerging in enterprise, public sector and 3G mobile networks. 3GPP IMS network is completely based on IPv6. The TISPAN IMS [ETS05] however relaxes the constraints of 3GPP IMS [3GP05] on the sole use of IPv6, and defines a functional element called Border Gateway Function (BGF). One of the functions of BGF is interworking between IPv4 and IPv6 networks.

5.2.9.1 IPv4/IPv6 Interworking Approaches in SBCs

SBCs can perform conversions between IPv4/IPv6 versions [Cam05]. The conversion is based on modifying addresses in IP packet headers and inside signalling messages. For SIP this means modifying the addresses in SIP headers and SIP message bodies carrying SDP in a similar way as is done to achieve the NAT traversal functionality described earlier. Two different methods for interworking were found in the SBC literature [Gla05].

5.2.9.2 SIP ALG at Network Boundary

A SBC located at the boundary of IPv4 and IPv6 networks operating as an ALG can be used to implement IPv4/IPv6 interworking for SIP. The SBC functions once again as a B2BUA and has one side connected to IPv4 and the other to IPv6. Requests from one side are received, then reformulated and sent out as a new request. Similar operation is performed in the other direction.

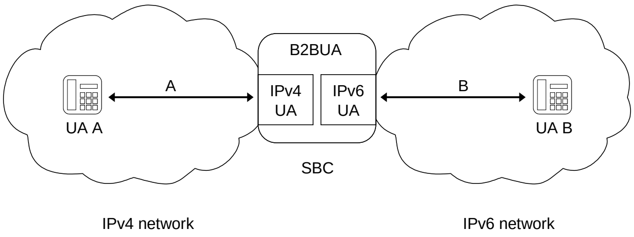

Figure 28 ALG with B2BUA at network boundary

In Figure 28 all traffic (A) between UA A and the IPv4 UA inside SBC takes place natively using IPv4. Similarly all traffic between UA B and the IPv6 UA on IPv6 side takes place using IPv6. The B2BUA application performs the required conversions of addresses in IP packet headers, SIP headers and SDP in SIP message bodies.

While the B2BUA concept is defined in SIP, the operation of modifying the actual addresses in numerous SIP header fields an in the message bodies carrying SDP, is not standardized and thus may vary from one implementation to the other.

5.2.9.3 NAT-PT and Centralized SIP ALG

One standard method for providing connectivity between IPv4 and IPv6 networks is Network Address Translation - Protocol Translation (NAT-PT) [Tsi00]. The NAT part of NAT-PT is very similar to the IPv4 NAT described earlier, but is not identical. IPv4 NAT translates one IPv4 address into another IPv4 address. In NAT-PT one IPv4 address is translated into another IPv6 address and vice versa. The PT part of NAT-PT refers to the translation of an IPv4 packet into a semantically equivalent IPv6 packet and vice versa.

As with ordinary IPv4 NAT, protocols such as SIP that carry address information in the protocol messages generally fail to operate when there is a NAT-PT device in the path between two endpoints. Due to this IPv4-IPv6 interworking with NAT-PT is faced with NAT traversal issues, as pure IPv4 NAT. In order to cope with these NAT unfriendly protocols NAT-PT allows the use of application specific ALGs in NAT-PT devices. The downside of a practical NAT-PT ALG approach is that a working solution e.g. for SIP requires, that all NAT-PT devices that happen to be in the path have SIP ALG functionality implemented, activated and properly configured.

The following approach with SBCs takes advantage of standard NAT-PT functionality without requiring SIP ALG support from NAT-PT devices. The SBC uses identical NAT traversal methods as in pure IPv4 NAT traversal scenario to solve the issues created by

NAT-PT without ALG with the exception, that the SBC needs to handle IPv6 addresses in addition to IPv4 addresses.

Figure 29 has two scenarios. First scenario has an IPv6 user agent (UA 6) communicating with UA A located in IPv4 network. The second scenario has an IPv4 user agent (UA 4) located in an IPv4 network communicating with another IPv4 user agent in IPv4 network, but via access network that is IPv6.

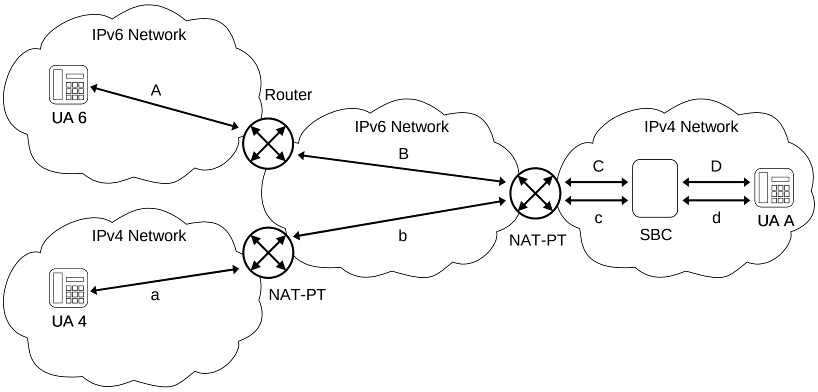

Figure 29 Standard NAT-PT and centralized SIP ALG

The operation of this solution is based on IP packet header translations performed by NAT-PT devices and the SBC fixing the problems created by the use of NAT.

The tables 4 and 5 summarize how address information is presented in the networks in Figure 29. The addresses of media stream packets are not included in the tables. They match the presented IP header address – the native addressing scheme used in each of the networks.

	A	**B**	**C**	**D**
IP Header	IPv6	IPv6	IPv4	IPv4
SIP Header	IPv6	IPv6	IPv6	IPv4
SDP in SIP	IPv6	IPv6	IPv6	IPv4

Table 4 Addresses in UA 6 – UA A communication scenario

	a	b	c	d
IP Header	IPv4	IPv6	IPv4	IPv4
SIP Header	IPv4	IPv4	IPv4	IPv4
SDP in SIP	IPv4	IPv4	IPv4	IPv4

Table 5 Addresses in UA 4 – UA A communication scenario

The SBC uses the non-standard NAT traversal method, described in section 6.2.3.5, to handle the media streams that are flowing through the NAT-PT devices. In fact the UA 4 – UA A communication scenario is identical to a pure IPv4 scenario with two IPv4 NAT devices on the path. No IPv6 addresses are handled by any of the SIP entities, as can be observed from Table 5. IPv6 addresses are present only in IP headers in IPv6 network. Figure 30 shows this equivalent IPv4 scenario.

