Firewall traversal methods and security considerations for unified communication architectures with industry case studies

Thomas Reisinger

Technical Report

RHUL–ISG–2017–10

10 February 2017

Information Security Group
Royal Holloway University of London
Egham, Surrey, TW20 0EX
United Kingdom
Firewall Traversal Methods and Security Considerations for Unified Communication Architectures with Industry Case Studies

Thomas Reisinger  
Registration No. 101083083  
March 21, 2016

Supervisor: Prof. Dr. Peter Komisarczuk  
Submitted as part of the requirements for the award of the MSc in Information Security of the University of London.
Declaration of Authorship

I, Thomas Reisinger, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed: 
(Thomas Reisinger)

Date: 
Acknowledgements

I would like to thank my wife Maria for her patience over the last couple of years during my journey through the MSc program and in the final stage of this dissertation and exams.

I’d like to thank my supervisor, Dr. Peter Komisarczuk for his support and guidance throughout this project.

Another “Thank You!” goes to Emily Hutchinson from the UK for proofreading.
Summary

This work informs the reader about various aspects of Unified Communication (UC), methods used and the problems and solutions faced by real-life organisations in its implementation surrounding firewall traversal and related security considerations.

The beginning provides an introduction into the various definitions that will aid in the reading of this document including the past, present and future of Unified Communication followed by an introduction to the various protocols used, with particular focus on Session Initial Protocol (SIP). With the help of examples it is explained how Session Initial Protocol works in concert with relevant industry standards including new technologies such as WebRTC.

Some time is spent discussing the problems of firewall transversal and network address translation (NAT) as this is a significant issue faced by organisations around the world. There is a focus put on the various solutions for these issues and their shortfalls. Hole Punching, STUN, TURN, ICE and SBCs are all featured, with examples and flow diagrams to aid understanding of how these methods are implemented.

A close look at existing literature and standards has been taken, to ensure that most common technologies, including the up to date approaches are covered. For demonstration purpose a dedicated test environment has been used with working examples of how the identified technologies work using genuine industry products as case studies.

A large section of this study is focused on UC security, as this is a major concern for all organisations and their users and customers. Firewall traversal and NAT is still a major issue, but there are solutions, and these solutions are covered within the scope of this document. Mention is also made of the need for high quality UC and UC that the average person will take up and use without complaint that it is too complicated or not fit for purpose.

During the research for this dissertation, and by experience of the author, it has been discovered that insecure UC is abused and several examples of this are given in this document. Poor or in-consequent implementation of the highlighted security services can have devastating effects on an organisation, with data security, reputation and financially.
Contents

1 Introduction 1
   1.1 Motivation and Objectives of the Document .......................... 1
   1.2 Structure of the Document ........................................... 1
   1.3 Scope of Document .................................................. 2

2 Overview 3
   2.1 Unified Communication .............................................. 3
   2.2 The History of UC .................................................. 3
   2.3 The UC Promise ..................................................... 10

3 Video Conferencing 15
   3.1 Endpoints ............................................................ 15
   3.2 Video Conferencing Specific Infrastructure ......................... 16
   3.3 Call Types, Video Protocols and Formats .......................... 18
   3.4 WebRTC ............................................................... 21
   3.5 Conclusion ........................................................... 23

4 Protocols and Related Standards 25
   4.1 Overview .............................................................. 25
   4.2 SIP - Session Initiation Protocol ................................... 25
      4.2.1 Point to Point Call (P2P) ..................................... 27
      4.2.2 SIP Call with Proxy Server ................................... 38
      4.2.3 Transport Protocol Selection .................................. 41
   4.3 XMPP - Extensible Messaging and Presence Protocol ............... 41
   4.4 VC Endpoint and Software Client Management ....................... 44
   4.5 Firewalls .............................................................. 45
      4.5.1 Network Address Translation - NAT .......................... 48
   4.6 SIP and Media NAT/Firewall Problems ................................ 51
      4.6.1 Hole Punching .................................................. 52
      4.6.2 STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) ................. 56
      4.6.3 TURN - Traversal Using Relays around NAT ................. 57
      4.6.4 ICE - Interactive Connectivity Establishment ................ 58
   4.7 Session Border Controller (SBC) ................................... 58
   4.8 Conclusion ........................................................... 59

5 Security Considerations for UC 61
   5.1 UC Threats, Attacks, and Risk Mitigation .......................... 61
      5.1.1 Generic Threats ................................................ 61
## Contents

5.1.2 SIP Threats .............................................................. 62  
5.1.3 SIP Security Services .................................................. 63  
5.1.4 SIP Security Services Implementation ................................ 66  
5.1.5 Media Security ........................................................... 69  
5.2 Real Life Examples ......................................................... 71  
5.2.1 Anatomy of an SIP Attack .............................................. 71  
5.2.2 ISDN Gateway Misuse .................................................. 72  
5.3 UC Security Recommendations from Vendors ........................... 74  
5.4 Conclusion ........................................................................ 75  

6 Case Studies ...................................................................... 77  
6.1 Test Environment ............................................................ 77  
6.2 SIP Registration and Management Interface ........................... 79  
6.3 SIP Based Firewall Traversal Calls via SBC ............................. 83  
6.3.1 Call Scenario: roomsystem2 → roomsystem3 (P2P) .......... 83  
6.3.2 Call Scenario: mobile and roomsystem2 → VMR (P2M) .... 85  
6.3.3 Call Scenario: roomsystem2 → mobile.android (P2P) ....... 87  
6.4 Conclusion ........................................................................ 88  

7 Conclusions and Key Findings .............................................. 89  
7.1 Introduction and Overview ................................................ 89  
7.2 Protocols and Standards ................................................... 89  
7.3 Security Considerations for UC .......................................... 90  
7.4 Case Studies ..................................................................... 91  
7.5 Future Work ..................................................................... 91  

Appendix .............................................................................. 101  
1 Test Environment Components ........................................... 101  
2 Public DNS Server ............................................................. 102  
3 Firewall HQ Configuration Details ....................................... 103  
4 Endpoint Configuration and Call Details ................................. 105  
5 SBC Configuration Details ................................................... 107
List of Figures

2.1 Ovum 2013 Study - Technology and Features[11] . . . . . . . . . . . . . . 6
2.2 Ovum 2013 Study - Managed UC[11] . . . . . . . . . . . . . . . . . . . . . . 7
2.3 Ovum 2013 Study - Supplier Trust [11] . . . . . . . . . . . . . . . . . . . 8
2.4 Wainhouse Total Endpoint Units Worldwide Sold[12] . . . . . . . . . . . . 9
2.5 Wainhouse Video Infrastructure Revenue History[12] . . . . . . . . . . . . 9
3.1 Video Conferencing Endpoints (Examples) . . . . . . . . . . . . . . . . . . 15
3.2 Polycom RealPresence Mobile iPad - Software Client[20] . . . . . . . . . . 16
3.3 VC Call Types . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.4 AVC Continuous Presence (CP) video streams and built layouts[30] . . . 20
3.5 SVC video streams and layouts[30] . . . . . . . . . . . . . . . . . . . . . . 21
3.6 WebRTC Simple Call Model[35] . . . . . . . . . . . . . . . . . . . . . . . . 22
4.1 Common UC Protocol Landscape mapped to the Internet Model . . . . . 25
4.2 Basic SIP P2P call setup and tear down . . . . . . . . . . . . . . . . . . . 27
4.3 SIP INVITE message . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
4.4 SDP fields[18] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
4.5 Media format details for audio example . . . . . . . . . . . . . . . . . . . 33
4.6 SDP attributes example . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
4.7 SIP 180 Ringing message . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
4.8 SIP 200 OK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.9 SIP ACK - Acknowledge example . . . . . . . . . . . . . . . . . . . . . . . 36
4.10 SIP BYE message example . . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.11 SIP BYE OK message example . . . . . . . . . . . . . . . . . . . . . . . . 37
4.12 SIP P2P call with proxy server . . . . . . . . . . . . . . . . . . . . . . . . 38
4.13 SIP P2P call with proxy . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.14 SIP REGISTER message . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.15 XMPP example flow . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.16 Polycom GAB example dialogue . . . . . . . . . . . . . . . . . . . . . . . 45
4.17 Example for ALG settings on Juniper SSG20 firewall . . . . . . . . . . . 47
4.18 Basic NAT example[56] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.19 NAPT example[56] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
4.20 SIP INVITE message highlighting NAT/firewall problems . . . . . . . . 52
4.21 Before UDP Hole Punching, Endpoints behind different NATs[69] . . . 53
4.22 UDP Hole Punching, Endpoints behind different NATs process[69] . . . 54
4.23 UDP Hole Punching, Endpoints behind different NATs completed[69] . . 54
4.24 STUN Flow Diagram[73] . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.25 TURN Flow Diagram[73] . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
4.26 Example ICE SDP candidates . . . . . . . . . . . . . . . . . . . . . . . . . 58
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.27</td>
<td>SBC Architecture[74]</td>
<td>59</td>
</tr>
<tr>
<td>5.1</td>
<td>SIP Security Implementation REGISTER and Interdomain INVITE</td>
<td>67</td>
</tr>
<tr>
<td>5.2</td>
<td>Example SRTP RTP/SAVP profile and a=crypto in SIP/SDP INVITE</td>
<td>70</td>
</tr>
<tr>
<td>5.3</td>
<td>Example ISDN Gateway Misuse</td>
<td>72</td>
</tr>
<tr>
<td>6.1</td>
<td>Project Test Environment</td>
<td>78</td>
</tr>
<tr>
<td>6.2</td>
<td>SIP Proxy Settings</td>
<td>79</td>
</tr>
<tr>
<td>6.3</td>
<td>roomsystem1 - HDX Certificate Settings</td>
<td>79</td>
</tr>
<tr>
<td>6.4</td>
<td>roomsystem1 - HDX SIP Settings</td>
<td>80</td>
</tr>
<tr>
<td>6.5</td>
<td>roomsystem3 REGISTER</td>
<td>80</td>
</tr>
<tr>
<td>6.6</td>
<td>roomsystem3 REGISTER OK</td>
<td>81</td>
</tr>
<tr>
<td>6.7</td>
<td>roomsystem2 - Registration in dynamic mode</td>
<td>81</td>
</tr>
<tr>
<td>6.8</td>
<td>Call Scenario: roomsystem2 → roomsystem3 (P2P)</td>
<td>83</td>
</tr>
<tr>
<td>6.9</td>
<td>Call Scenario: roomsystem2 → roomsystem3 (P2P) signalling</td>
<td>84</td>
</tr>
<tr>
<td>6.10</td>
<td>Call Scenario: mobile.android and roomsystem2 → VMR (P2M)</td>
<td>85</td>
</tr>
<tr>
<td>6.11</td>
<td>Call Scenario: roomsystem2 → roomsystem3 (P2P) signalling</td>
<td>86</td>
</tr>
<tr>
<td>6.12</td>
<td>Call Scenario: roomsystem2 → mobile.android (P2P)</td>
<td>87</td>
</tr>
<tr>
<td>6.13</td>
<td>Call Scenario: roomsystem2 → mobile.android (P2P) signalling</td>
<td>88</td>
</tr>
<tr>
<td>1</td>
<td>Firewall Rules: Internet → DMZ-PUBLIC</td>
<td>103</td>
</tr>
<tr>
<td>2</td>
<td>Firewall Rules: DMZ-PUBLIC → Internet</td>
<td>104</td>
</tr>
<tr>
<td>3</td>
<td>Firewall Rules: DMZ-PRIVATE → Inside</td>
<td>104</td>
</tr>
<tr>
<td>4</td>
<td>Firewall Rules: Inside → DMZ-PRIVATE</td>
<td>104</td>
</tr>
<tr>
<td>5</td>
<td>GroupSeries SIP Settings roomsystem3</td>
<td>105</td>
</tr>
<tr>
<td>6</td>
<td>GroupSeries Certificate Settings roomsystem3</td>
<td>106</td>
</tr>
<tr>
<td>7</td>
<td>roomsystem2 call statistics in VMR call</td>
<td>106</td>
</tr>
<tr>
<td>8</td>
<td>SBC SIP Signalling settings</td>
<td>107</td>
</tr>
<tr>
<td>9</td>
<td>SBC Access Proxy settings</td>
<td>107</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>VC Bandwidth Requirement Examples[21, 22]</td>
<td>17</td>
</tr>
<tr>
<td>3.2</td>
<td>Recommended VC DSCP QoS settings[23]</td>
<td>17</td>
</tr>
<tr>
<td>4.1</td>
<td>SDP parameter example</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>SIP 200 OK media information example</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>SIP Security Service Requirements for SIP Elements</td>
<td>66</td>
</tr>
<tr>
<td>1</td>
<td>Test Environment Components 1 of 2</td>
<td>101</td>
</tr>
<tr>
<td>2</td>
<td>Test Environment Components 2 of 2</td>
<td>102</td>
</tr>
</tbody>
</table>
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>Access Control List</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standards</td>
</tr>
<tr>
<td>ALG</td>
<td>Application Layer Gateway</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interfaces</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>B2B</td>
<td>Business to Business (call)</td>
</tr>
<tr>
<td>B2BUA</td>
<td>Back to Back User Agent</td>
</tr>
<tr>
<td>BFCP</td>
<td>Binary Floor Control Protocol</td>
</tr>
<tr>
<td>BYOD</td>
<td>Bring Your Own Device</td>
</tr>
<tr>
<td>CDR</td>
<td>Call Detail Records</td>
</tr>
<tr>
<td>CODEC</td>
<td>Coding and Decoding</td>
</tr>
<tr>
<td>CPE</td>
<td>Customer (on) premises equipment</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Services (attack)</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DMZ</td>
<td>Demilitarized Zone</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signalling Processor</td>
</tr>
<tr>
<td>DTMF</td>
<td>Dual Tone Multiple Frequency</td>
</tr>
<tr>
<td>GAB</td>
<td>Global Address Book</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messaging</td>
</tr>
<tr>
<td>IMTC</td>
<td>International Multimedia Telecommunication Consortium</td>
</tr>
<tr>
<td>ISMS</td>
<td>Information Security Management System</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telegraph Union</td>
</tr>
<tr>
<td>IVR</td>
<td>Interactive Voice Response</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MCU</td>
<td>Multipoint Control Unit</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>NAPT</td>
<td>Network Address and Port Translation</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>P2M</td>
<td>Point To Multipoint (MCU call)</td>
</tr>
<tr>
<td>P2P</td>
<td>Point To Point (call)</td>
</tr>
<tr>
<td>PBX</td>
<td>Private Branch Exchange</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RDP</td>
<td>Remote Desktop Protocol</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments (IETF)</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource Reservation Protocol</td>
</tr>
<tr>
<td>RTT</td>
<td>Round Trip Time</td>
</tr>
<tr>
<td>SBC</td>
<td>Session Border Controller</td>
</tr>
<tr>
<td>SfB</td>
<td>Skype for Business (previously Lync)</td>
</tr>
<tr>
<td>SDP</td>
<td>Session Description Protocol</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UA</td>
<td>User Agent (SIP)</td>
</tr>
<tr>
<td>UAC</td>
<td>User Agent Client (SIP)</td>
</tr>
<tr>
<td>UAS</td>
<td>User Agent Server (SIP)</td>
</tr>
<tr>
<td>UC</td>
<td>Unified Communication</td>
</tr>
<tr>
<td>UCIF</td>
<td>Unified Communication Interoperability Forum</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UM</td>
<td>Unified Messaging</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Indicators</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VC</td>
<td>Video Conferencing</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual LAN</td>
</tr>
<tr>
<td>VMR</td>
<td>Virtual Meeting Room</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WebRTC</td>
<td>Web Real-Time Communications</td>
</tr>
<tr>
<td>XMPP</td>
<td>Extensible Messaging and Presence Protocol</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Motivation and Objectives of the Document

Firewall traversal is a common challenge for organisations around the globe and needs to be properly designed in order to allow Unified Communications (UC) to work across security boundaries such as firewalls, which are a significant building block to enforce security policies to achieve an organisation’s security requirements and reduce risk.