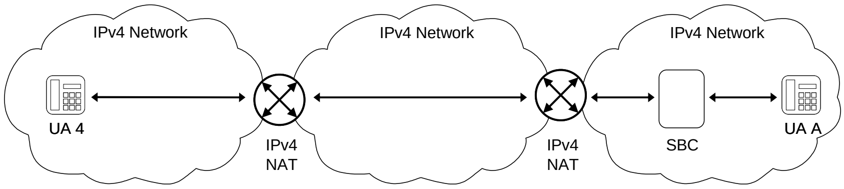

Figure 30 Two IPv4 NATs in path

5.2.10 Transport Protocol Interworking

Transport protocol interworking means using a different transport layer protocols with different endpoint. The current SIP RFC [Ros02] requires TCP and UDP support for all SIP elements but the former SIP RFC [Han99] only requires UDP. All SBCs in the scope of this study support SIP as in RFC 3261 and thus can perform transport protocol interworking between older and newer endpoints. MGCP uses only UDP and H.323 TCP.

In addition to TCP and UDP, transport layer interworking can take place between unencrypted and TLS [Die99] encrypted transport layers. Some SBCs support TLS for SIP and for H.323 signalling [ITU03].

Transport protocol interworking is standards based functionality found in SIP servers and H.323 gatekeepers, thus SBCs performing it can be modelled as one of those entities.

5.2.11 DoS and Overload Prevention

SBC in the signalling path can perform validation of signalling messages. Signalling message structures may be inspected and only legitimate messages forwarded [Jun05]. This helps to block attempts to exploit security flaws in network elements by the use of malformed, or exceedingly large signalling messages.

Signalling rate limiting can be used to protect gateways, servers and other devices from DoS attacks or error conditions in endpoints causing packet flooding towards servers. Call gapping protects the softswitch, gatekeeper, SIP proxy, or other signalling entities from excessive signalling requests. This helps to prevent attacks that attempt to heavily load the call control entities with multiple concurrent requests. Protecting against excessive signalling attacks is achieved by dropping signalling packets at a specific threshold.

Limiting criteria encountered in literature were: Number of simultaneous requests directed at a specific destination, threshold limit per source IP, accumulated threshold limit, summing the entire signalling attempts rate and threshold per-interface or logical location.

No standards support by SBCs was identified for DoS and overload prevention functions. The implementations vary from one vendor to the other.

5.2.12 Call Admission Control

Admission control can be defined as the process of deciding whether a newly arriving request for service from a network element can be granted or not [She97]. SBCs may be used to determine this for IP multimedia sessions they are controlling. It can be determined if a call should be admitted to a particular network [Acm02]. The decision is based on the availability of network resources such as real-time bandwidth, and the requirements of a newly arriving request.

5.2.12.1 Call Admission Control Approaches

SBCs can keep track of the current network utilization by observing the bandwidth requirements of codecs used by already established sessions. Comparing this information to configured network topology with capacities it is possible to start refusing new sessions that would exceed the capacity of a particular network or access link. This kind of control of network utilization can help to avoid exceeding the real time capabilities of networks and avoid degrading the perceived quality and QoS metrics of all existing sessions. Sessions that can not be admitted to a network due to insufficient network capacity are gracefully rejected by the SBC. This is done using signalling e.g. in SIP by sending an INVITE with “503 Service Unavailable”. In case the rejected session is a phone call, the end user SIP device may translate “503 Service Unavailable” into a fast busy audio signal.

No standards support by SBCs was identified for call admission control (CAC) functions. The implementations vary from one vendor to the other.

5.2.13 Legal Intercept

Legal intercept means enabling the authorities or government agencies to perform “wiretapping” of conversations for authorized legal purposes. In the EU these regulations are national. The requirement for Legal intercept exists for the PSTN and PLMN, but at the time of writing no final decision exists for how to apply these requirements to IP networks [Kar05]. In The United States the Federal Communications Commission (FCC) requires certain broadband and VoIP Providers to accommodate wiretaps based on Communications Assistance for Law Enforcement Act (CALEA) [FCC05].

5.2.13.1 Legal Intercept Approaches

SBCs in the media path can be used to implement Legal intercept by creating a copy of the media streams that the SBC manages.

Some SBCs supporting legal intercept have standard interfaces [Jas05] to legacy CALEA/ETSI infrastructure. No standards based legal intercept for IP networks, such as described in RFC 3924 [Bak04] was encountered in SBCs.

5.2.14 Emergency Traffic

Telecommunications operators providing publicly available telephone service are required to make it possible for the user of the service to dial the general emergency numbers such as in the EU 112 and 911 in The United States. This is required by the authorities in most of the countries of the world. At least in Finland [Vie03] and The

United States [FCC05a] this requirement also applies to IP based communication services that are interconnected to the PSTN/PLMN. Calls from each geographic area to the general emergency number (112/911) must be routed to the emergency response centre specified for the relevant area.

5.2.14.1 Emergency Traffic Routing Approaches

SBCs can be used to route emergency calls based on the originating number or IP address. This helps in being able to route calls to the emergency centres serving specified geographical areas.

SBC literature mentions support for the U.S. FCC Enhanced 911 Service (E911) [FCC05b]

5.2.15 Media Encryption

SBCs can be used to perform media encryption and decryption for media stream legs terminating to or originating from the SBC. It may be desired to encrypt traffic travelling in potentially insecure networks.

5.2.15.1 Media Encryption Approaches

For media encryption the SBC operates as a B2BUA and a media relay terminating the RTP/SRTP media stream on one side of the B2BUA and regenerating in on the other side.

The SBC literature mentions standard SRTP [Bau04] as the encryption method.

5.2.16 Media Transcoding

SBCs can be used to perform media transcoding between endpoints using different codecs for their media. Transcoding is needed if a common codec between two endpoints is not available. The IMS specifies the use AMR codec for voice communications. This codec is rarely found in user agents outside the 3G mobile networks and voice communication between IMS and typical IP telephony services would require transcoding AMR to for example G.729 and vice versa.

5.2.16.1 Media Encryption Approaches

For transcoding the SBC operates as a B2BUA and a media relay terminating the media stream on one side of the B2BUA, performing the transcoding function and regenerating a new media stream on the other side.

Even thought the transcoding performed by SBCs is done between standard codecs such as G.711 and G.729, the actual behaviour B2BUA is vendor specific and implementation dependant.

5.3 SBC Functions and Standards from Different Organizations

This section and the subsections deal with the secondary goal of this thesis, which is to compare the relationship and approaches that the different standardization organizations specifying IP multimedia communication infrastructures have towards the kind of functionality performed by session border controllers.

SBCs are typically found in IP multimedia systems using SIP. As described earlier, some SBCs implement an interworking function with H.323 and some provide firewall and NAT traversal for MGCP as well. As this study focuses on SIP, the relationship of the SBC to H.323 and MGCP standards is not covered here.

The following sub-sections study the relationship between SBC functionality and SIP based communication infrastructures as defined by the IETF and ETSI TISPAN NGN Release 1 specifications. IETF and TISPAN approaches were selected because SBC literature references these infrastructures and on the other hand material from both the IETF and TISPAN has references to SBC functionality.