During the research by the author for this project no comprehensive documentation was identified, which highlights this problem area and solutions including emerging technologies such as WebRTC in a holistic approach. It is important to have some real life related examples in order to see all the theory and protocols in “action” with examples and explain at an appropriate level how it works. The main objectives of this dissertation are:

- Setting the scene and covering some definitions e.g. history, future, and the building blocks of Unified Communication (UC).
- Introducing UC related protocols, with a focus on Session Initiation Protocol (SIP) and related standards with examples.
- Highlighting the firewall traversal and network address translation (NAT) problem.
- Explaining the use of Hole Punching, STUN, TURN, ICE, and SBCs as solutions for the firewall traversal problem with examples and flow diagrams.
- Reviewing existing literature and standards to ensure state of the art technologies like WebRTC and approaches are covered.
- Utilising a dedicated test environment to demonstrate, verify and show how the identified technologies work with real industry products as case studies.

1.2 Structure of the Document

Chapter 2 starts with an introduction and sets the scene for Unified Communication (UC), its’ history, and a glimpse into potential future developments of the industry and trends. In chapter 3, a closer focus on video conferencing is presented as part of the UC landscape, highlighting the specifics surrounding real time traffic and the requirements and technologies used. Chapter 4 The Protocols and Related Standards explains in more detail the required protocols surrounding SIP, along with associated technologies. In Chapter 5, Security Considerations, the UC security specifics are highlighted and explained with real life examples. The case studies are listed in chapter 6 followed
by the conclusions and key findings in Chapter 7. Configuration details of the test environment are listed in the appendix.

1.3 Scope of Document

The scope of this document is focused around the UC building blocks SIP based call control, video conferencing and the specifics of firewall traversal and security considerations. Out of scope of this document is for example the signalling protocol H.323, foundation technologies such as TCP/IP, DNS, DHCP, and related services. Some good sources regarding these technologies can be found here [1, 2, 3].
2 Overview

This chapter discusses the Unified Communication ecosystem and related technologies including the history of UC. The scope of this document is focused on Unified Communication surrounding Video Conferencing, which is part of the UC landscape.

2.1 Unified Communication

The term Unified Communication (UC) has been around for more than 10 years and it has various definitions. Sometimes the term is simply being used as a buzzword, but a general definition would be the interactive use of different real time communication methods/channels and the integration with IT business applications and processes across multiple devices and media types presenting a consistent unified user interface[4, 5]. Please keep in mind that the term communication channel is not limited to voice only, it includes all kinds of transport of information. This could be for example voice, video, content sharing or instant messaging.

2.2 The History of UC

The Circuit Switched Age
In the early days of enterprise communication, the telephone was key and was largely operated by telephone companies, providing and operating circuits/lines for their customers. The telephone calls to and from the customer were routed by the telephone provider. This left the customer highly dependent on their telephone service provider and there would be recurring costs to pay. Over time, the customers moved away from this kind of service, moving it in-house, including all the hardware, and operated their own telephone systems, giving them more flexibility, and a greater degree of control.

With the growth of IP based networks, Private Branch Exchange (PBX) vendors such as Alcatel, Nortel and Avaya have developed interfaces to interconnect their PBX systems via IP links instead of with cost intensive static circuits. They are also utilising their IP networks for sharing media for a variety of services. The main advantage of using IP as a carrier for real time video and audio data, is that it provides a converged network. This same network infrastructure can be used to carry voice, video, web traffic, e-mail, business applications, file sharing and much more. This concept can provide financial savings on implementation, operation and management of this infrastructure. This development has led to a move from hardware based PBX into more flexible software based products and solutions, running on regular servers or routers.
2 Overview

The Early 1980s
It started in the early 1980s with voice messaging, Interactive Voice Response (IVR), which allows computer programs to interact with humans via voice and dual tone multiple frequency (DTMF) tone keypad input, and e-mail integration. This was the time before the cell phone boom. A time before voice mail systems with e-mail connectivity and “e-mail reading” capabilities were recognised as important mechanisms enabling mobile workers to access corporate information.[6]

The Mid 1990s
Unified Messaging (UM) rose in popularity in the 1990s. Research, development and financial investment were all focused on bringing voice and email communications to office and mobile telephones, and to computer screens. Some of the early providers were Sametime (later bought out by IBM), and Microsoft, who partnered up with a division of VMX/Octel/vMail to become Octel Unified Messaging in 1997. This was a precursor to Microsoft’s own UM offering, Exchange.

For the first time, e-mail and voice messaging were brought together in a single entity to computer screens. Call control features were introduced, resulting in the first Unified Communication (UC) and softphone computer client. At this time IP telephony with software based PBX systems and IP based handset technology matured, using the same IP network infrastructure without the need for dedicated telephone lines. One example of this development was the acquisition of Selsius Systems by Cisco Systems in October 1998, which provided the foundation of Cisco’s software based PBX system Cisco Unified Communications Manager - CUCM.[7]

The significant video conferencing standard, H.323 was first published by the ITU in November 1996[8]. The final UC event of note during this period was the introduction in 1999 of the RIM BlackBerry product. To start with, it was just a portable e-mail reader, but this introduced a significant move forward in communications and UC.[6]

Growth in the 2000s
In the early 2000s many start-up companies came up with new solutions combining Unified Messaging (UM) and Unified Communication (UC) into new products, which provided an integrated experience for the end user. Some of these start-up companies have since been acquired by big market players such as Cisco Systems, 3Com, Nortel Networks and others. The adoption rate by the market was slow, due to the long investment cycles for classical communication systems such as telephony. These new solutions created some pressure on the established Private Branch Exchange (PBX) vendors to provide similar functionality. Vendors such as Siemens (OpenScape), Avaya (one-X) and Mitel (Your Assistant). In the early 2000s the Internet Engineering Task Force (IETF) published RFC 2543 SIP Session Initiation Protocol[9] (1999) obsoleted by RFC 3261 SIP[10] in June 2002, which established an additional standard for signalling that was adopted fast by the community and the market (see also chapter 4 on page 25).
2.2 The History of UC

At the same time e-mail and desktop office software package vendors such as IBM Lotus Sametime and Microsoft Live Communication Server (Office Communications Server - OCS) began to have a presence on enterprise desktops and provided services such as Instant Messaging (IM) that could show the availability of the user, such as ‘Away’ or ‘Busy’. These services quickly grew in popularity. These interfaces also provided a method of integration with call control links “click-to-communicate/collaborate” to established PBX systems.[6]

2010 and Beyond

The global system integrator, Dimension Data, sponsored a study conducted by Ovum in 2013, which identified several interesting trends and observations throughout the UC industry[11]:

- Based on the Ovum study feedback regarding the technology used and the future plans for technology, the highest deployment rate for a UC component is IP telephony (IPT) with 75 %. The next highest deployment rate is Instant Messaging (IM)/Presence with 66 % and Audio/Web conferencing with a 62 % deployment rate. This is a strong indicator that those UC components are used to a certain level within the organisations, but are not linked or integrated significantly together (see figure 2.1 on the next page). Another observation from the author of this document, is the deployment rate for Microsoft Lync voice/video (now called Skype for Business) over the last 2-3 years, which has increased significantly.

- The next observation of the report shows a near majority (between 46 - 54 %) of UC component deployments are based on the premises, but are managed by a third party. The second biggest group are those based on premises that are managed internally (between 29 - 38 %) (see figure 2.2 on page 7). In the near future the on premises managed deployments will still be in the majority, but emerging “cloud based” or hosted solutions are entering the arena and offer attractive services such as Video as a Service (VaaS) or UC as a Service (UCaaS). These hosted offerings can range from simple web collaboration to multipoint high definition quality video & audio, endpoint management and meeting scheduling. This approach doesn’t require the installation of costly infrastructure on premises, but does have the disadvantage of limited features, for example interoperability between different UC environments or recording of a meeting. There are also options available that combine on premises and hosted solutions.
Figure 2.1: Ovum 2013 Study - Technology and Features[11]
2.2 The History of UC

Figure 2.2: Ovum 2013 Study - Managed UC[11]
• The security aspect was reflected in the trust of the suppliers, operating, implementing, or hosting of a specific technology (see figure 2.3). Service providers and Telcos are trusted most for fixed telephone systems (PBX). Vendors of the technology show a good “trust level” across all technologies.

Figure 2.3: Ovum 2013 Study - Supplier Trust [11]

• In general the majority of organisations have a budget for UC and a strategy, but it’s also important for decision makers to measure the investment e.g. UC adoption rate of the end users, to justify these investments. The success rate is also significantly influenced by the contributions and feedback of the end users. If this is ignored, the acceptance of UC will be reduced. Users expect the technology to match their requirements, to be flexible, and to support them.

• The integration of a BYOD strategy, (bring your own device - employee owned devices) into the overall organisation mobility UC strategy is important to increase the level of acceptance of the users. Users want to access and use UC systems provided by the organisation on their personal devices such as laptops, smart phones and tablets, which creates several challenges with regards to managing and maintaining security, a fact that can’t be ignored.

Market Trend

The market data compiled by Wainhouse Research in the quarterly report Group Video Conferencing Endpoint and Infrastructure Market Statistics Q3-2015[12] identified several trends:

• There is a moderate growth of this industry sector (Video Endpoints and Infrastructure) with approximately 100,000 endpoint units sold in Q3 2015 and growing( 2.4 on the facing page).
2.2 The History of UC

- The on premises infrastructure is declining and there is an increasing trend in transition to cloud-based solutions.
- The technology change from transcoding to media switching introduces lower costs and can be built within software on industry standard servers.
- There is a trend towards software clients and low cost single codec endpoints/room systems.

The short extract above, showing the market development, confirms the findings in the Ovum report and a steady growth of this industry with shifting focus to reduce costs and use cloud/hosted services. There are many reports with similar content available, and so it’s important to verify what kind of UC market “components” are covered. The Wainhouse report covers video infrastructure and endpoints. Other reports focus
on VoIP call control and handsets or UC Software clients e.g. Microsoft Skype for Business, Cisco Jabber and IBM Sametime.

**Interoperability**
In the preceding sections several organisations and companies are mentioned as having an involvement in the history of UC. It is obvious that most of them have different interests and goals for their involvement, which creates some issues with the interoperability of those technologies and products offered. The “simple” expectation of a UC end user or customer is: “It should work as easily as a telephone call.” This is quite a challenge. Some problems could be, for example: no video/audio received or poor quality video/audio received. No connection at all, caused by signalling or capability exchange issues, content channels that can’t be established and many more. For that reason, in May 2010, some vendors and organisations collectively formed the **Unified Communication Interoperability Forum (UCIF)**. The original founding members were: HP, Juniper Networks, Logitech, Lifesize, Polycom and Microsoft. In June 2014 the UCIF merged with International Multimedia Telecommunication Consortium (IMTC)[13] to combine their efforts to improve the interoperability and education surrounding UC.

### 2.3 The UC Promise
Unified Communication promises to enable people to connect, communicate and collaborate seamlessly to improve business agility and results. These results include better user and group productivity, dynamic collaboration and simplified business processes, with the goal of increasing revenues, decreasing costs and improving customer service. A characteristic UC based conversation could start by verifying the availability of the communication partner, based on the presence information. If the partner is available, an instant messaging (IM) chat session would be initiated between the two parties. In case the chat capabilities are not sufficient for the communication, a video conference or content sharing session, or both, could be established/added to the existing UC session. If other parties are required to join, they could be brought into the UC session as needed. There are several elements associated with UC, below is a selection of the most common components:[5]

**Call Control**
Call Control is mostly considered as a UC enabler, by providing signalling (control) for VoIP and/or video conferencing services and connecting entities together e.g. software based clients (soft-phones) on a PC with a headset, IP telephones (handsets), room based video conference systems (endpoints) or different UC systems. The control element is a fundamental building block for a UC solution and is generally based on dial rules. These rules could be based for example on a dialling prefix, SIP domain, class of service, location, available bandwidth or endpoint type. The most used signalling protocols are SIP[10] and H.323[8]. The trend to move from circuit switched based PBX over to software based IP based PBX systems to hosted offerings (on premises or in the “cloud”) demonstrates a permanent change in this sector.
2.3 The UC Promise

Voice over IP

Voice over IP (VoIP) is implemented by a set of protocols and technologies to carry voice communication over the public Internet or private networks instead of using the public switched telephone network (PSTN). VoIP is the encoding of the analogue voice signals into digital encoded IP packets, the IP packets are then transported via packet-switched networks instead of via circuit-switched networks. At the receiving end, the digital IP packets are decoded back into analogous audio signals and played back. There are several benefits to using this technology e.g. utilisation of existing wide area network (WAN) connections, least-cost-call routing, flexibility of routed IP packets and integration into UC systems. After years of adoption the main signalling protocol used for VoIP is SIP and is now well established. There are several requirements on the network that are necessary to make VoIP work and be accepted by the user; sufficient network bandwidth, constant (low variation) and low jitter (IP packet delay variation), low round trip time (RTT) and quality of service (QoS) requirements for network congestion management.

Presence

The presence or status indicator of a user and his communication availability such as: available, in call, away, do-no-disturb or off line are very important for communication partners, this may be a human, an automatic process or an application. Presence information is often the starting point for a communication. One problem surrounding presence is the federation (trust relationship) between different UC islands and the security around that. For example it is beneficial, that someone on the user’s buddy list (grouping of regularly used communication partners) can see the status information, but not someone else online who is not known to the user.

Instant Messaging

Another important UC building block is the Instant Messaging (IM) text chat service, which is often enhanced with the addition of file transfer or recording/history logging capabilities (e.g. for legal requirements in the finance industry). IM systems usually include a presence indicator system, allowing the user to know if their communication partner can or will answer. Federation (trust between systems under different administration) can cause security issues surrounding file transfer or exposing presence information, which needs to be addressed. For presence and IM, the Extensible Messaging and Presence Protocol (XMPP) is very popular[14, 15, 16] and covered in more detail in section 4.3 on page 41.

Video Conferencing

Video Conferencing (VC) allows a two-way video & audio communication between two locations (point to point/P2P) direct or between multiple locations (point to multi-point/P2M) with a Multipoint Control Unit (MCU), which interconnects/mixes all the calls together, and creates a continuous presence of the participants including audio. There are different protocols available e.g. H.323 and SIP signalling and codecs for coding and decoding of audio, video e.g. H.264/AVC High Profile, G.729 including the provisioning of a content channel. Please see chapter 3 on page 15 for more information.
Collaboration
Under the term “collaboration” fall many different methods and technologies and many people understand something different by the term, depending on their experience and expectations of UC. During an ongoing UC session video, audio is perhaps already established and an additional content channel for collaboration is added. For classical video conference systems using SIP for signalling, the binary floor control protocol (BFCP) RFC 4582[17] is commonly used to feed a media source e.g. PC screen via VGA into a video conference call. The other participants in the conference can see the screen, but it’s under the full control of the participant who is “pushing” the content.

Another concept is desktop sharing, where a participant joining the UC session, for example via a web browser with a plug-in or a dedicated desktop sharing application, is sharing his PC desktop. All other UC session participants need to have the same application or join the same web portal for sharing. This content channel is separate from the existing video conference communication and runs in parallel. The person sharing his desktop can give control to other participants and they can operate the shared PC screen. The UC session participants need to use two different technologies: one for the video/audio communication in a meeting room and one for the desktop sharing.

Content Collaboration drives the desktop sharing concept one step further and allows UC participants to share and work on the same document simultaneously. For example a spreadsheet or text file can be modified and multiple cursors or mouse pointers in different colours with the name of the participants will be shown on the screen and the content can be modified at the same time by different people. Similar to desktop sharing, an additional technology for adding audio/video needs to be utilised by the users for the collaboration. Some solutions include audio and video into the collaboration session by having a web-cam and microphone on the PC, but this works only for a single person in front of a computer and not multiple people in a room. One example of this approach is Microsoft Skype for Business (SfB) using Microsoft’s proprietary Remote Desktop Protocol (RDP). The SfB client combines video, audio, presence, IM and content in one client, and runs on a desktop.

Other trends in this area provide multiple streams of content (e.g. showing a presentation on one screen, on the next screen a spreadsheet, and having video on a third screen), touch interfaces, similar to that used by tablets, on the screen, for collaboration, and BYOD approaches.

Mobility
With the significant distribution of mobile devices such as smart phones and tablets, the users expect the same UC features on their mobile device as in their office environment e.g. audio, video, presence, same dial in approach, IM and content sharing. Support for mobility is becoming increasingly important with a shift to a mobile centric work style.

Business Process Integration
Another important enabler of UC is the reduction of response times for business processes
by contacting the right individual to get the information required in a shorter period of time. Early adopters are around, Enterprise Resource Planning (ERP) to reduce production or development cycles, human resource (HR) departments for job interviews, help desk to get connected with the next available and the right support person. The trend is to have the UC client directly embedded into a web based interface connected to a business application instead of having additional software installed (WebRTC).

**Recording, Streaming & Content Management**

To give the end user the capabilities to record and/or stream a UC session for later use is very powerful, for example for training purposes and also for quality control and auditing purposes. In some industries it may also be a legal requirement such as in the finance sector. The recordings can be tagged and searched by other users to find the right content and played back on demand.

The above list of elements provide an overview as to how complex the topic of UC can become, based on an organisation’s requirements. It’s also important to understand that UC is not a single software or hardware product which can be purchased off the shelf. Usually there are multiple vendors and many different technologies involved, which makes this task rather a complicated one.
3 Video Conferencing

3.1 Endpoints

Hardware Based Clients

To enable a meeting room for Video Conferencing (VC), the most important component is the video conferencing system (endpoint) or CODEC (COder-DECoder). The endpoint is available in different physical formats and brings all required audio/video components of the room together. Most endpoints come with LAN, multiple camera (HDCI, ...), VGA, microphone input and multiple video output connections (HDMI, YPbPr, RGB etc.) on board. The choice of endpoint depends on the room layout and what components need to be integrated e.g. number of monitors, touch screens, microphones etc. Other auxiliary connections are also possible such as USB or serial connections. The endpoint negotiates with the other end via the required communication sessions using SIP[10] for session management and Session Description Protocol (SDP)[18] for the multimedia session negotiation, which could be an endpoint or an MCU, taking the video and audio signals, compressing them into Real-time Transport Protocol (RTP)[19] format, using, for example, Advanced Video Coding (AVC) H.264 high profile, transmitting the RTP packages via the network, decompressing incoming encoded video and audio signals, and taking care of the correct playback. More information about the used protocols and their interaction will be covered in chapter 4 on page 25. The endpoints use special build hardware to support the CPU intensive coding/decoding process. Figure 3.1 shows two example endpoints from popular video conferencing manufacturers Polycom Group Series 500 with camera, remote control, microphone and from Cisco Systems C60, with connection options.

(a) Polycom Group Series 500  (b) Cisco Systems C60 (rear view)

Figure 3.1: Video Conferencing Endpoints (Examples)
Software Based Clients

On mobile devices or desktops, software clients are used, which are usually additionally installed applications or in the case of WebRTC, running in the web browser natively without the need for a plug in. They come with different feature sets, but usually include the following functionality; video/audio (via the built in camera, microphone and screen), Conference Control Buttons (e.g. stop call, video/audio mute, dial pad), Presence and Instant Messaging. Some software clients provide additional features such as content sharing, file transfer options or address book features. Figure 3.2 shows an example of a software client from Polycom. There are of course other clients available, such as Cisco Systems Jabber, Microsoft Lync, IBM Sametime, or Siemens OpenScape UC Desktop Client depending on the vendor infrastructure used.

Figure 3.2: Polycom RealPresence Mobile iPad - Software Client

3.2 Video Conferencing Specific Infrastructure

Video Conferencing and UC can be seen as another combination of data and protocols utilising IP networks, but with some specific infrastructure components and requirements to ensure the expected quality using such a solution. The most important points are highlighted in this section to the level required for the testing and analysis for this dissertation. More detailed information is provided via the references highlighted.

Network Requirements

One of the most challenging prerequisites for VC are the requirements on the underlying network infrastructure to ensure the “correct” delivery of the real-time media and
signalling packets. The bandwidth requirements depend on the used technology and CODEC. Some example bandwidth requirements are listed in Table 3.1. The most important network influencers for VC quality experience are listed below. The values shown below could differ slightly, depending on the recommendations by different vendors:

- **Packet Loss:** should be no more than 1 percent
- **Jitter:** should be no more than 40 ms, the packet delay variation can be addressed to a certain level, by using buffers
- **One-way latency:** should be no more than 100 ms

<table>
<thead>
<tr>
<th>Resolution/Frame Rate</th>
<th>H.264 Baseline Profile</th>
<th>H.264 High Profile</th>
<th>Skype for Business H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td>4CIF30</td>
<td>256 kbps</td>
<td>128 kbps</td>
<td>N/A</td>
</tr>
<tr>
<td>720p30</td>
<td>1024 kbps</td>
<td>512 kbps</td>
<td>2500 kbps</td>
</tr>
<tr>
<td>1080p30</td>
<td>2048 kbps</td>
<td>1024 kbps</td>
<td>4000 kbps</td>
</tr>
</tbody>
</table>

Table 3.1: VC Bandwidth Requirement Examples[21, 22]

There are several techniques built into TCP/IP such as sliding window and slow start but those are not sufficient to guarantee the network quality end-to-end to the particular level required for VC. The TCP mechanisms will not apply, because most of the time, real time traffic such as RTP and voice/video payload is carried by UDP. In real world scenarios, a proper Quality of Service (QoS) strategy needs to be implemented to limit the quality impact on voice and video for the end user. A very common approach is the classification of the traffic (media and signalling) by the VC application or endpoint by tagging the packets with DiffServ - DSCP at the IP header. The classified and tagged packets are interpreted on a per hop basis by the network devices (e.g. routers) during the forwarding process and congestion control mechanisms are enforced, for example, by queuing or even dropping of specific traffic by the forwarding network device. The main problem is that regarding the enforcement, when the VC traffic is passing through networks with different administration domains e.g. two different Multiprotocol Label Switching (MPLS) Wide Area Network (WAN) service providers. It’s not guaranteed that both providers enforce the same QoS policy and may ignore or rewrite the tagging of the packets. The recommended VC DSCP QoS settings are defined in Configuration Guidelines for DiffServ Service Classes - RFC 4594[23] to tag on the IP layer (OSI Layer 3 - Network Layer):

<table>
<thead>
<tr>
<th>Application</th>
<th>DSCP</th>
<th>Binary Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Signaling</td>
<td>CS5</td>
<td>101000</td>
</tr>
<tr>
<td>Audio</td>
<td>EF</td>
<td>101110</td>
</tr>
<tr>
<td>Video</td>
<td>AF41</td>
<td>100010</td>
</tr>
<tr>
<td>Text</td>
<td>EF</td>
<td>101110</td>
</tr>
</tbody>
</table>

Table 3.2: Recommended VC DSCP QoS settings[23]

Another approach is the usage of Resource Reservation Protocol (RSVP), which reserves
resources on the network path. In real deployments, RSVP is not used very often, because of the additional complexity and the fact that it needs to be supported by all network components. For further reference on QoS please refer to [24, 25].

**Call Control, Conference Management & Multipoint Control Unit (MCU)**
The capabilities of the Call Control and Conference Management VC components heavily depends on the focus of the individual vendor or open source project. Some of the solutions have focus on Voice over IP (VoIP)/IP Telephony and include some VC specific features and some have more of a VC background and have added some VoIP features. The Call Control and Conference Management components should provide a reliable and scalable distribution of voice and video calls across multiple media server MCUs (Multipoint Control Units) and provide the following common features:

- H.323 Gatekeeper
- SIP registrar and proxy server
- H.323 ↔ SIP transition gateway
- Dial plan and prefix services
- Device authentication
- Bandwidth management
- *Virtual Meeting Rooms (VMRs)* for ad-hoc conferencing

The Multipoint Control Units (MCUs) are used for multipoint conferences (bridge conferences) one in which multiple endpoints are connected, with all participants able to see and hear each other. If required, the MCU handles the transcoding or interworking between different audio/video formats, media or protocols (IP, ISDN, H.323, SIP, etc.). The transcoding process is very CPU intensive and for that reason is often realised in hardware based on purpose built *digital signalling processors - DSPs*. There is currently a trend to do this transcoding with standard x86 CPUs in virtual machines and Cloud Services to remove the hardware dependency, improve scale and provide more flexibility. The conference management is often closely connected to the call control component. Cisco TelePresence Video Communication Server (VCS)[26], Cisco TelePresence Conductor[27], Cisco Unified Communications Manager (CallManager - CUCM)[28], Polycom DMA[29] and Polycom Collaboration Server - RMX[30] are some industry examples.

### 3.3 Call Types, Video Protocols and Formats

The key points for a successful VC connection depends on the signalling protocol used to establish a call, the used CODEC and the packet transmit scheme used for sending the real time traffic across the network. Some very popular and widely supported coding standards are for example H.264 or MPEG-4 Part 10 and Advanced Video Coding (MPEG-4 AVC). There are three main call types for VC. These are:
Point to Point (P2P)
A VC point to point call (P2P) is the simplest call scenario and can be formed between two endpoints directly without registration on a call control infrastructure. This is implemented by dialling the other participants IP address, which can be stored in a local address book to make the system simpler for the end user, or with registration at a SIP registrar, the SIP URI can be used. After the endpoints have negotiated the video/audio coding standard, compression, video resolutions and call speed, the call is established and the real time traffic is exchanged directly between them. See Figure 3.3a.

Point to Multipoint via MCU (P2M) with Advanced Video Coding (AVC)
If there are more than two participants, a Multipoint Control Unit (MCU) is required and the endpoints negotiate the individual connections with the MCU, similar to a P2P call. The MCU receives the video and audio from each endpoint, encodes the data, composites the videos into a layout e.g. 4x4 or 2x2 and sends the individually encoded stream back to each endpoint participating in the same conference. This transcoding process is often very CPU intensive. It is conducted by special DSP hardware and introduces a delay of around 80 - 200 ms into the conference’s call, which can be noticeable for the end user. See Figure 3.3b for the signalling and media traffic flow including a SIP signalling server. During this process the MCU takes care of the transcoding between different individual endpoints e.g. call rate, video/audio codec, connection type, resolution and signalling protocol. The receiving layout for each endpoint is created by the MCU. Figure 3.4 on the following page shows an example AVC built layout. More information about AVC P2M calls can be found at [30, 31]
Point to Multipoint with Scalable Video Coding (SVC)

The Scalable Video Coding (SVC) video protocol is based on the Annex G extension of the H.264/MPEG-4 AVC standard, ratified by the ITU in 2007 and provides video without the need for transcoding of an MCU, hence it requires less CPU resources while providing better resiliency and lower latency. Compared with AVC described in P2M AVC, each SVC endpoint transmits multiple video bit streams at different resolutions, frame and line rates called simulcasts such as 720p at 30 fps, 15fps and 7.5 fps, 360p at 15 fps and 7.5 fps, and 180p at 7.5 fps. Each individual receiving endpoint composes the layout according to their settings and capabilities. The MCU or SVC switch simply forwards individual streams to specific endpoints requesting them via the SIP signalling channel similar to Figure 3.3b on the previous page, the MCU switching the streams instead of transcoding them. The introduced latency is about 20 ms vs. 80 - 200 ms with AVC and provides a better end user experience. There are more specific SVC details regarding enhancement layers and modes of SVC defined by the UCI Forum (Unified Communications Interoperability Forum)[13], in this document UCIF Mode 1 is described and used by Polycom’s SVC implementation. More detailed information about SVC is covered in [31, 32].
3.4 WebRTC

With the trend towards technologies such as SVC, Web Real-Time Communications (WebRTC) is the next step of evolution to provide UC functionality directly in web browsers without the need to install any additional software or browser plug-ins. The standardisation efforts are combined by the World Wide Web Consortium (W3C) together with the IETF. The WebRTC standard enables web developers to use HTML5 together with JavaScript API (Application Programming Interface) calls to build UC capabilities directly into websites and browsers across different computing platforms.