5.3.1 SBC Functions and IETF Standards

The concept of SBC is not defined in any IETF standards. SBC vendors have their own definitions, but none of them are universally accepted. Internet drafts discussing SBCs have been published. The SBC is not a logical IETF specified SIP entity, but a SBC or at least individual functions of a SBC can be modelled as logical SIP entities.

RFC 3261 describes the following set of SIP entities:

- SIP Servers
    - Registrar
    - Proxy Server
    - Call Stateful Proxy
    - Transaction Stateful Proxy
    - Stateless Proxy
- User Agent Server (UAS)
- User Agent Client (UAC)
- User Agent (An entity which can simultaneously act as a UAC and UAS)
- Back-To-Back User Agent (An entity composed of the UAS and UAC which acts as a UAC to determine how to answer an incoming request on the UAS).

Depending on implementation, an SBC could be a SIP Proxy or a SIP Back-To-Back User Agent. RFC 3261 prohibits a SIP proxy from modifying the SDP information in SIP message bodies. Many SBC functions, as shown in the previous sections modify the addresses, ports and even codec descriptions in SDP. Due to this the SBC must be classified as a B2BUA in nearly all cases. Classifying a SBC, that modifies SDP as a SIP proxy violates RFC 3261.

The B2BUA is a standard SIP entity, but the internal functions of a B2BUA are not defined in a generic way. A B2BUA conforming to RFC 3261 can perform any task or function internally as long as the two individual user agents that a B2BUA is composed of conform to SIP specifications. Figure 31 below shows a B2BUA. Requests arriving via UAS are processed and responses are sent out via UAC.

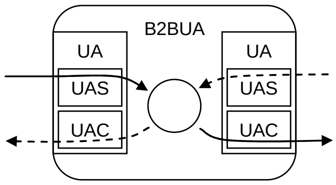

Figure 31 B2BUA

The impact of SBCs on SIP messages range from functionality that can be classified as a conforming SIP proxy to a B2BUA completely rewriting SIP headers and SDP. The impact of processing can be on either end of the spectrum depending on implementation and configuration. SBC in the middle of the path between user agents usually tries to be as transparent as possible, but the transparency of these B2BUAs varies depending on the functions they perform. A SBC might simply update the SDP and insert a Record-Route header. However, SBC might as well remove a header like Record-Route, or replace the Contact with the SBCs address. A new Call-ID, or even new tags might be created in the From and To headers. This kind of unknown behaviour by an intermediate entity can produce unexpected and unintended results in applications that expect standard behaviour and end-to-end connectivity. There are fears that Internet application developers might be forced to take into account the impact of non-conforming intermediaries making application development more difficult.

5.3.1.1 SIP Unfriendly Functions

The IETF considers many of the functions performed by SBCs SIP unfriendly [Cam05a]. All functions where the SBC acts as a B2BUA and inserts itself into the media path by modifying the SDP are considered SIP unfriendly. Stateful operation required for some functions is considered unfriendly. Also manipulating the SIP header fields and parameter values in SIP headers in ways not allowed for SIP servers in RFC 3261 is considered unfriendly.

SIP unfriendly SBC functions of the ones described earlier in this study are summarized in Table 6.

Topology hiding	• Modifies the SIP headers and forces the media through the SBC by modifying SDP • Utilizes B2BUA and media relay • Topology hiding requires keeping internal state information in order to route the responses correctly. This is because the headers do not contain the original senders of the messages after hiding has been performed
Proprietary NAT and Firewall Traversal	• Media traversal functionality is implemented by forcing the media streams through the SBC on specific UDP ports by modifying SDP • REGISTER messages are modified in order to implement UDP SIP traversal • Utilizes B2BUA and media relay
Traffic monitoring	• Implemented by forcing the media streams through the SBC for monitoring by modifying SDP • Utilizes B2BUA and media relay
Traffic shaping	• Implemented by forcing the media streams through the SBC for shaping by modifying SDP • Utilizes B2BUA and media relay
IPv4/IPv6 Interworking	• Uses proprietary NAT traversal techniques modifying SDP to implement interworking. • Utilizes B2BUA and media relay

Legal Intercept	• Implemented by forcing the media streams through the SBC for creating a copy of the media by modifying SDP • Utilizes B2BUA and media relay
Transcoding	• Implemented by forcing the media streams through the SBC for Transcoding one codec to another by modifying SDP • Utilizes B2BUA and media relay

Table 6 SIP unfriendly features

All of the SIP unfriendly functions involve B2BUA mode of operation and SDP manipulation. Some functions perform SIP header manipulation and stateful operation.

5.3.1.2 Reasons for SIP Unfriendliness

B2BUA mode used to implement SDP manipulation to control the media path is considered SIP unfriendly [Cam05a] by the IETF because it breaks the end-to-end integrity of the media descriptions and does not work at all for user agents that encrypt or integrity protect their message bodies. There is no way for the user agents to distinguish manipulation done by a SBC from a malicious Man-in-The-Middle attack. In addition, the SBC needs to understand the session description protocol and all the extensions that might be used by the user agent in order to perform its functions successfully. Failing to do so can prevent a SBC from operating as intended. Correcting the problem could require updating software or at least configuration of SBCs in the network between user agents. This is feared to slow down overall new service innovation in the Internet.

Stateful operation required for some functions is considered unfriendly. The reason for this is that if state information is required for bi-directional message routing (e.g. topology hiding), losing that information due to failure, like a software restart, prevents the SBC from routing the responses related to established sessions.

SIP header field manipulation that is not allowed for SIP servers is considered harmful for obvious reasons. The user agents are usually unaware of the presence of a SBC in the path. The impact of a SBC that attempts to be transparent, but performs non-standard manipulation of SIP between user agents expecting standard operation can be unexpected. One example of this is the failure of SIP messages using end-to-end confidentiality or integrity protection in REGISTER message manipulation for NAT traversal. Again, there is no way for the user agents to distinguish manipulation done by a SBC from a malicious Man-in-The-Middle attack.

The SBC has significant impact on the functioning, security and privacy of a SIP network based on Internet standards. There is ongoing work by the IETF to decide if standard approaches should be developed to address the functions that SBCs are currently performing in an SIP unfriendly way.

5.3.2 SBC Functions and ETSI TISPAN NGN Standards

The European Telecommunications Standards Institute (ETSI) Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN) is defining the release 1 of the TISPAN Next Generation Network (NGN). TISPAN NGN has its foundation in the 3rd Generation mobile Partnership Project (3GPP) IP Multimedia Subsystem (IMS) Release 7. The TISPAN NGN project has selected SIP profiled by 3GPP TS 24.229 [3GP05] for the IMS as the protocol used to establish and tear down multimedia sessions in the context of NGN. The goal for TISPAN is to specify a common IMS core that is access independent serving both wireless and fixed-line access networks. TISPAN NGN Functional Architecture Release 1 is specified in ETSI ES 282 001 [ETS05].