In Figure 3.6 on the following page the browser model with a simple WebRTC call flow is illustrated. The application is transferred from the web server to the browser using standard HTTP/S or WebSockets[33] and transmits the JavaScript and HTML code. The JavaScript web application is executed locally in the browser and uses APIs with RTC functions to communicate with the computer operating system. The signalling server provides the signalling channel between all the browser participants for the peer to peer communication. The media channel is established directly between the browsers and usually forms a full mesh communication topology, which introduces some bandwidth concerns for multipoint calls with several participants. For firewall transversal, ICE is used, and this is explained in more detail in 4.6.4 on page 58. More information on WebRTC and the implementation can be found in WebRTC : APIs and RTCWEB protocols of the HTML5 real-time web[34] and Polycom RealPresence Web Suite[35].
Figure 3.6: WebRTC Simple Call Model[35]
3.5 Conclusion

We have covered in the first chapters of this document the past and the present, and we have taken a glimpse into the future of UC as an introduction, to create an awareness of the complexity which is involved. It’s important to understand the challenges surrounding real time traffic and especially the traversal through firewalls. In the next chapter we take a closer look at the involved protocols and technologies to address the firewall traversal problem.
4 Protocols and Related Standards

4.1 Overview

In the previous sections, protocol names and functions have been mentioned several times. This chapter goes into a bit more detail regarding which protocols are relevant for UC and this dissertation, their function, and their interaction with each other. Figure 4.1 provides an overview of UC related protocols, the standards, landscape, and the relationship with the Internet Model[36].

![Common UC Protocol Landscape mapped to the Internet Model](image)

Figure 4.1: Common UC Protocol Landscape mapped to the Internet Model

In the list below are some references for the protocols above that are not covered in detail in this document:

- Stream Control Transmission Protocol (SCTP) - RFC3257, RFC4168[37, 38]
- Real Time Protocol (RTP) - RFC3550[19]
- Real Time Streaming Protocol (RTSP) - RFC2326[39]
- Secure Real Time Transport Protocol (SRTP) - RFC3711[40]

4.2 SIP - Session Initiation Protocol

One of the most popular signalling protocols for UC is the *Session Initiation Protocol* (SIP) defined in the latest version in RFC 3261[10]. For video conferencing H.323[8] is still very common, but will be replaced in many cases by SIP. One of the main reasons
for the SIP popularity may come from the wide distribution of VoIP and the movement from telephony to the UC/VC space. Another advantage of SIP signalling, is that it is text based and does not use ASN.1 encoding like H.323, and its flexibility of protocol usage. It can use UDP, TCP, or SCTP. UDP can be used to decrease the protocol overhead, increase speed and efficiency. TCP can be used if TLS is used for security. More recent implementations may use SCTP, which offers better resistance against DoS attacks by using a four-way handshake process.

Several protocols have been designed around real-time multimedia session data such as voice, video, or text messages, and SIP works together with these protocols by enabling endpoints (user agents - UAs) to discover one another and to agree on a characterization of a session they would like to share. A SIP user agent must support both a client (UAC) and server (UAS) application. The UAC initiates the communication and the UAC response.

SIP allows a network host infrastructure (proxy servers) to be created. Through this infrastructure, registrations, session invitations and other requests can be sent. An alternative to the proxy server is the redirect server which can be used to respond to, but not forward UA requests. The redirect server uses a location service or local database to lookup a SIP user and send the target user location information back to the caller. The caller UA directly contacts the called UA with the provided location information from the redirect server.

SIP is a general-purpose tool for creating, modifying and terminating sessions that works independently of underlying transport protocols and without dependency on the type of session that is being established. SIP supports the following main functions for media session handling[10]:

- User location: Choosing an end system to use for the communication.
- User availability: Checking to see if the end user is available and willing to join the communication.
- User capabilities: Choosing the media to use, and the parameters of that media.
- Session setup: establishing the call at both ends.
- Session management: This may include transfer of the session, modifying the session or ending the session. Starting other services may also be an aspect of session management.

SIP is a component that can be used with other IETF protocols such as Real-time Transport Protocol (RTP) for transporting real-time data and providing QoS feedback or Session Description Protocol (SDP) for describing multimedia sessions. This does not provide services itself, instead it provides primitives that can be used to implement different services.[10]
4.2 SIP - Session Initiation Protocol

4.2.1 Point to Point Call (P2P)

Figure 4.2 demonstrates the call flow of a SIP based P2P call between two user agents (endpoints) in the Test Environment 6.1 on page 77, roomsystem3 (192.168.1.153) calling roomsystem1 (192.168.1.151). Both endpoints are not registered on a SIP proxy and roomsystem3 is using the username + IP address (SIP URI) sip:roomsystem1@192.168.1.151 as a dial string.

Figure 4.2: Basic SIP P2P call setup and tear down

**SIP INVITE**

The calling party roomsystem3, begins the message exchange by sending a SIP INVITE message to the called party, roomsystem1. In the INVITE the details of the type of session are included. The session could be a VoIP call with audio only, a gaming session, or in this case a videoconferencing session. The INVITE message will include the following data (Figure 4.3 on the next page) and is an example of a SIP request message defined in the core SIP RFC 3261[10] with others defined in extension RFCs.
INVITE sip:roomsystem1@192.168.1.151 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.153:5060;branch=z9hG4bK2760909730−821
Max-Forwards: 70
Allow: INVITE, BYE, CANCEL, ACK, INFO, PRACK, COMET, OPTIONS, SUBSCRIBE, NOTIFY, MESSAGE,
REFER, REGISTER, UPDATE
From: roomsystem3 <sip:roomsystem3@mobile−lab.local>; tag=plcm_2760909783−821; epid =8213100FDF6ACV
To: <sip:roomsystem1@192.168.1.151>
Call-ID: 2760909413−821
CSeq: 1 INVITE
Session−Expires: 90
Supported: replaces, ms−dialog−route−set−update, ms−forking, timer
Contact: roomsystem2 <sip:roomsystem2@192.168.1.153:5060;transport=udp>;proxy=replace;+sip.instance="<urn:uuid:d474c94c−baff−552d−a565−bbf321e35d>"
User-Agent: PolycomRealPresenceGroup500 /5.0.0
MRD:MRE; MRC−V=1.0.1
P_PREFERRED.ENTITY:roomsystem2@mobile−lab.local
Content-Type: application/sdp
Content-Length: 1533
v=0
o=gs5002GroupSeries 713043897 0 IN IP4 192.168.1.153
s=MRED MRC−V=1.0.1
c=IN IP4 192.168.1.153
b=AS:512
r=0
m=audio 16478 RTP/AVP 115 102 9 15 0 8 18 101
a=rtpmap:115 G7221/32000
a=fmtp:115 bitrate=48000
a=rtpmap:102 G7221/16000
/*/ further audio and video maps are omitted

Figure 4.3: SIP INVITE message

SIP uses text-encoded messages and the example above is what the SIP message looks like on the UDP datagram or TCP flow. The fields in the INVITE message are called header fields and have the form **Header: Value** CRLF. The first line of the request message lists the method, which is **INVITE**, the Request-URI, which indicates the resource to which the request is being sent, then the SIP version number 2.0, all separated by spaces. The line is terminated by a CRLF (Carriage Return Line Feed).

Uniform Resource Locators (URLs) are an addressing scheme developed for the World Wide Web (WWW), defined in RFC1738[41], and have the following general form:

```
<scheme>:<scheme-specific-part>
```

For example `https://www.royalholloway.ac.uk/isg/home.aspx` `http` defines the scheme or protocol to be used and after the “:” `www.royalholloway.ac.uk` resolves to a domain name, via DNS, pointing to an IP address, and a file name `/isg/home.aspx`. URLs can include additional protocol parameters relating to the transport e.g. `ssh:host1.example.com:22` indicates that secure shell should be used to connect to `host1.example.com` using port 22. Most protocols use URLs, but SIP uses **Uniform Resource Indicators** (URIs) due to the mobility aspect which means a particular address (URI) is not tied to a single physical device, instead it is a logical entity. For secure TLS SIP the `sips:` scheme is used. The `userinfo` part of SIP and SIPS URIs is case sensitive.
4.2 SIP - Session Initiation Protocol

roomsystem1@mobile-lab.local is different to RoomSystem1@mobile-lab.local. The term URL and URI are often used interchangeably. [41, 42]

Each SIP device that originates or forwards a SIP message adds its own address in a Via header field. The Via header field contains the SIP version number 2.0, a “/”, then UDP for UDP as transport protocol, a space, the hostname or IP address, a colon, then a port number e.g. the well-known SIP UDP/TCP port 5060. The branch=z9hG4bK2760909730-821 parameter is a transaction identifier. Responses can be correlated to this request because they use the same transaction id.

The next header field Max-Forwards: 70 is decremented by one at each SIP hop to limit possible loops. The default value should be 70 and must be inserted by each user agent client.

The next header fields are From: and To:, which show the originator and destination of the SIP request. SIP requests are routed by the Request-URI instead of the To: URI. The To: URI stays the same during transit but the Request-URI could be re-written. When a name label is used, as in this example, the SIP URI is enclosed in brackets ⟨⟩ and could be displayed during an alert.

The Call-ID header field is used to keep track of a particular SIP session. The originator creates a locally unique string, each party in the session also contributes a random identifier, unique to each call. These identifiers are called tags, and are included in the To and From header fields as the session is established. The initiator of the session that generates the establishing INVITE generates the unique Call-ID and From tag. In the response to the INVITE, the user agent answering the request will generate the To tag. The combination of the local tag (From header field), remote tag (To header field), and the Call-ID uniquely identifies the established session, known as a dialogue. This dialogue identifier is used by both parties to identify the call, because there could be multiple calls setup between them. Subsequent requests within the established session will use this dialogue identifier.

The CSeq (Command Sequence) header field contains a number, followed by the method name, in this example INVITE. This number is incremented for each new request sent.

The minimum required header fields in any SIP request are: Via, Max-Forwards, To, From, Call-ID, and CSeq. Other header fields can be included as optional additional information, or information for specific request types. A Contact header field is required in this INVITE message, which contains the SIP URI of the roomsystem2 communication devices, known as user agent (UA). This URI can be used to send messages directly to the device.

The Content-Type and Content-Length header fields indicates that the message body is Session Description Protocol (SDP)[18] and the length is 1533 octets. All lines after
the Content-Length field are SDP data describing the media attributes that the caller
roomsysterm3 desires for the call. The media information is required by the endpoints
to negotiate the attributes.[41, 42]

Session Description Protocol (SDP)
During the initiation of a multimedia session, certain data must be transferred between
the parties involved. The information transferred includes transport addresses and other
session description data. SDP is a standard format for session data and is separate from
the transport protocols which are utilised as required. These may include the Session
Announcement Protocol[43], Session Initiation Protocol [10], Real Time Streaming Pro-
tocol, electronic mail using the MIME extensions and the Hypertext Transport Protocol.
SDP is not intended to support negotiation of session content or media encodings.[18]

The SIP messages used to create sessions carry session descriptions that allow partic-
ipants to agree on a set of compatible media types. These session descriptions are com-
monly formatted using SDP. When used with SIP, the offer/answer model [44] provides
a limited framework for negotiation using SDP, which is not covered in this document.

The Real Time Streaming Protocol (RTSP) [39] and Real Time Protocol (RTP)[19],
are application-level protocols for control over the delivery of data with real-time prop-
ties. RTSP/RTP provides an extensible framework to enable controlled, on-demand
delivery of real-time data, such as audio and video. An RTSP or RTP client and server
negotiate an appropriate set of parameters for media delivery, using SDP syntax to de-
scribe parameters such as media type, codec or bit rate for the media session IP address
and port information. The fields in the SDP section have to follow a certain order to
allow easier error detection and parsing. The mandatory fields are listed in Figure 4.4
on the facing page, line items marked with an * are optional.

<table>
<thead>
<tr>
<th>SDP Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v=0</td>
<td>Version number</td>
</tr>
<tr>
<td>o=gs5002GroupSeries 713043897 0 IN IP4 192.168.1.153</td>
<td>Origin</td>
</tr>
<tr>
<td>s=MRD=MRE MRC-V=1.0.1</td>
<td>Session Name</td>
</tr>
<tr>
<td>c=IN IP4 192.168.1.153</td>
<td>Connection</td>
</tr>
<tr>
<td>b=AS:512</td>
<td>Bandwidth information</td>
</tr>
<tr>
<td>t=0 0</td>
<td>Time</td>
</tr>
<tr>
<td>m=audio 16478 RTP/AVP 115 102 9 15 0 8 18 101</td>
<td>Media</td>
</tr>
<tr>
<td>a=rtpmap:115 G7221/32000</td>
<td>Attributes</td>
</tr>
<tr>
<td>a=fmtp:115 bitrate=48000</td>
<td></td>
</tr>
<tr>
<td>a=rtpmap:102 G7221/16000</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: SDP parameter example

Table 4.1 shows some example SDP parameters and attributes from the SIP INVITE
message example above (Figure 4.3 on page 28). The v= field specifies the version and
at this stage only version 0 is defined. The origin field o= uniquely identifies the session
and has the following format:

```
o=<username><sess-id><sess-version><nettype><addrtype><unicast-address>
```
4.2 SIP - Session Initiation Protocol

Session description
v= (protocol version)
o= (originator 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=*/ (zero or more bandwidth information lines)

One or more time descriptions ("t=" and "r=" lines; see below)
z=*/ (time zone adjustments)
k=*/ (encryption key)
a=*/ (zero or more session attribute lines) Zero or more media descriptions

Time description

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

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

Figure 4.4: SDP fields[18]

The <username> contains the originator’s login, host or a - for none. <sess-id> is a numeric string such that the combination of <username>, <sess-id>, <nettype>, <addrtype>, and <unicast-address> forms a globally unique identifier for the session. The standard recommends that a Network Time Protocol (NTP) format timestamp be used to ensure uniqueness [45]. <nettype> is a text string giving the type of network, "IN" is defined to have the meaning "Internet". The <addrtype> is a text string, either "IP4" or "IP6" specifying the <unicast-address> of the machine from which the session was created and it can be the fully qualified domain name or the dotteddecimal representation of the IP version 4 address of the machine. For IP6, this is either the fully qualified domain name or the compressed textual representation of the IP version 6 address of the machine.

The s= field contains the session name and there must be one defined per session description. In the event there is no meaningful name this can be a single space.

The connection data field c= has the following format:

\[c=<nettype> <addrtype> <connection-address>\]

<nettype> is a text string giving the type of network, "IN" is defined to have the meaning "Internet”. The <addrtype> is a text string either "IP4" or "IP6" specifying the host sending the media packets. This address can be a multicast or unicast address. In our
example it is a unicast address. In the case of a multicast address it has the following format:

\[
c=\langle\text{base multicast address}\rangle[/\langle\text{ttl}\rangle]/\langle\text{number of addresses}\rangle
\]

where \(\text{ttl}\) is the time to live value, and \(\langle\text{number of addresses}\rangle\) specifies how many contiguous multicast addresses are included starting with the \(\langle\text{base multicast address}\rangle\).

The optional bandwidth field \(b=\langle\text{bwtype}:\langle\text{bandwidth}\rangle\rangle\), defines the type of bandwidth either as \(\text{CT}\) for the total conference or \(\text{AS}\) for application specific. In multicast sessions, \(\text{CT}\) is used to specify the total bandwidth, which can be used by all participants. Normally, \(\text{AS}\) is the application’s “maximum bandwidth” control if applicable. For RTP-based applications, \(\text{AS}\) gives the RTP ”session bandwidth”.

The \(t=\langle\text{start-time}\rangle\ <\text{stop-time}\rangle\) specifies the start and stop time of the session and it uses NTP\[45\] timestamps. If the \(\langle\text{start-time}\rangle\) and \(\langle\text{stop-time}\rangle\) is zero for a scheduled session, it means that it is a permanent session. A zero value for \(\langle\text{stop-time}\rangle\) means there is no stop time and the session lasts forever. The optional \(r=\) field specifies the repeat times together with the optional time zone field \(z=\).

A session description can contain multiple media descriptions or none at all, and have the format: \(m=\langle\text{media}\rangle\ <\text{port}\rangle\ <\text{proto}\rangle\ <\text{fmt}\rangle\ldots <\text{fmt}\rangle\). \(\langle\text{media}\rangle\) is the media type and currently “audio”, “video”, “text”, “application”, and “message” are defined/used. The sub-field \(\langle\text{port}\rangle\) specifies the port number to be used. \(\langle\text{proto}\rangle\) specifies the transport protocol and the following are defined: “udp” indicates an unspecified protocol running over UDP. “RTP/AVP” denotes RTP\[19\] using RTP Profile for Audio and Video Conferences with Minimal Control \[46\] using UDP. “RTP/SAVP” denotes the Secure Real-time Transport Protocol\[40\] using UDP. If the \(\langle\text{proto}\rangle\) sub-field is ”RTP/AVP” or ”RTP/SAVP” the \(\langle\text{fmt}\rangle\) sub-fields contain RTP payload type numbers. In a case where a list of payload type numbers is given, this implies that all of these payload formats may be used in the session, but the first of these formats is the preferred format for the session. The list of RTP types is managed by the Internet Assigned Numbers Authority (IANA)\[47\]. Figure 4.5 on the next page shows the audio format details for the SDP example 4.1 on page 30.

The optional attribute \(a=\) field may follow a media description field \(m=\) and can be used to define additional “session-level” and/or “media-level” attributes. Attribute fields can be added before the first media field and these “session-level” attributes provide additional information, which apply as a whole for the session instead of to individual media. The two forms of the attribute field can be a property attribute \(a=\langle\text{flag}\rangle\) to specify the property of a session e.g. \(a=\text{receiveonly}\) or a value attribute in the form of \(a=\langle\text{attribute}\rangle:\langle\text{value}\rangle\) to specify for example a whiteboard orientation \(a=\text{orient:portrait}\). The SDP attributes are defined in RFC4566 \[18\] and Figure 4.6 shows examples from the SDP parameter example in Figure 4.1 on page 30. SDP attributes also provide extensions such as the Binary Floor Control Protocol (BFCP).
4.2 SIP - Session Initiation Protocol

Figure 4.5: Media format details for audio example

\[17, 48\] a=confid, a=userid, a=floorid to establish a content sharing channel in a video conference or a=fingerprint to share the fingerprint of a certificate during TLS negotiation.\[42\]

m=audio 16478 RTP/AVP 115 102 9 15 0 8 18 101
a=rtpmap:115 G7221/32000
a=fmtp:115 bitrate=48000
a=rtpmap:102 G7221/16000
a=fmtp:102 bitrate=32000
a=rtpmap:9 G722/8000
a=rtpmap:15 G728/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv

Figure 4.6: SDP attributes example

The offer/answer negotiation model for SDP with SIP defined in [44] is not covered and out of scope of this document

SIP 180 Ringing

The SIP 180 Ringing message is an example of a SIP response message and uses the first three digits for classification. The 1 indicates an informational response to transfer non-critical information about the call status. The majority of the SIP response codes are derived from HTTP version 1.1 response codes. A SIP response code of 404 Not Found, which is similar to the HTTP 404 Not Found response code, indicates that the user is unknown. Only the response code is interpreted by a server or user agent. More information about the response codes can be found in RFC3261[10] and [42].

Figure 4.7 on the following page shows the response message with most fields copied from the original example SIP INVITE message including: Via, To, From, Call-Id, and CSeq. In the first line the response start line with the SIP version number and the reason phrase Ringing is added. Noticeable is that the To: and From: header fields are
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.153:5060;branch=z9hG4bK2760909730-821
From: roomsystem3 <sip:roomsystem3@mobile-lab.local>;tag=plcm_2760909783-821;epid=8213100FDF6ACV
To: <sip:roomsystem1@192.168.1.151>;tag=plcm_2761016364-3331
Call-ID: 2760909413-821
CSeq: 1 INVITE
Supported: timer
Contact: roomsystem1 <sip:roomsystem1@192.168.1.151:5060;transport=udp>;x-cisco-multiple-screens=1
User-Agent: PolycomHDX8000HD / 3.1.8
Content-Length: 0

Figure 4.7: SIP 180 Ringing message

not reversed in the response message, this is because with SIP the two fields specify the
direction of the request and not the direction of the message:

From:roomsystem3 -> To:roomsystem1

In the To: line a tag was added, generated by roomsystem1 and both tags (From: and To: lines) are used for any further request and responses during a session. In the response message the Contact: header field carries the address at which roomsystem1 can be contacted directly after the session is established.[42]

SIP 200 OK
If the called party, in our case roomsystem1, decides to accept the call by picking up the receiver or accepting the VC call, the 200 OK response message, which is a success class response, is sent. The response message indicates, that the proposed media session is acceptable and contains the media session information in Figure 4.8 on the next page:

The 200 OK response is constructed the same way as the 180 Ringing and uses the same To: tag and Contact URI. The media information is transferred by the SDP message and contains information as shown in Table 4.2 (only the audio media stream is shown):[42]

<table>
<thead>
<tr>
<th>Endpoint IP address</th>
<th>192.168.1.153</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media format</td>
<td>audio</td>
</tr>
<tr>
<td>Port number</td>
<td>49176</td>
</tr>
<tr>
<td>Transport protocol</td>
<td>RTP udp</td>
</tr>
<tr>
<td>Media encoding</td>
<td>SIRENLPR</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>48000</td>
</tr>
</tbody>
</table>

Table 4.2: SIP 200 OK media information example
4.2 SIP - Session Initiation Protocol

SIP /2.0 200 OK
Via: SIP /2.0/UDP 192.168.1.153:5060; branch=za9hG4bK2760909730−821
From: roomsystem3 <sip:roomsystem3@mobile−lab.local>; tag=plcm_2760909783−821; epid =8213100FDF6ACV
To: <sip:roomsystem1@192.168.1.151>; tag=plcm_2761016364−3331
Call-ID: 2760909413−821
CSeq: 1 INVITE
Session−Expires: 90; refresher=uas
Supported: timer
Contact: roomsystem1 <sip:roomsystem1@192.168.1.151:5060; transport=udp>; x−cisco−multiple−screen=1
Allow: INVITE, BYE, CANCEL, ACK, INFO, PRACK, COMET, OPTIONS, SUBSCRIBE, NOTIFY, MESSAGE, REFER, REGISTER, UPDATE
User−Agent: PolycomHDX8000HD /3.1.8
Content−Type: application/sdp
Content−Length: 1500
v=0
o=roomsystem3 0 0 IN IP4 192.168.1.151
s=−
c=IN IP4 192.168.1.151
b=AS:512
m=audio 49176 RTP/AVP 118 115 102 9 15 0 8 18 106 99 101
a=rtpmap:118 SIRENLPR/48000/1
a=fmtp:118 bitrate=64000
a=rtpmap:115 G7221/32000
a=fmtp:115 bitrate=48000
/* further audio and video maps are omitted

Figure 4.8: SIP 200 OK
4 Protocols and Related Standards

SIP ACK - Acknowledge
The last step is the confirmation of the media session with the ACK acknowledgement request and means that roomsystem3, the initiator, successfully received the response from roomsystem1. The exchange for the media session is now completed and can be established, in this example by RTP.

ACK sip:roomsystem1@192.168.1.151:5060;transport=udp SIP/2.0
Via: SIP/2.0/UDP 192.168.1.153:5060;branch=z9hG4bK2760984600–821
Max-Forwards: 70
From: roomsystem2 <sip:roomsystem2@mobile-lab.local>;tag=plcm_2760909783–821;epid =8213100FDF6ACV
To: <sip:roomsystem1@192.168.1.151>;tag=plcm_2761016364–3331
Call-ID: 2760909413–821
CSeq: 1 ACK
/* additional vendor specific header information is omitted
Content-Length: 0

Figure 4.9: SIP ACK - Acknowledge example

The command sequence, CSeq: is using the same number as the INVITE, but the method changes to ACK and the media session can begin. This typically uses RTP, with the media information being carried in the SIP messages. The branch parameter in the Via: field has changed to a newer value and indicates that the ACK sent to acknowledge the 200 OK is a separate transaction.

This example message flow demonstrates that SIP is an end-to-end signalling protocol and a SIP server is not required for the protocol usage. The two endpoints running a SIP protocol stack know each other’s IP address and so can use SIP to establish a media session between them. This example also shows the client-server behaviour of SIP, when roomsystem3 initiates the call with the INVITE request, it’s acting as the SIP client and roomsystem1 responds by acting as the SIP server. After the media session is established, roomsystem3 initiates the BYE message and roomsystem1 acts as SIP server responding with OK. In the event that roomsystem1 sends the BYE message the roles would change and roomsystem1 would act as the SIP client and roomsystem3 as SIP server. This is the reason why a SIP device must support both SIP user agent client and user agent server. Compared with other client/server model protocols such as HTTP or SSH, there is always a dedicated server and client. SIP switches back and forward between the two roles.[42]

SIP BYE
The SIP BYE message in Figure 4.10 on the next page is sent by roomsystem3 to terminate the established media session.

The Via: field is populated by roomsystem3’s host address and a new transaction identifier, because BYE is a separate transaction from the INVITE or ACK transaction. The dialogue is identified by the two tags in the From: and To: fields and the Call-ID:, which are still the same as in the INVITE and OK messages. The BYE message could have been initiated by either side, roomsystem1 or roomsystem3, by changing the message direction and reversing the To: and From: fields.[42]
4.2 SIP - Session Initiation Protocol

BYE sip:roomsymt1@192.168.1.151:5060;transport=udp SIP/2.0
Via: SIP/2.0/UDP 192.168.1.153:5060;branch=z9hG4bK2783650469-821
Max-Forwards: 70
Allow: INVITE,BYE,CANCEL,ACK,INFO,PRACK,COMET,OPTIONS,SUBSCRIBE,NOTIFY,MESSAGE,
REFER,REGISTER,UPDATE
From: roomsymt2 <sip:roomsymt2@mobile-lab.local>;tag=plcm_2760909783-821;epid =8213100FDF6ACV
To: <sip:roomsymt1@192.168.1.151>;tag=plcm_2761016364-3331
Call-ID: 2760909413-821
CSeq: 4 BYE
User-Agent: PolycomRealPresenceGroup500/5.0.0
P,Preferred,Identity:roomsymt2@mobile-lab.local
Supported: timer, replaces, ms-dialog-route-set-update, ms-forking
Content-Length: 0

Figure 4.10: SIP BYE message example

SIP BYE - 200 OK

The confirmation to the previous BYE message is shown in Figure 4.11:

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.153:5060;branch=z9hG4bK2783650469-821
From: roomsymt2 <sip:roomsymt2@mobile-lab.local>;tag=plcm_2760909783-821;epid =8213100FDF6ACV
To: <sip:roomsymt1@192.168.1.151>;tag=plcm_2761016364-3331
Call-ID: 2760909413-821
CSeq: 4 BYE
Contact: roomsymt1 <sip:roomsymt1@192.168.1.151:5060;transport=udp>;x-cisco-
multiple-screen=1
Allow: INVITE,BYE,CANCEL,ACK,INFO,PRACK,COMET,OPTIONS,SUBSCRIBE,NOTIFY,MESSAGE,
REFER,REGISTER,UPDATE
User-Agent:PolycomHDX8000HD/3.1.8
Content-Length: 0

Figure 4.11: SIP BYE OK message example

The OK uses the same CSeq: 4 BYE of the original BYE request and no ACK is sent, which is required only for an INVITE message.[42]
4 Protocols and Related Standards

4.2.2 SIP Call with Proxy Server

The endpoint in the P2P SIP call scenario 4.2.1 on page 27 used the IP address for initiating the call by sending the INVITE message directly to the called endpoint. This is normally not the case, because it is not user friendly to type in the IP address like a telephone number and the IP address can often change, for example DSL or mobile ISP connections use DHCP and are dynamically assigned. The IP address has no relation to the user-to-user driven communication approach SIP uses in general. For user addressing, SIP uses URIs as described above at the SIP INVITE section (4.2.1 on page 28) and can handle telephone numbers, transport parameters and other items. The SIP URI is a name, which resolves to an IP address by using a SIP proxy server and DNS lookups during the call initiation.