Among others, the document specifies entities performing very similar functions to those of SBCs described in this document. The IMS and TISPAN architecture are complex systems and not central to this thesis. Therefore only the parts directly related to SBC functionality are discussed. Further details are available in 3GPP IMS [3GP05] and TISPAN NGN [ETS05] specifications.

5.3.2.1 Resource and Admission Control Subsystem (RACS)

RACS provides admission control and gate control functionalities, including the control of NAPT and priority marking.

- Admission control involves checking authorization.
- Checking resource availability verifying whether the requested bandwidth is compatible with both the subscribed and available bandwidth

The RACS functionality closely resembles the SBC access control functionality.

5.3.2.2 Border Gateway Function (BGF)

A Border Gateway Function (BGF) provides the interface between two IP-transport domains. It may reside at the boundary between an access network and the customer premises equipment, between an access network and a core network or between two core networks. It supports one or more of the functionalities listed in Table 7.

BGF Functionalities
Opening and closing gates i.e. firewall pinholes
Packet marking for outgoing traffic
Resource allocation and bandwidth reservation for upstream and downstream traffic
Allocation and translation of IP addresses and port numbers (NAPT)
Hosted NAT traversal
Policing of incoming traffic
Anti-spoofing of IP addresses
Usage metering
Interworking between IPv4 and IPv6 networks
Topology hiding.

Table 7 BGF functionalities

The present specification [ETS05] identifies two main types of BGF. The Core BGF (C-BGF) that sits at the boundary between an access network and a core network and the Interconnection BGF (I-BGF) that sits at the boundary between two core networks.

In addition to C-BGF and I-BGF, a specific type of BGF known as Resource Control Enforcement Function (RCEF) has been specified. It sits in an access network or at one of its edges. This functional entity implements a reduced subset of the functionalities identified for a generic BGF and holds a model of the access network resources. Sometimes an entity called A-BGF can be found with reference to TISPAN NGN architecture. The A-BGF stands for Access Border Gateway Function and may be considered a predecessor of the RCEF proposed during the development of the Release 1 specification. References to the I-BGF can be found i.e. in 3GPP-TISPAN joint workshop material [3GP04].

Counterparts for the TISPAN BGF and RCEF functions can be found in the SBC functions. Table 8 presents counterparts.

TISPAN Function	**SBC Function**
Opening and closing gates	Access control
Packet marking for outgoing traffic	QoS marking
Resource allocation and bandwidth reservation for upstream and downstream traffic	Call admission control
Allocation and translation of IP addresses and port numbers (NAPT)	NAT and firewall traversal
Hosted NAT traversal	NAT and firewall traversal
Policing of incoming traffic	DoS and overload prevention
Anti-spoofing of IP addresses	Access control
Usage metering	Traffic monitoring
Interworking between IPv4 and IPv6 networks	IPv4/IPv6 interworking
Topology hiding.	Topology hiding

Table 8 TISPAN BGF and RCEF functions with SBC counterparts

The roles of the BGF entities including RCEF have strong correlation with different SBC deployment scenarios. TISPAN I-BGF interfaces with other operators networks. A SBC in a peering scenario between operators in current infrastructures has a similar role with I-BGF. The RCEF and C-BGF are located between access and core network. A SBC deployed on the border of an ITSP service platform and the public Internet has a similar role. Figure 32 shows RACS, BGF and RCFE functions.

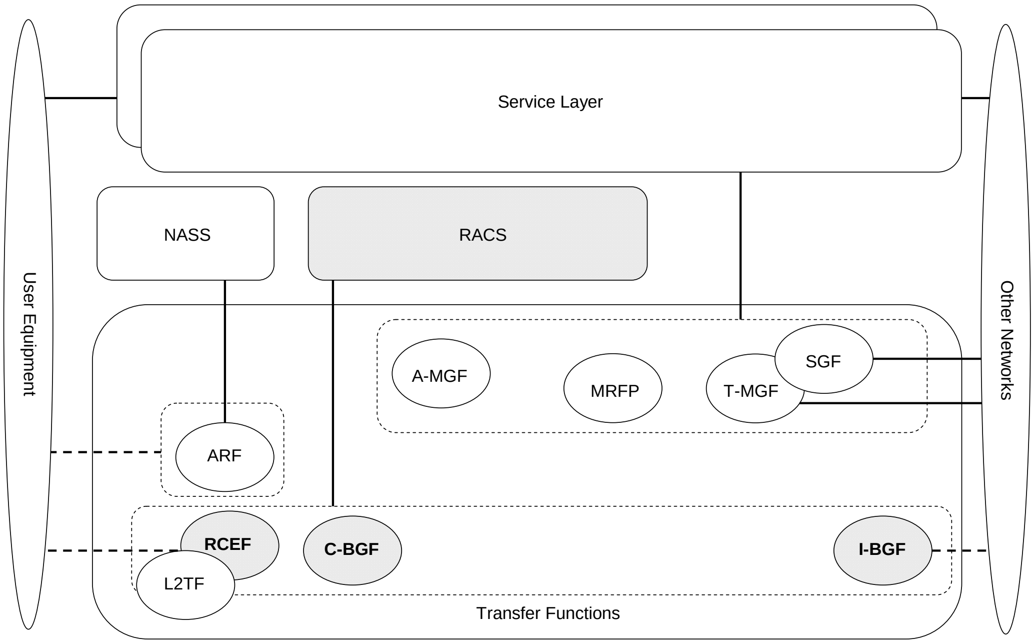

Figure 32 RACS, BGF and RCFE functions

5.3.2.3 Interworking Function (IWF)

The Interworking Function (IWF) performs the interworking between protocols used within TISPAN NGN service control subsystems and other IP-based protocols between the SIP profile used in the IMS and other SIP profiles or IP-based protocols such as the H.323 protocol. The IWF TISPAN performs similar function to the IWF found in SBCs.

5.3.2.4 The Interconnection Border Control Function (IBCF)

The Interconnection Border Control Function (IBCF) controls the boundary between two operators' domains. The IBCF controls the following functionality:

- Interaction with transport resources, through the resource and admission control subsystem (including NAPT and firewall functions)
- Insertion of the IWF in the signalling route when appropriate
- Screening of signalling information based on source/destination
- IMS Application Layer Gateway defined in TS 123 228.