![Figure 4.12: SIP P2P call with proxy server](image)

Figures 4.12 and 4.13 on the facing page show a call initiated by roomsystem3 to roomsystem1 by using a SIP proxy in the middle of the signalling communication receiving and forwarding messages. It’s possible that multiple SIP proxies may be in the signalling path. The endpoint roomsystem3 doesn’t know how to find roomsystem1@mobile-lab.local, the SIP proxy is used to route the SIP INVITE. The SIP proxy conducts a DNS lookup for the URI domain part mobile-lab.local which would provide the IP address handling the domain mobile-lab.local, more details are provided in 4.2.3 on page 41.

In this example, the domain part of the URI of both endpoints is the same (mobile-lab.local) and both endpoints are registered to the same SIP proxy/registrar. In this case a DNS lookup is not required, instead the proxy queries the registrar database for the URI.
<table>
<thead>
<tr>
<th>roomsystem3</th>
<th>192.168.1.153</th>
<th>SIP Proxy</th>
<th>192.168.1.92</th>
<th>roomsystem1</th>
<th>192.168.1.151</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request: INVITE</td>
<td></td>
<td>SIP/SDP: Request: INVITE</td>
<td>sip:<a href="mailto:roomsystem1@mobile-lab.local">roomsystem1@mobile-lab.local</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Request: INVITE</td>
<td>SIP/SDP: Request: INVITE</td>
<td>sip:roomsystem1@192.168.1.151:5060;udp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status: 180 Ringing</td>
<td>SIP: Status: 180 Ringing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status: 200 OK</td>
<td>SIP/SDP: Status: 200 OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request: ACK</td>
<td>SIP: Request: ACK</td>
<td>sip:roomsystem1@192.168.1.92:5060;udp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request: ACK</td>
<td>SIP: Request: ACK</td>
<td>sip:roomsystem1@192.168.1.151:5060;udp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;-Media Session between endpoints-&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request: BYE</td>
<td>SIP: Request: BYE</td>
<td>sip:192.168.1.92:5060;udp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request: BYE</td>
<td>SIP: Request: BYE</td>
<td>sip:roomsystem3@192.168.1.153:5060;udp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status: 200 OK</td>
<td>SIP: Status: 200 OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status: 200 OK</td>
<td>SIP: Status: 200 OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.13: SIP P2P call with proxy
endpoint mapping, which in this example is on the same server, to find the IP address to contact. For the endpoint registration process, the SIP registrar server, which can be a different server than the SIP proxy, receives a REGISTER message from the endpoint and confirms it with a 200 OK message. The information from the REGISTER message is used by the registrar server to update the database used by the proxy for SIP routing.

The To: <sip:roomsystem3@mobile-lab.local> from Figure 4.14 is the well-known address used to call roomsystem3. This example references directly to an endpoint as the name suggests, but it could also be a URI for a user e.g. Mad.Max@mobile-lab.local. The registrar binds the SIP URI and the IP address from the Contact URI including the IP address of the device. As soon as the SIP proxy receives an INVITE for the roomsystem3 URI, it will be proxied to the Contact URI (IP address) of the currently registered device. The registration is confirmed with a 200 OK message including an expire time of the registration of for example 300 seconds. The confirmation message is not shown here. The registration process begins during the start-up of the SIP endpoint and needs to be re-sent before the expiry time to stay registered. Multiple devices can register to the same URI. In the event of multiple URI registrations the SIP proxy may forward INVITE requests to all of them in parallel or in sequence.[42]

REGISTER sip:mobile−lab.local SIP/2.0
Via: SIP/2.0/UDP 192.168.1.153:5060;branch=z9hG4bK3502011676−814
Max-Forwards: 70
Allow: INVITE, BYE, CANCEL, ACK, PRACK, OPTIONS, REFER, REGISTER, UPDATE
From: roomsystem3 <sip:roomsystem3@mobile-lab.local>;tag=plcm.3502011733−814;epid=8213100FDF6ACV
To: <sip:roomsystem3@mobile-lab.local>
Call-ID: 3502011260−814
CSeq: 1 REGISTER
Expires: 300
Contact: roomsystem3 <sip:roomsystem3@192.168.1.153:5060;transport=udp>;proxy=replace;+sip.instance="<urn:uuid:d474c94c−bafe−552d−a565−bbe48321c35d>"
/* further details are omitted

Figure 4.14: SIP REGISTER message
4.2.3 Transport Protocol Selection

SIP supports different transport protocols as discussed above, and there must be a way of selecting one. The full description of DNS utilisation with SIP is defined in [49]. As the first step, a SIP client verifies the explicit indication of the transport protocol in the SIP URI. If the `transport=tcp`, `transport=udp`, or `transport=sctp`[38] is present the appropriate protocol is used. If the `transport=` is not specified in the URI, the following rules are followed:

1. If the URI is using a *numeric IP address*, UDP should be used, if the SIP URI uses the `sips:` schema for TLS encryption, TCP should be used.
2. If the URI is not using a numeric IP address, but has a *numeric port number*, then UDP should be used for a `sip:` URI and TCP for a `sips:` URI.
3. If the URI doesn’t have a numeric IP address or port and Naming Authority Pointer Resource Record (NAPTR) DNS queries are supported, then a NAPTR DNS query should be conducted on the host part of the URI. The supported service fields are: `SIP+D2U` for UDP, `SIP+D2T` for TCP, and `SIP+D2S` for SCTP transport. The result of the NAPTR query with the regex, forms a new URI, which is used in the next step: `SRV` lookup. If there are multiple NAPTR entries, the preference field is used for the protocol selection. If there is no NAPTR entry, the next step `SRV` lookup will be performed.
4. The DNS SRV records used by SIP are `_sip` or `_sips` for the protocol and `_udp`, `_tcp`, and `_sctp` accordingly for the transport. The SRV DNS query provides a hostname and port number and should be used by SIP requests. More details can be found here[50].
5. If no SRV records are found, then an A or an AAAA DNS query should be performed and UDP used for a `sip:` URI and TCP for a `sips:` URI.

4.3 XMPP - Extensible Messaging and Presence Protocol

For instant messaging and presence information there are several protocols available. SIMPLE, for example is based on SIP, and used by Microsoft Skype for Business. Another common protocol is XMPP, which is used by Cisco’s (Jabber) and Polycom’s (XMPP) VC infrastructure and is covered in this document. The Extensible Messaging and Presence Protocol (XMPP) is defined in RFC6120-6122[14, 15, 16] and uses XML for the exchange of data between the network entities in almost real-time.

XMPP uses small pieces of structured data, called “XML stanzas” using a client-server architecture to exchange those XML stanzas. The general process where a client connects to a server, exchanges XML, and ends the connection is outlined below in Figure 4.15 on page 43.[14]
1. IP address and connection port is determined via resolving a fully qualified domain name.

2. A TCP (Transmission Control Protocol) connection is established.

3. An XML stream is opened over TCP.

4. TLS (Transport Layer Security) is negotiated for the purposes of encrypting the channel. (This is optional)

5. Authentication is established via Simple Authentication and SASL (Security Layer).

6. A resource is bound to the stream.

7. An unbound number of XML stanzas are exchanged with the other network entities.

8. The XML stream is closed.

9. The TCP connection is closed.

The XMPP example in Figure 4.15 on the facing page demonstrates the XMPP session established by the test environment desktop system (192.168.1.200) to the video resource manager (192.168.1.64), which is acting as the XMPP server. The first step is the establishment of the TCP and XML session to server port 5222. The authentication process is conducted by a challenge response mechanism. After successful authentication a TLS encryption could be established, but for easier analysis of the protocol the encryption is disabled. The resource binding happens next and the presence information is exchanged. The last message is an instance message from the desktop client to a room system via the server.
4.3 XMPP - Extensible Messaging and Presence Protocol

Figure 4.15: XMPP example flow
4 Protocols and Related Standards

4.4 VC Endpoint and Software Client Management

When there are several endpoints and software clients geographically distributed, it is a challenge to manage them. The management tasks typically cover provisioning, address book, monitoring, software update, etc. Usually, for these tasks different known protocols are used to form vendor proprietary protocols.

This can create some level of interoperability between the different UC vendors. For example, two well known management suites are Polycom RealPresence Resource Manager[51] and Cisco TelePresence Management Suite (TMS)[52]. Both products cover the tasks above, but don’t fully support each other’s integration level, especially in the vendor’s own products e.g. Cisco provisioning works very closely with the Cisco endpoints and might have some limits with other vendor’s management products. For this document, only some of the Polycom implementations are highlighted and used in the test environment.

Provisioning

With provisioning service, endpoint and software client configuration can be centralised, and configuration settings can be transferred to the endpoint based on policies and profiles. On some endpoints, approximately 300 or more configuration items can be configured via templates. The preferred provisioning mode with Polycom endpoints is called dynamic-mode, which is only supported by Polycom clients and other vendors may use a different approach. In this mode, HTTPS and XML are used by the endpoint to poll the Resource Manager for new software updates and configuration provisioning profiles. Another benefit of endpoints in dynamic-mode is the configuration lock down. It’s not possible for the end user to change the configuration on the endpoint directly for example with the remote control. The HTTPS nature of the dynamic-mode makes it “firewall friendly”.[51]

Address book

The address book functionality allows the end user to search the address book either via remote control, a touch panel, or the ability to search for rooms or individual users to initiate a call is built into the software client or endpoint. There are many different address book integrations possible, depending again on the vendor’s product. Polycom’s simplest form of address book support with RealPresence Resource Manager is the Global Address Book (GAB) running as client/server protocol unencrypted on TCP port 3601 and provides, after a short registration process, in ASCII format, a flat list of endpoints and software clients. This list can be used by the endpoint to search, select, and dial.

In Figure 4.16 on the next page the client part of the conversation is highlighted in red and the answers from the Resource Manager are highlighted in blue. After registering itself as roomsystem3 with serial number, software release, and other information, roomsystem3 queries the address book with the GETALL command and the server (Resource Manager) responds with the full list of GAB entries with a time stamp at the end. This protocol is proprietary to Polycom and some other vendors are supporting it, by
means of reverse engineering. Other address book services are based on LDAP queries and can either query a local Resource Manager database or are proxied to a Microsoft Active Directory LDAP back end via the Resource Manager. The LDAP approach may, for example, be used by dynamic managed endpoints, or endpoints supporting the LDAP protocol for address book. LDAP provides a secure authentication with NTLMv2 and TLS encryption. LDAP and the encrypted versions are in general “firewall friendly”.[51]

4.5 Firewalls

One core functionality of SIP is to negotiate the ports, IP addresses and domain names required to describe the sessions it controls. SIP also manages session traffic to be established, for example RTP streams transporting audio and video, usually via dynamic UDP ports. There are two main issues with getting SIP to traverse firewalls; packet filtering and NAT. The first issue is getting SIP itself through, and the second is getting the media sessions it initiates to pass through. The second part is the more challenging one. This is not only limited to SIP signalling, H.323 and H.248/MEGACO are also affected. Because of the wide deployment of firewalls, which is in general a very good idea and it’s important to have one in place, UC traffic will not work across firewalls and NAT devices without special solutions.

A firewall is one of the most common security architecture components used to enforce perimeter security and can take many different forms such as hardware appliances, software that is integrated into an operating system like UNIX, Linux, or BSD, or routers or switches with firewall software enabled. A firewall enforces authorization policies on traffic between different security domains. A security domain, in the firewall context, is a network segment with common security criteria, which needs to be kept separated from other security domains and allow only communication between the domains based on a security policy. Very common security domains are outside, which represents the Internet, inside, which is normally the internal LAN, and a demilitarized zone (DMZ), where servers are located providing a service to the outside, but still have some connectivity to the inside. In real world scenarios, there can be more than the described domains depending on the complexity, organisation size, and assets to protect. A security policy definition should contain the following items:
4 Protocols and Related Standards

- Type of traffic to permit
- The users allowed
- The computer or machine
- Other restrictions such as time of day, content type, duration the access is allowed or transfer volume

Traffic not meeting the policy is dropped. An example rule may be as follows: allow ANY computer from inside to ANY computer at the outside security domain requesting the well known HTTP traffic (TCP/80) as destination service:

```
allow inside->outside source=ANY destination=ANY protocol=TCP port=80
```

Beside traffic policy enforcement between security domains, firewalls can provide additional services such as:

- Network address and port translation (NAT/PAT)
- User authentication
- Application Layer Gateway/Proxy (ALG)
- Protocol anomaly and intrusion detection/prevention
- Denial of Service (DoS) prevention
- Content filtering e.g. block Java
- URL filtering
- Virtual private networking (VPN)
- Antivirus and malware protection
- Network routing
- Logging and reporting

The simplest form of traffic policy enforcement is packet filtering. The firewall examines the source/destination IP address, source/destination port, and protocol type e.g. TCP/UDP, called 5-tuple, of each individual packet and compares it with the policy. Based on the policy, the packet is permitted (forwarded) or denied (rejected). The biggest issue with this type of filtering is that the state of a session is ignored. A stateless firewall does not maintain information about a session’s status, for example a TCP three-way-handshake, or return FTP traffic belonging to an outbound session. This type of filtering was typical in the early days of firewalls and has now been replaced by more sophisticated mechanisms.[53]

Stateful Firewalls and Application Layer Gateway/Proxy (ALG)

A stateful firewall (stateful packet inspection (SPI)) keeps track of related information to a network connection (TCP streams and UDP communication) passing through, and permits related session packets while others are rejected. TCP sessions can be monitored by TCP header information such as the sequence: SYN, SYN-ACK, ACK indicating...
a session establishment. The TCP sequence FIN, ACK indicates a connection close. For UDP, traffic timers are often used for a communication to be identified as active or closed.

An application layer gateway (ALG) enabled firewall reassembles all application related TCP/UDP packets, which can be for example DNS, HTTP, or SIP messages and examines the messages, possibly modifies them, and permits or rejects the traffic based on the configured policy. These secured application proxies, built into a firewall, are focused on application specific message content and ensure that protocol anomalies, misuse attempts, message sequence and protocol standards are all correct and they can block traffic to protect the application when necessary. For each application, a specific ALG needs to be implemented and kept updated by the firewall vendors when standards are changed or extended. The example ALG settings on the test environment firewall 6.1 on page 78 are shown in Figure 4.17 and are turned off.

![Figure 4.17: Example for ALG settings on Juniper SSG20 firewall](image)

In the context for SIP, ALG’s are also “fixing” issues caused by firewall packet filtering, NAT, and dynamically modify the configuration/traffic based on application specific information such as IP addresses, port numbers used on the application layer, and remapping of RTP traffic together with SDP. ALG’s built into firewalls don’t cover all aspects of UC requirements and using an “unknown” SIP extension may sometimes cause parts of the UC to fail to work. For example content sharing could be blocked but video and voice is working with ALG turned on. Because all the traffic, signalling and
media needs to traverse the ALG firewall, they have an impact on performance and add some additional delay into the real time communication. By experience of the author, it’s not recommended to solely put your trust in ALG’s built into firewalls for UC.