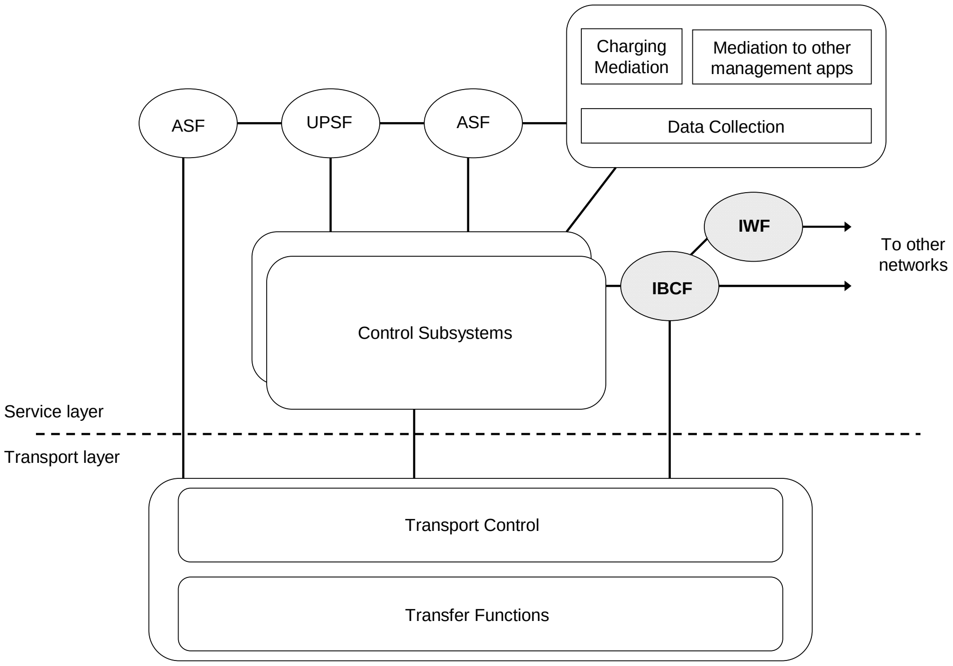

Figure 33 IBCF and IWF functions

There appears to be a close match between the functions specified in the TISPAN NGN Functional Architecture Release 1 and the functions performed by the current SBCs. It is not surprising that after the publication of the architecture, some SBC vendors have announced emerging support to some of the TISPAN NGN functions.

SBCs can be used to implement the following functions [Acm05a], [New05c]: P-CSCF, A-BGF, I-BGF, IBCF and IWF. The P-CSCF means Proxy-Call Session Control Function and is the first contact point for users within the 3GPP IMS [Poi04]. All SIP signalling traffic from or to the UE go via the P-CSCF. As the name of the entity indicates the P-CSCF behaves like a proxy as defined in RFC3261.

6 Conclusion

The aim of this thesis was to study the functions of a network element called Session Border Controller, and analyze the relationship of the functions it performs to IP multimedia standards. This chapter contains the conclusion of the work done.

After introducing the key concepts and protocols of IP multimedia technology, a high level description of the SBC was presented along with typical deployment scenarios. The goal of this was to provide the reader with an understanding how SBCs are used in the current IP communications infrastructure.

The key functions of SBCs were identified. Each function was presented and the motivation for using it was discussed. The implementations of the functions were presented along with descriptions of what standard each function is based on or if it represents non-standard functionality. The level of detail in available reference material describing SBC functionality varied between individual functions and from vendor to vendor. As a result of this it was not possible to describe and compare the different functions with an equal level of detail. After function descriptions, the relationship between the SBC functions and the specifications of related major standard bodies was discussed. This was the second goal of the thesis.

6.1 The Role of SBC in Current IP Multimedia Infrastructures

SBCs are used in operator, service provider and enterprise networks. They are used for a wide range of things centred on security, service assurance and quality, interoperation and even legal requirements. They are located on network boundaries and the boundaries can be related to technology or administrative responsibility. Technology related boundaries can be formed by different transport protocol versions like IPv4-IPv6, different methods to represent QoS, different signalling protocols like SIP versions and H.323, etc. Administrative boundaries are found for example between two peering operators, or between enterprises and service providers.

The role of the SBC on the boundary of networks is to enforce conformance to the technical and service requirements of the provider and to implement the administrative policies defined for the internal provider network.

6.1.1 SBC Functionality and the Standards

Concerning the first goal of the thesis, it was observed that functionality available in SBCs has evolved to address the practical real world issues, that hinder the wide spread use of standards based IP multimedia. The key functions of SBCs analyzed in this study appear to fill the gap between evolving or immature standards and a working real world implementation. One good example is NAT traversal.

Problems that SIP and H.323 have with NAT traversal in both residential and enterprise environment limit the usability of IP multimedia by limiting the number of users that can reach each other. Standard solutions based on STUN and TURN together with ICE provides a working solution when all three are used together, but both TURN and ICE are still drafts and considered to be work-in-progress. As a result they are rarely supported by endpoints, thus limiting the number of users that are able to communicate using them. The ability to communicate between two users behind NAT depends on level STUN/TURN/ICE support by the endpoints of both users and the types of NAT being used in the path. SBCs placed in the network by the service provider attempt to enable standard endpoints without any additional support for NAT traversal to be able to establish communications.

The functionality of a SBC is largely based on a SIP concept called Back-To-Back User Agent, B2BUA. This is a perfectly standard entity defined in SIP specification. One could argue that because the B2BUA is a standard SIP entity, all functions implemented using a B2BUA are standard too, as long as the SIP User Agent Client (UAC) and User Agent Server (UAS) contained within a B2BUA conform to the behaviour specified for a SIP User Agent (UA). The SIP specification does not restrict or specify the internal functionality of a B2BUA allowing it to perform any functionality.

When a SBC behaving as a perfectly legal B2BUA is used in a SIP infrastructure in the role of a proxy, redirect or registration server, it is possible to violate SIP specifications. A proxy server for example is allowed to modify requests and responses only according to strict rules set out in RFC 3261, while the B2BUA can perform any modifications.

Unfortunately, in order to perform its functions as intended, the SBC usually has to be deployed as a proxy or a registration server. This is the main reason why many of the SBC functions are non-standard.

6.1.2 The Controversial SBC

The SBC is used to enable communication between interconnected networks in a secure way. It performs functions like DoS prevention, DoS detection, access control and acts as a stateful firewall implementing security policies on application layer. On the other hand it breaks the end-to-end integrity and confidentiality between SIP user agents by inserting itself in the media path and modifying signalling in ways not allowed for SIP servers.

The SBC is used to improve the interoperation of IP multimedia systems. These cases include fixing signalling errors of endpoints, that claim to be standard compliant but have flaws in the implementation. SBCs fix interoperability issues between versions of the same protocol and implement protocol interworking between different protocols. While this functionality is used to enable and improve interoperation between systems, there are fears that some of the non-standard SBC behaviour is so application specific that the presence of a SBC in the path may prevent an application that the SBC does not understand from operating correctly. This is feared to slow down the development of new services in the Internet, because a new application, not violating any standards, might perform in a way has not been taken into account in the SBC application. The SBC might prevent the application from operating correctly.

6.1.3 SBC and IETF

As part of the second goal the views of the IETF on SBC were studied. The IETF views many key functions performed by the SBC SIP unfriendly. Standards based solutions are being developed for some functionality such as NAT traversal. Internet drafts on the SBC functions have been submitted and discussed in the SIPPING working group. At the moment there is no decision weather or not the IETF is going to develop standard mechanisms to the functions that current SBCs are performing.