4.5.1 Network Address Translation - NAT

The depletion of available IPv4 addresses, which are essential for inter IP communication, brought about a solution called network address translation (NAT)[54, 55, 56] to map private IP addresses (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) defined in RFC1918[57] to public (in the Internet) routed IP addresses. IPv6 would be a much better solution with its address space of 128 bits compared with IPv4 32 bit, but NAT is maybe one of the reasons why IPv6 is not as highly distributed as it should be. Another reason for using NAT is to avoid IP address renumbering of computers when the Internet service provider changes. NAT handles the IP addresses on network layer 3 in the OSI model[58] and network address and port translation (NAPT) also modifies the port numbers in the IP packets and covers network layer 3 and transport layer 4 in the OSI model. Like packet filtering firewalls, NAT/NAPT rejects inbound connections from the Internet until a mapping exists. On the one hand, NAT provides some additional privacy, because the original private IP address can’t be recognised and is in general not reachable from the Internet. On the other hand it causes some additional challenges for troubleshooting, because it is difficult to determine the source of the problem since the IP address of the host is hidden. There are many problems NAT may cause, breaking the end-to-end architecture of the Internet, breaking the transitive reachability of computers, creating a single point of failure by the NAT device, and protocols must be aware of NAT operation in order to overcome the limitations.[55]

There are many different NAT classifications and variations out there, for example in RFC3489[59] Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) STUN: full cone, restricted cone, port restricted cone and symmetric are described. In this document only the basic NAT and NAPT based on RFC3022[56] Traditional IP Network Address Translator (Traditional NAT) are highlighted to explain the NAT impact on UC.

Basic NAT

With basic NAT a stub domain which usually utilises a set of private network addresses (RFC 1918[57]) will be dynamically mapped to a set of external globally routed IP network addresses. As long there are enough external addresses available, internal machines are able to communicate with hosts on the Internet. It is also possible to statically map a global to an internal host address to ensure that machine is always reachable from, and can communicate to, the Internet. The local host can run multiple sessions via the same IP address mapping. The addresses inside a stub domain have only local significance and are not valid or visible outside the stub domain. Thus, these local IP addresses e.g. 10.0.0.0/8 can be reused in a different stub network. At each stub router between the local network and the backbone, NAT is implemented.
In the basic NAT example in Figure 4.18 from RFC 3022[56] Traditional IP Network Address Translator (Traditional NAT), both stub networks A and B are using the same private IP address network 10.0.0.0/8. On the stub A router the class C network 198.76.29.0/24 is used and on stub B router 198.76.28.0/24 is used for NAT. The two NAT networks are globally routed IP networks unique for each stub network to enable them to appear on the Internet. If host 10.33.96.5 at stub A network wants to communicate with host 10.81.13.22 in the stub B network, it needs to use the globally unique address 198.76.28.4. The router at stub A maps and changes the source address to 198.76.29.7 before the packets are sent to stub B router. The NAT router B looks up the static mapping table and changes the destination IP address from 198.76.29.7 to 10.81.13.22 and forwards the packets to the local network. The return communication goes through the same process with the source and destination reversed. In many cases the NAT process is transparent to the hosts and doesn’t need to be reconfigured.[56]

Network Address Port Translation - NAPT
A very common stub network scenario for Small Office Home Office (SOHO) is a stub router provided with a single globally routed IP address on the WAN interface and the remaining hosts on the local network using private RFC1918[57] IP addresses, which have only local internal organisation significance. RFC 2663[60] NAT Terminology and Considerations introduces the term “TU ports”, which refers to TCP/UDP ports associated with an IP address. The TU port range 0-1023 are also called well-known ports.
defined in RFC1700[61] Assigned Numbers for listening services like HTTP on TU port 80. The NAPT concept can allow inbound traffic by statically mapping a specific service e.g. Telnet TU port 23 onto the global IP address to a local node. NAPT uses tuples for mapping in the form of:

```
local IP address, local TU port number to
globally routed IP address, assigned(well known) TU port number
```

The example illustrated in Figure 4.19 from RFC3022[56] uses on the local stub A side, the private IP network 10.0.0.0/8 and on the WAN side of the router, the globally routed single IP address 138.76.28.4 provided by the service provider. Host 10.0.0.10 on the local stub A network initiates a Telnet (TCP port 23) session to destination host 138.76.29.7 on the Internet via its stub A router. The NAPT on the router changes the source IP address 10.0.0.10 and source port 3017 into the globally unique IP address 138.76.28.4 with source port 1024, and forwards the packet to the next router. The return traffic goes through the same process and modifies the IP and ports based on the previously created mapping. Again this process is transparent to the hosts and doesn’t need to be reconfigured.

```
\ | /
+-+----------------+
| Service Provider Router|
+-+----------------+
|                 |
| WAN             |
+----------------+------------------+
| Stub A           |
+----------------+------------------+
|                 |
| |                |
| | 10.0.0.0/8      |
| |                |
| |                |
| | 138.76.28.4, sport=1024, v[s=138.76.29.7, sport = 23, d=138.76.28.4, dport = 1024] |
| |                |
| +----------------+------------------+
| | Stub Router W/NAPT |
| |                 |
| | LAN             |
| | 138.76.29.7, dport=23} v d=138.76.28.4, dport = 1024} |
| |                |
| |                |
| | 10.0.0.10, sport=3017, v[s=138.76.29.7, sport=23, d=138.76.28.4, dport = 1024} |
| |                |
| |                |
| |                 |
| | 10.0.0.1        |
| | 10.0.0.2        |
| | 10.0.0.10       |
```

Figure 4.19: NAPT example[56]
Many Internet application protocols work fine with NAT and firewalls as long as some guidelines are followed as described in RFC3235\[62\] Network Address Translator (NAT)-Friendly Application Design Guidelines. Unfortunately with SIP and RTP the initial RFCs ignored NAT and assumed IPv6 would remove the need for NAT. Some of the main protocol design recommendations for NAT are:

- Client/server systems are more workable compared with peer-to-peer applications acting as servers
- Applications requiring end-to-end IPSec will fail
- Use DNS names, not IP addresses in payload
- Multicast is difficult
- Avoid session bundles (example is FTP: one for control and one for data transfer)
- Use TCP instead of UDP

SIP and RTP violates most of the points above and early solutions to this problem used ALGs and later STUN defined in RFC3489\[59\] Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) directly on the SIP User Agent to identify if it is behind a NAT device and its' mapped addresses. Unfortunately STUN doesn’t cover all NAT variations and the IETF developed RFC5245\[63\] Interactive Communication Establishment (ICE), both described in the paragraphs below. Other UC protocols such as XMPP for IM, HTTPS provisioning, or the Polycom GAB protocol are more NAT friendly and are not affected, only firewall filter rules need to be applied.\[42\]

In Figure 4.20 on the following page the same SIP INVITE message is shown as in the previous example in Figure 4.3 on page 28 highlighting the problem fields in red.

There are three main problems with this message:

1. In the Via header field the private IP address 192.168.1.153 and listening port 5060 is used and may not have an active mapping and filter rule.
2. The Contact URI may not be routable because of its private IP address 192.168.1.153
3. The SDP information in the c= and m= lines will not work behind NAT using a local IP address and port.

For the SIP part there are several extensions available such as: symmetric SIP\[64\], connection reuse\[65\] and SIP outbound\[66\] to help overcome the NAT/firewall problems for the SIP signalling part, but media issues are not resolved. The main focus on media traversal is to get RTP and RTCP negotiation using SIP to work, which can be achieved with symmetric RTP, using RTCP attributes described in RFC3605\[67\], or a self-fixing approach using STUN information.
The best solution for media NAT traversal is to use, at the first stage hole punching in combination with ICE and if this is not successful fall back on TURN, both described below.[42]

4.6.1 Hole Punching

_Hole punching_ is a probing technique, used to enable two clients, for example two SIP UAs, to set up a direct session (UDP or TCP) with the help of a rendezvous server, for example a SIP proxy, even when the endpoints are behind NATs. This is a very common method used for peer-to-peer networks such as online gaming, file sharing, and as in this document for UC. There are several different hole punching processes for UDP and TCP documented[68, 69]. In this document only the UDP hole punching with endpoints behind two different NATs is described by the example taken from[69] with minor modifications below:

Registration

1. In Figure 4.21 on the facing page you will see that both clients A and B have a private IP address and each is behind a separate NAT and initiated UDP communication sessions from their local port 4321 to port 1234 on server S, the well known rendezvous point.

2. To deal with the outbound sessions, NAT A assigns at its own public IP address, 155.99.25.11, port 62000. This is for A’s session with S. Likewise, NAT B, at IP...
3. When A sends the registration message to S, the private endpoint is sent as 10.0.0.1:4321, with 10.0.0.1 being the IP of A on the local private network. S then notes A’s private address alongside the public address. A’s public address in this case is 155.99.25.11:62000, the temporary address assigned to the session by NAT A. Similarly, when client B registers, S records B’s private address as 10.1.1.3:4321 and B’s public address as 138.76.29.7:31000.

Hole Punching Process

1. In Figure 4.22 on the next page Client A sends a request message to S asking for help connecting with B.

2. In response, S sends B’s public and private address to A, and sends A’s public and private address to B.

3. A and B each start trying to send UDP datagrams directly to each of these addresses.

Assume the first message sent from A’s to B’s public endpoint is as shown in Figure 4.23 on the following page. A’s NAT receives the outbound message and identifies it as
Figure 4.22: UDP Hole Punching, Endpoints behind different NATs process

Figure 4.23: UDP Hole Punching, Endpoints behind different NATs completed
the initial UDP packet of a fresh outgoing session. The source endpoint is the same as that of the session already existing between A and S, ie. 10.0.0.1:4321, but it also notices that the destination address differs. All being well, NAT A will maintain A’s private address identity and translate all the outbound sessions from the private source address 10.0.0.1:4321 to the public source address that corresponds, ie. 155.99.25.11:62000. A’s first outgoing message to B’s public endpoint thus, in effect, punches a hole in A’s NAT for a new UDP session identified by the address tuple (10.0.0.1:4321, 138.76.29.7:31000) on A’s private network, and by the addresses (155.99.25.11:62000, 138.76.29.7:31000) on the Internet side.[69]

If A’s message to B’s public endpoint reaches B’s NAT before B’s first message to A has crossed B’s own NAT, then B’s NAT may interpret A’s inbound message as unsolicited incoming traffic and drop it. B’s first message to A’s public address however, similarly opens a hole in B’s NAT, for a new UDP session identified by the addresses (10.1.1.3:4321, 155.99.25.11:62000) on B’s private network, and by the addresses (138.76.29.7:31000, 155.99.25.11:62000) on the Internet. Once the first messages from A and B have crossed their respective NATs, holes are open in each direction and UDP communication can proceed normally. Once the clients have verified that the public endpoints work, they can stop sending messages to the alternative private addresses.[69]

About 82 % of the NATs tested in Peer-to-Peer Communication Across Network Address Translators[69] support hole punching for UDP across the tested router/firewalls and operating systems.
4.6.2 STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)

Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) was introduced in RFC3489[59], updated, and obsoleted by Session Traversal Utilities for NAT (STUN) RFC5389[70] with a new acronym definition. The STUN protocol provides a tool for dealing with NATs and enables an endpoint to determine the IP address and port allocated by a NAT mapped to its private IP address and port. It also provides a way for an endpoint to keep a NAT binding alive. A STUN client, which can be built into an SIP UA, sends mapping requests to a STUN server located on the public Internet. The STUN server responds with the mapped address/ports received from the last NAT device on the communication path (see Figure 4.24). It is possible to have multiple NATs and ALGs on the communication path and the response IP address needs to be XOR to prevent modification of the response message by ALGs or NATs. The main functions provided by STUN arc:[42]

- Mapping discovery
- Performing connectivity check with a server or SIP UA used with ICE as described below
- Media relay usage defined by the TURN extension RFC5766[71] as described below
- Keep-alive for UDP NAT sessions mapping used by SIP outbound[66]
- NAT behaviour discovery defined in RFC 5780[72]
- Detecting ALGs, which rewrite IP addresses

![Figure 4.24: STUN Flow Diagram][73]

It is important to highlight, that STUN can discover the NAT characteristics only for a specific IP address at a specific time, which can change.
4.6.3 TURN - Traversal Using Relays around NAT

Traversal Using Relays around NAT (TURN) defined in RFC 5766[71] is a relay extension for STUN[70] and provides a relay service for hosts behind a NAT. If both of the hosts reside behind NATs that do not display good behaviour e.g. that have a mapping behaviour of "address-dependent mapping" or "address- and port dependent mapping", then hole punching techniques generally fail.

In the event that a direct communication path cannot be found, the services of an intermediate host (TURN) are required to be used. This acts as a relay for the packets located in the public Internet and relays packets between two hosts that both sit behind NATs. In Figure 4.25 the protocol flow of TURN is demonstrated.

![TURN Flow Diagram](73)

The disadvantages of the TURN relay server are:

- Bandwidth and CPU requirements for each media stream (inbound + outbound)
- Single point of failure by the TURN server
- Introduced delay/latency by relaying
- Longer network path higher latency
- ICMP is not relayed, hence no MTU path discovery
- TOS/DiffServ fields may not be relayed, loss of QoS information
- Keep-alives are required

With the disadvantages mentioned above, the TURN server approach should be avoided and used only when other mechanisms such as hole punching fail.
4.6.4 ICE - Interactive Connectivity Establishment

Interactive Connectivity Establishment (ICE) defined in RFC5245[63] makes use of the Session Traversal Utilities for NAT (STUN) protocol and its extension, Traversal Using Relay NAT (TURN). ICE can be used by any protocol utilizing the offer/answer model, such as the Session Initiation Protocol (SIP) together with Session Description Protocol (SDP).