6.1.4 SBC and ETSI TISPAN

As part of the second goal the views of ETSI on SBC were studied. The TISPAN R1 architecture includes a lot of functionality found in SBCs. Also the deployment scenarios are similar to those of SBC scenarios. It is interesting to note, that the same kind of functionality considered SIP unfriendly by the IETF, is explicitly required by the TISPAN architecture.

As noted in this study, SBC functionality has evolved out of the requirements of IP telephony service providers and operators focusing on providing telephony and other real-time IP multimedia services in the public Internet and enterprise IP VPN networks.

The IMS on the other hand is an architecture defined by 3GPP for the delivery of real-time multimedia services using SIP over IP networks, focusing on mobile wireless access. This architecture has been extended by TISPAN NGN to better satisfy the service requirements in fixed IP access networks. TISPAN NGN defines access independent (fixed, mobile) converged network architecture.

The SBC functions together with standard solutions like SIP enable fixed network operators and service providers to move from voice centric circuit switched services to IP based multimedia. The IMS is a similar shift form circuit switched voice to IP multimedia for mobile operators. TISPAN NGN can be considered as a union of the fixed and mobile approaches and therefore it is natural, that the TISPAN architecture includes elements from both worlds, including SBC functions.

6.2 Self Assessment and Future work

Writing this thesis was somewhat hard work, but nevertheless fun to do. The goals of this thesis were to compare the SBC with standards from different sources. While doing this, it revealed how the same set of SBC functionality is considered unfriendly by one standards body and as required by some other. I see that this controversy reflects the current state of convergence between Internet communications and communication services provided by telecommunication operators. I feel that the goals of this thesis were reached pretty well. One of the challenges was finding reference material for the literature study as very few independent publications on the subject exist. Vendor material and standards documents were used. Mailing lists such as the IETF SIPPING list provided a good source for clues. This thesis provides a snapshot of the current role of session border controllers in IP multimedia communication infrastructures.

The results of this work could be used in the development of IP multimedia services for converged networks combining Internet, PLMN and PSTN services and standardization of SBC functionality. The summary of SBC functionality by vendor presented in

Appendix A, could be used as a starting point for selecting a SBC product for a service platform implementation.

As the standards and best current practices relevant to SBCs are still evolving rapidly, it might be feasible to take another look at the state of standardization after a year or two. Traditional communication systems such as the ones based on 3GPP architecture represent centralized control of services, while Internet communications are often de-centralized or even peer-to-peer. SBCs are, among other things, used to connect the two worlds and it would be interesting to further study the issues of this border crossing.

References

[3GP04] 3GPP, “3GPP and ETSI TISPAN Workshop on NGN-IMS”, Report, June 2004, http://www.3gpp.org/ftp/workshop/Archive/2004-06-22_NGN-IMS_Sophia-Antipolis/Report%20WS%20NGN-IMS_TISPAN-3GPP_v01.doc, Referenced July 2005

[3GP05] 3GPP, “TS 24.229, IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), Stage 3 (Release 6)”, January 2005

[3GP05a] 3GPP, “3GPP Specifications - Releases (and phases and stages)”, http://www.3gpp.org/specs/releases.htm, Referenced September 2005

[Acm02] Acme Packet, “Session Admission Control: Interactive Communication SLAs over Skinny Pipes”, White Paper, 2002, http://www.acmepacket.com/images/wp_acmepacket_sessionadmincontrol.pdf, Referenced June 2005

[Acm04] Acme Packet, “Acme Packet Net-Net Session Border Controllers”, Product information, 2004, http://www.acmepacket.com/images/ap_product_info.pdf, Referenced June 2005

[Acm05] Acme Packet, “Session border controllers: Delivering interactive communications across IP network borders”, White Paper, http://www.acmepacket.com/images/whitepaper_SBC.pdf , September 2005

[Acm05a] Acme Packet Press Release, ”Acme Packet Defines Role of Session Border Controllers in Converged Fixed-Mobile IMS Architecture”, SUPERCOMM 2005, http://www.acmepacket.com/images/IMS-SBC_AcmePacket6_3_05.pdf, 2005

[Agr01] Agrawal H. & al, “SIP-H.323 Interworking Requirements”, Internet-Draft, Work in Progress, <draft-agrawal-sip-h323-interworking-reqs-00>, January 2001

[And03] Andreasen F., Foster B., “Media Gateway Control Protocol (MGCP) Version 1.0”, RFC 3435, Informational, January 2003

[Bak04] Baker F. & al, “Cisco Architecture for Lawful Intercept in IP Networks”, RFC 3924, October 2004

[Bar05] Barnes M., “Middlebox Communications (MIDCOM) Protocol Evaluation”, RFC 4097, June 2005

[Bau04] Baugher M. & al, “The Secure Real-time Transport Protocol (SRTP)”, RFC 3711, Standards Track, March 2004

[Bei98] Beijar Niclklas, ”Signalling Protocols for Internet Telephony – Architectures based on H.323 and SIP”, 1998

[Bla98] Blake S. & al, “An Architecture for Differentiated Service”, RFC 2475, December 1998

[Bra94] Braden R., “Integrated Services in the Internet Architecture: an Overview”, RFC 1633, June 1994

[Bra97] Braden R., “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification”, RFC 2205, September 1997

[Cam02] Camarillo, G, ”SIP Demystified”, McGraw-Hill, 2002

[Cam05] Camarillo G., “Functionality of Existing Session Border Controller (SBC)”, Internet-Draft, Work in Progress <draft-camarillo-sipping-sbc-funcs-00>, January 2005

[Cam05a] Camarillo, G, “SIP (Session Initiation Protocol)-Unfriendly Functions in Current Communication Architectures”, Internet-Draft, Work in Progress, <draft-camarillo-sipping-sbc-funcs-01>, July 2005

[Com04] IEEE Computer, “Will Interoperability Problems Give IP Telephony a Busy Signal?”, March 2004

[Cue00] Cuervo F. & al, “Megaco Protocol Version 1.0”, RFC 3015, Standards Track, November 2000

[Die99] Dierks T. & al, “The TLS Protocol Version 1.0”, RFC 2246, Standards Track, January 1999

[ETS05] ETSI ES 282 001, ” NGN Functional Architecture Release 1”, June 2005

[ETS05a] ETSI, “Want to know about ETSI?”, http://www.etsi.org/about_etsi/5_minutes/home.htm, Referenced September 2005

[Fau02] Le Faucheur F. & al, “Multi-Protocol Label Switching (MPLS) Support of Differentiated Services”, RFC 3270, May 2002

[FCC05] Federal Communications Commission, ”FCC Requires Certain Broadband and VoIP Providers to Accommodate Wiretaps”, News Media Information, August 2005, http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-260434A1.pdf, Referenced August 2005

[FCC05a] Federal Communications Commission, “STATEMENT OF CHAIRMAN KEVIN J. MARTIN”, June 2005 http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-05-116A2.pdf, Referenced July 2005

[FCC05b] Federal Communications Commission, “E911 Requirements for IP-Enabled Service Providers”, June 2005, http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-05-116A1.pdf, Referenced July 2005

[Fle05] Flextronics Software Systems, “Session Border Controller – A solution for NAT/Firewall traversal issues”, White Paper, http://www.hssworld.com/whitepapers/overview.htm, September 2005

[Gla05] Gladwin Dave, “Inter-working IPv4/IPv6: Crossing the Generation Gap”, Conference presentation at International SIP 2005 Paris, January 2005

[Han98] Handley M., Jacobson V, “SDP: Session Description Protocol”, RFC 2327, Standards Track, April 1998

[Han99] M. Handley & al, “SIP: Session Initiation Protocol”, RFC 2543, Standards Track, March 1999

[Har03] Hardy, William C, “VoIP Service Quality, Measuring and Evaluating Packet-Switched Voice”, McGraw-Hill, 2003

[IEE98] IEEE Standard, “IEEE 802.1p – Traffic class expediting and dynamic multicast filtering”, 1998

[IET05] The Internet Engineering Task Force, “Overview of the IETF“, http://www.ietf.org/overview.html, Referenced August 2005

[Int05] Intel, Jasomi, “Delivering Secure IP-Based Services”, Solutions White Paper, 2005, http://download.intel.com/design/telecom/papers/9668sw.pdf, Referenced August 2005

[ITU00] ITU-T, “Packet-Based Multimedia Communications Systems Recommendation H.323”, 2000.

[ITU03] ITU-T “Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals “, Recommendation H.235, 2003

[ITU04] ITU-T, “General Overview of NGN”, Recommendation Y.2001, 2004

[ITU96] ITU-T, “Packet-Based Multimedia Communications Systems Recommendation H.323”, 1996.

[ITU97] ITU-T, “Packet-Based Multimedia Communications Systems Recommendation H.323”, 1997

[Jas04] Jasomi, “PeerPoint far-end SBC”, http://www.jasomi.com/PPfes%20Data%20Sheet-small.pdf, 2004, Referenced June 2005

[Jas05] Jasomi Networks, “PeerPoint A500 Session Border Controller”, Data Sheet, June 2005, http://www.jasomi.com/PPA500%20Data%20Sheet.pdf, Referenced August 2005

[Joh03] Johnston Alan B. & al, “Session Initiation Protocol (SIP) Basic Call Flow Examples”, RFC 3665, Best Current Practice, December 2003

[Joh04] Johnston Alan B., “SIP: Understanding the Session Initiation Protocol”, Artech House, ISBN 1-58053-655-7, 2004

[Jun05] Juniper Networks, “VF-Series Network Protection“, White Paper, http://www.juniper.net/solutions/literature/white_papers/200122.pdf, September 2005

[Jun05a] Juniper Networks, “Evolution of Session Border Controllers, The Need for Three Architectures”, White Paper, April 2005

[Jun05b] Juniper Networks, “The Role of Session Border Controllers in Public IP Networks”, White Paper, http://www.juniper.net/solutions/literature/white_papers/200123.pdf, 2005

[Jun05c] Juniper Networks, “Voice Flow 3000 series”, Product Data Sheet, February 2005

[Kar05] Karila Arto, “Internet-puhelut (VoIP). Selvitys”, Raportti, February 2005, http://www.mintc.fi/oliver/upl402-Julkaisuja%2016_2005.pdf, Referenced June 2005

[Kum01] Kumar Vineet & al, “IP Telephony with H.323”, John Wiley & Sons, Inc., ISBN 0-471-39343-6, 2001

[New05] Newport Networks, “The Need for a Carrier Class Session Controller”, White paper, http://www.newport-networks.com/whitepapers/index.html, Referenced September 2005

[New05a] Newport Networks, “Extending the Reach of Voice and Multimedia Services over IP“, White paper, http://www.newport-networks.com/whitepapers/index.html, Referenced September 2005

[New05b] Newport Networks, “Session Controllers and 3GPP“, White paper, http://www.newport-networks.com/whitepapers/index.html, September 2005

[New05c] Newport Netwoks, “Session Border Control Functions in IMS Based Converged Networks”, White Paper, September 2005, White paper, http://www.newport-networks.com/whitepapers/index.html, Referenced September 2005

[New05d] Ed Luff, Newport Networks, “Session Controllers, for Fun & Profit“, VON 2005 Canada conference presentation, 2005

[New05e] Newport Networks, “Session Border Control Functions in IMS Based Converged Networks “, White paper, http://www.newport-networks.com/whitepapers/index.html, Referenced September 2005

[New05f] Newport Networks, “Solving the Firewall and NAT Traversal Issues for Multimedia Services over IP “, White paper, http://www.newport-networks.com/whitepapers/index.html, Referenced September 2005

[New05g] Newport Networks, “IPv4 – IPv6 Inter-Working in Multimedia Networks“, White paper, http://www.newport-networks.com/whitepapers/index.html, Referenced September 2005

[Nex04] NexTone, “NexTone Session Border Controller”, Data Sheet, 2004, http://www.nextone.com/docs/Datasheet_SBC.pdf, Referenced September 2005

[Nic98] Nichols K. & al, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC 2474, December 1998

[Nok04] Nokia, “SIP Frequently Asked Questions”, Forum Nokia, Version 1.0, May 2004”, http://sw.nokia.com/id/4e5f782f-1848-49f6-9cb1-63c553d1a7b0/SIP_FAQ_v1_0_en.pdf, Referenced August 2005

[Poi04] Poikselkä Miikka & al, “The IMS IP Multimedia Concepts and Services in the Mobile Domain”, John Wiley and Sons, Ltd, 2004

[Pos94] Postel J., “Media Type Registration Procedure“, RFC 1590, Informational, March 1994

[Rod04] Rodrigues Marco P., ”VoIP Signalling in Carrier Networks”, 2004, http://www.rodrigues.ca/papers/carriervoip/VoIP_Signaling_in_Carrier_Networks.pdf, Referenced August 2005

[Ros00] Rosenberg J. & al, “Getting SIP through Firewalls and NATs”, Internet-Draft, Work in Progress < draft-rosenberg-sip-firewalls-00>, February 2000

[Ros02] Rosenberg J. & al, “SIP: Session Initiation Protocol“, RFC 3261, Standards Track, June 2002

[Ros03] Rosenberg J. & al, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)”, RFC 3489, March 2003

[Ros03a] Rosenberg J. & al, “Traversal Using Relay NAT (TURN)”, Internet-Draft, Work in Progress < draft-rosenberg-midcom-turn-02>, October 2003

[Ros03b] Rosenberg J., “Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)”, Internet-Draft, Work in Progress <draft-rosenberg-sipping-ice-01>, June 2003

[Ros05] Rosenberg J., “Considerations for Selection of Techniques for NAT Traversal”, Internet-Draft, Work in Progress <draft-iab-nat-traversal-considerations-00>, February 2005

[Sch04] Schulzrinne H., Agboh C., “Session Initiation Protocol (SIP)-H.323 Interworking Requirements”, Internet-Draft, Work in Progress, < draft-agrawal-sip-h323-interworking-reqs-07>

[Sch05] Schulzrinne H., Agboh C., “Session Initiation Protocol (SIP)-H.323 Interworking Requirements”, RFC 4123, Informational, July 2005

[She97] Shenker S., Wroclawski J., “Network Element Service Specification Template”, RFC 2216, September 1997

[Sno05] Snom AG, “Snom 4S NAT Filter Admin Manual”, http://www.snom.com/download/natf_211.pdf, Referenced September 2005

[Sri02] Srisuresh P. & al, “Middlebox communication architecture and framework”, RFC 3303, Informational, August 2002

[Stu04] Stukas Michael, Sicker Douglas C., ”An Evaluation of VoIP Traversal of Firewalls and NATs within an Enterprise Environment”, Kluwer Academic Publishers, 2004

[Tel03] Telecommunications Magazine, “Session Border Control: making dreams reality”, September 2003

[Tsi00] Tsirtsis G., Srisuresh P., “Network Address Translation - Protocol Translation (NAT-PT)”, RFC 2766, Standards Track, February 2000

[Vie03] Viestintävirasto, “VIESTINTÄVIRASTON PÄÄTÖS KOSKIEN SONERA PUHEKAISTA -PALVELUN LAINMUKAISUUTTA“, FICORA Decision, October 2003, http://www.ficora.fi/suomi/document/paat_teliasonera_2910.pdf, Referenced June 2005

Appendix A

Studied Session Border Controller Functions by Vendor

	Acme Packet	Flextronix Software	Jasomi	Juniper Networks	Newport Networks	NexTone	Snom
NAT & FW traversal	-B2BUA	-MIDCOM	- B2BUA	-proxy -B2BUA -MIDCOM	-B2BUA Proprietary (Automatic Channel Mapping, ACM)	-Yes	-STUN -TURN -ICE -B2BUA
Traffic monitoring	Yes	?	?	Yes	Yes	?	SNMP
Traffic shaping	Yes	?	?	Yes	Yes	?	Yes
QoS marking	Yes: DiffServ, MPLS, RSVP	?	?	802.1p, DiffServ	ToS bits, DiffServ)	?	No
Signalling Protocol repair and variant interoperation	Yes	?	Yes	Yes	Yes	Yes	?
Signalling IWF	SIP H.323 IMS SIP	?	SIP H.323	SIP H.323	SIP H.323 IMS SIP	SIP H.323	No
IPv4/IPv6 Interworking	?	Yes	?	?	Yes	Yes	No
Transport protocol interworking	SIP TCP/UDP	SIP TCP/UDP	SIP TCP/UDP/TLS	SIP TCP/UDP/TLS	SIP TCP/UDP	SIP TCP/UDP H.235	SIP TCP/UDP/TLS
DoS and Overload prevention	Yes	?	Yes	Yes	?	Yes	No
Call Admission Control	Yes	?	?	Yes	Yes	Yes	No
Legal Intercept	Yes	?	Yes	Yes	Yes	Yes	No
Emergency services	Yes	?	?	Yes	Yes	?	?
Media encryption	?	?	Yes	?	?	?	No
Media transcoding	?	?	Yes	?	?	Yes	No
References	[Acm02] [Acm04] [Acm05]	[Fle05]	[Jas05] [Jas04] [Int05]	[Jun05a] [Jun05b] [Jun05c]	[New05f] [New05g] [New05d] [New05e]	[Nex04]	[Sno05]

Yes	Functionality exists, but details on method are unknown
No	Functionality does not exist
?	Support of functionality is unknown

Appendix B: SBC test setup

The following test setup was used to gather traces in order to analyze non-standard SBC functions in this thesis:

- SIP UA: X-Lite release 1103m build stamp 14262
- NAT Firewall: Fedora Core Linux 3, Kernel 2.6.9-1.667, Firewall with port restricted cone NAT configuration
- SBC: Snom 4S NAT Filter session border controller v2.11 running on Fedora Core Linux 3, Kernel 2.6.9-1.667
- IP telephony services provided by Free Wold Dialup (http://www.fwd.pulver.com/).

The traces were obtained using Ethereal (http://www.ethereal.com/) network protocol analyzer running on the firewall node. Call flow diagrams were generated using a tool called SIP Scenario Generator (http://www.iptel.org/~sipsc/).

The setup is illustrated by the Figure below.

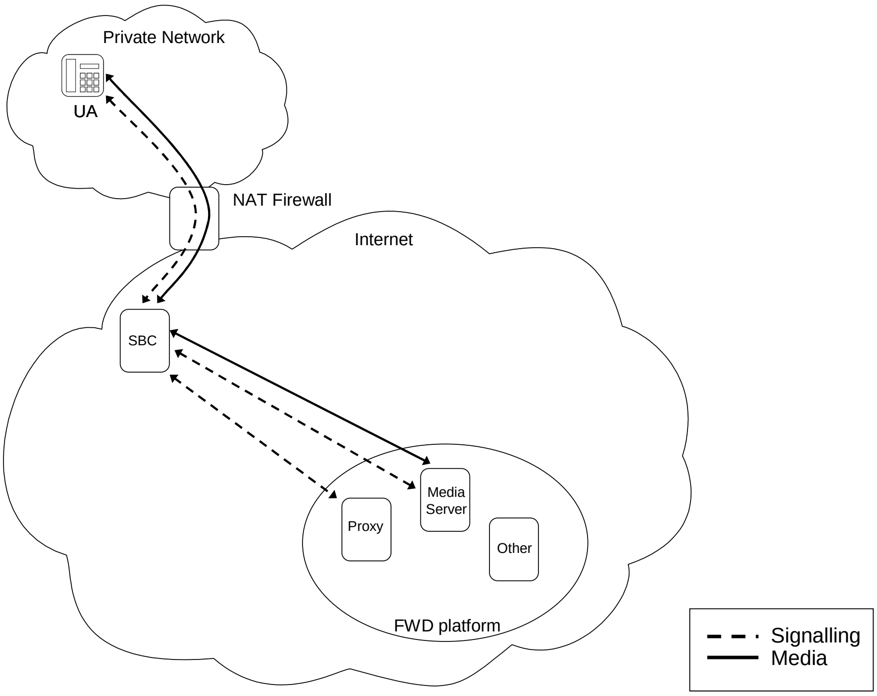

Figure: Test setup with SBC