ICE enabled SIP UAs gather as many private and public transport address candidates as they can, by using physical/logical local addresses and others discoverable via STUN and TURN. The candidates are listed by priority with TURN having the lowest priority. After the address candidates are exchanged via SDP and offer answer exchange using SIP, both sides begin hole punching and keeping track of successes and failures of the tested candidates. The UAs choose the working candidate with the highest priority, which could be, in a worst case scenario TURN, if hole punching fails. The following Figure 4.26 is an example SDP message that includes in the two a=ice fields and the last two a=candidate: lines, the ICE attributes[63]:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 192.0.2.3 45664 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr 10.0.1.1 rport 8998
```

Figure 4.26: Example ICE SDP candidates

In addition to the NAT firewall traversal features ICE supports:

- IPv4 and IPv6 for dual stack UAs
- Keep-alive for UDP mappings
- Some level of media authorization with a ICE exchange check to ensure both UAs agree to receive and send media

4.7 Session Border Controller (SBC)

Session Border Controller (SBC) is under discussion for “breaking” the direct SIP UA end-to-end architecture by introducing a back to back user agent (B2BUA) between a private and public network. In Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments RFC5853[74] many aspects are discussed including the conflict with SIP architectural principles[75]. SBCs solve the most critical
problems surrounding NAT/firewalls and provide successful UC calls vs. failed calls following a clean architecture. The B2BUA acts on the public side as a user agent server and handles UA sessions from the public network side, applies policies, processes the sessions, possibly relays the media, and “proxies” the session into the private network. A B2BUA uses different Call-IDs, To tags and From tags for each side of the sessions and maps the two sides (see Figure 4.27). The outbound communication would be managed in a similar way.

![Figure 4.27: SBC Architecture](74)

SBCs provide the following common functions:

- Topology and IP address hiding
- Media traffic management
- Fixing capability mismatches between UAs
- Maintaining SIP-related NAT bindings
- Access control
- Protocol repair for badly implemented clients
- Media encryption/decryption
- Lawful interception
- Protocol transcoding e.g. SIP/H323
- Media relay and transcoding

SBCs are very common in service provider environments and are increasingly used by enterprises in general, especially for cloud based UC solutions. It’s required for organisations to control the traffic of their users from and to the cloud services.

### 4.8 Conclusion

In this chapter the focus was on relevant protocols and formats and highlighted the strengths and weaknesses of them. We have seen how firewalls add security, but also pose a problem for communication, and we’ve looked at how this problem can be solved via Hole Punching, STUN, TURN and other means. There is still a need to cover other
UC relevant security considerations, which are covered in the next chapter followed by some real life examples. Chapter 6 on page 77 shows the ability to transfer theory into real working scenarios with products used to build UC environments today.
5 Security Considerations for UC

With the wide distribution and exposure of UC architectures to the Internet, the need for security is obvious. Compared with the classical PSTN and its perimeter and physical security, UC is much more exposed to threats via the Internet. Hence, UC, like other Information Technology areas, needs to be covered by a regular Information Security Management System (ISMS). This chapter highlights the threats to UC systems and possible approaches to reduce the risks. Some of the classical security risks are shown in the following list, but is not a comprehensive list of all known risks.

- Information leakage (allowing unauthorised people to view information)
- Integrity violation (changes to the information)
- Denial of Service DoS and distributed DoS (user is denied access to a service when they wish to use that service)
- Illegitimate use (use by unauthorised users)
- Masquerade/man-in-the-middle attack (adopting the identity of another entity)
- Bypassing controls (exploiting system security weaknesses to gain unauthorised rights)

5.1 UC Threats, Attacks, and Risk Mitigation

5.1.1 Generic Threats

ARP poisoning

The address resolution protocol (ARP) is an OSI layer 2 protocol for mapping OSI layer 3 IP addresses to OSI layer 2 MAC addresses. A possible attacker could “poison” the ARP cache of network devices by flooding their cache with forged gratuitous ARP messages and redirect traffic on a LAN or broadcast domain of UC devices. It’s also possible to use this technique for a DoS attack, preventing hosts communicating via OSI layer 2 with their correct servers or default gateway. Unfortunately ARP doesn’t provide any authentication mechanism to prevent this kind of attack. One solution is to disable the acceptance of gratuitous ARP packets or use manually defined ARP caches, which is not very administration friendly. Switch port authentication together with 802.1x and digital certificates could address the authentication and authorization of devices connected to a switch port. As an example the Security Guide for Cisco Unified Communications Manager [76] describes the hardening of IP phones.
5 Security Considerations for UC

DNS

The domain name system (DNS) is used in many cases for UC e.g. SVR lookups or regular A record queries and is vulnerable to reply, DoS and impersonation attacks on different levels. One example is the direct attack on the root servers of a certain domain and modification of the DNS databases directly via a virus or spyware, which would provide forged DNS responses to UC clients. A DNS zone file transfer alternation is another possible attack vector to modify and redirect DNS communication. If the client resolver doesn’t validate the DNS responses with a mechanism such as DNSSEC (DNS security extensions)[77] it’s possible to “poison” the local resolver cache. Alternatively the local cache could be disabled and only DNSSEC queries are accepted by the UC infrastructure and clients. One reason for slow adoption of DNSSEC is the requirement of implementing digital signing of DNS responses beginning at the root of DNS.

QoS Misuse

One common weakness related to bandwidth reservation enabled for a service (e.g. video transport) is another service or an attacking traffic stream using it. This risk is inherent in DiffServ technology, which depends on correct packet markings. When bandwidth reservation or a priority queuing system is used in a vulnerable network, the use of authentication and flow admission is recommended as described in A Survey of Authentication Mechanisms[78].

5.1.2 SIP Threats

This section highlights some specific threats to SIP that stress the need for implementing security mechanisms. The examples below are based on SIP RFC 3261[10] and don’t provide an exhaustive list of possible attacks to SIP, but instead demonstrate the need for security services, which can reduce the risks.

Registration

The registration process of a UA at a SIP registrar with its address of record is used for the identification of a device belonging to a specific user. A possible attacker could change these registration associations with a different address of records and redirect “hijack” a SIP URI, by pointing to a different device. The attacker could also flood the registrar with registrations or de-register devices and cause a denial of service attack (DoS). These examples demonstrate the requirement for an authentication mechanism of the originator (UA) of such requests.

Impersonating a Server

Another possible attack could be conducted on the infrastructure side by an SIP proxy or redirect server, which pretends to be responsible, for example for domain company.com. A UA sending requests to this bogus server could be cheated with forged response messages and be redirected to unsecure resources. To prevent this kind of attack, UAs need to be enabled to authenticate the servers.
Message Body Tampering

To prevent the modification of, or information leakage about a session by a man-in-the-middle attack, users need to secure the SIP message bodies and (with some limitations) some of the header fields. An attacker could, for example modify SDP attributes or eavesdrop the UC communication. The security services need to provide confidentiality, integrity and authentication end-to-end, but should not prevent intermediaries such as proxy servers from handling the signalling correctly.

Tearing Down Sessions

Another possible attack could focus on ending initiated sessions by previously observed session parameters and inserting a BYE message. The best countermeasure is the authentication of the sender of the BYE message.

Denial of Services and Amplification

Denial of service attacks focus on the prevention of service usage for legitimate users and come in many different forms. One example attack could be based on sending out bogus requests with a fake source IP address including a modified SIP Via header field, which indicates the DoS target as the originator of the request. By sending this specially crafted message to a large number of SIP network elements like UAs and proxies, answering these requests and thus the “overloading” of the target occurs, also known as distributed denial of services (DDoS) attack.

Another example attack could send a large number of REGISTER messages to a SIP registrar, exhausting its memory and CPU and preventing legitimate users from using the service. These problems require a properly designed architecture and implemented security service recommendations to limit the risk of such potential attacks.

5.1.3 SIP Security Services

As described in the threats and attacks on SIP above, the following fundamental security services are identified and described in RFC 3261[10]:

- Confidentiality and integrity of messaging
- Preventing replay attacks & message spoofing
- Authentication and privacy of the session participants
- Preventing denial-of-service attacks

During the definition of SIP, the focus was on reusing existing technologies and not on “reinventing the wheel”, by using existing security models derived from HTTP and SMTP. The best approach would be the full encryption of messages end-to-end to provide confidentiality and integrity. Unfortunately the intermediary proxies need to be able to read and modify some of the SIP header fields. Hence, proxy servers must therefore be
trusted by SIP UAs to a certain level and encrypt the entire SIP messages on a hop-by-hop basis utilising basic security mechanisms, including the ability for the endpoints to verify the identity of proxy servers.

### TLS & VPN/IPSec

With *Transport Layer Security (TLS)* [79], network layer security (VPN/IPSec - *Security Architecture for the Internet Protocol* [80]), together with digital certificates (most of the time), confidentiality and integrity is ensured for signalling traffic.

IPSec uses several networking protocols to securely replace traditional IP communication between two sites or hosts by using security gateways e.g. firewalls with IPSec capabilities or direct implementation on the host operating system. The security architecture provides confidentiality and integrity, defined for specific traffic or all traffic. Most of the time IPSec is transparent to the overlay UC architecture and has no direct integration point. All required IPSec profiles, configuration and key management are handled separately. When VPN/IPSec is used together with UC and its real time traffic requirements, it could have a negative impact on QoS, latency, jitter and packet loss.

TLS provides confidentiality and integrity on the transport layer over connection oriented protocols such as TCP. In the context of SIP, TLS means TLS over TCP and can be specified in the *Via* header field or directly in the SIP URI by using the SIPS: scheme. Alternatively SCTP could be used for transport. TLS is a very good architecture for hop-by-hop security without there being a pre-existing trusted association between the hosts. In contrast with IPSec, TLS needs to be tightly integrated into the SIP UAs. With the UA trusting its local proxy (after verifying the certificate), there is no assurance for the initiator that all hop-by-hop communication is secured with TLS. As defined in *The Transport Layer Security (TLS) Protocol Version 1.2 - RFC5246* [79] the cipher suite `TLS_RSA_WITH_AES_128_CBC_SHA` must be supported as a minimum.

### SIPS URI Scheme

If SIPS URI (e.g. `sips:roomsystem2@mobile-lab.at`) is used as address-of-record for a user and registrations in the *To* and *REGISTER* requests, the SIPS scheme requires that each hop over which the request is forwarded, until the request reaches the SIP UA responsible for the domain portion of the Request-URI, must be secured with TLS. As soon as the domain responsible proxy is reached, the local security policy can be enforced to keep using TLS on the internal SIP domain with the UAs. The use of SIPS URI enables the use of mutual TLS authentication. The mutual TLS authentication should be supported and is not a requirement. During the authentication process, the received certificate should be validated against the root certificate stored by the UA. If the certificate doesn’t exist or is not valid the request/connection should be refused.
5.1 UC Threats, Attacks, and Risk Mitigation

HTTP Authentication

SIP utilizes HTTP Digest authentication with 401 and 407 response codes together with header fields to carry challenges and credentials for replay protection and one way authentication. The details regarding the implementation are covered in Section 22 in RFC 3261[10].

S/MIME

As described above, it is not possible to encrypt the entire SIP header between two UAs, without breaking the functionality for intermediate hosts, such as proxies, to route the SIP messages. S/MIME allows SIP UAs to encrypt MIME bodies without affecting the relevant SIP message header fields for routing purposes. With S/MIME confidentiality, integrity and mutual authentication service can be provided as described in detail in section 23 in RFC 3261[10].
5 Security Considerations for UC

5.1.4 SIP Security Services Implementation

The SIP security service implementation requirements are defined in RFC 3261[10] section 26.3 and are summarized in Table 5.1. The requirements terminology is based on Key words for use in RFCs to Indicate Requirement Levels - RFC2119[81].

<table>
<thead>
<tr>
<th>Component</th>
<th>Security Service and Behaviour</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>All SIP Components</td>
<td>Digest Authorization</td>
<td>MUST</td>
</tr>
<tr>
<td>All SIP Components supporting TLS</td>
<td>mechanism to validate certificates during TLS negotiation</td>
<td>MUST</td>
</tr>
<tr>
<td>All SIP Components supporting TLS</td>
<td>SIPS URI scheme</td>
<td>MUST</td>
</tr>
<tr>
<td>Proxy\Redirect server, and Registrars</td>
<td>TLS, mutual and one way authentication</td>
<td>MUST</td>
</tr>
<tr>
<td>Proxy\Redirect server, and Registrars</td>
<td>possess site certificate with subject matching the canonical hostname</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Proxy\Redirect server, and Registrars</td>
<td>configured with at least one Digest realm</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Proxy\Redirect server, and Registrars</td>
<td>configured with at least one Digest realm</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Proxy\Redirect server, and Registrars</td>
<td>implement IPSec or other lower-layer security protocol</td>
<td>MAY</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>initiate TLS</td>
<td>RECOMMENDED</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>initiate a TLS connection to Proxy\Redirect server, and Registrars</td>
<td>SHOULD</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>reuse of certificate validation to verify S/MIME</td>
<td>SHOULD</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>receive requests via TLS</td>
<td>MAY</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>act as TLS server</td>
<td>MAY</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>certificate for mutual TLS authentication</td>
<td>MAY</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>transference of credentials with S/MIME</td>
<td>MAY</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>hold specific root certificates for S/MIME</td>
<td>MAY</td>
</tr>
<tr>
<td>User Agents (UAs)</td>
<td>signing and encrypting of MIME bodies</td>
<td>MAY</td>
</tr>
</tbody>
</table>

Table 5.1: SIP Security Service Requirements for SIP Elements

The “toolbox” of security services described in Table 5.1 needs to be coordinated and correctly implemented to reduce the risk of threats as described above. Based on the requirements in the table, not the most appropriated or secure services are MUST e.g. mutual TLS authentication and clearly show a compromise of minimum requirements. In this document, only a selection of common implementation scenarios are covered below and are based on the SIP standard RFC3261[10]. Figure 5.1 on the facing page demonstrates the SIP REGISTER and INVITE process between two different SIP domains abc.com ⇔ xyz.com and further details are provided in the next sections.
Registration

The SIP UA roomsystem1 labelled Alice registers after start-up with its local SIP registrar (SHOULD) using TLS. How the UA finds its SIP registrar server is described in RFC3261[10] section 10. During the TLS negotiation process, the registrar server SHOULD offer a certificate corresponding to the domain the UA tries to register. In this example, Alice tries to register as alice@abc.com to the server and the registrars certificate site identification provided to Alice MUST correspond with the domain, for example sip-proxy@abc.com. Alice SHOULD verify the certificate. If the certificate is not valid or is expired, Alice MUST NOT proceed with the registration process. Alice forms a REGISTER request which SHOULD be addressed in the Request-URI to the same domain name as provided in the registrar provided certificate:

```
REGISTER sip:abc.com SIP / 2.0
```

This process is highlighted in Figure 5.1 indicated by number 1. After the registrar receives the REGISTER message via the TLS connection that is already established, it SHOULD challenge Alice with a 401 Proxy Authentication Required response. In the authentication header sent from the registrar to the UA, the “realm” SHOULD match to the domain name previously received in the certificate during TLS negotiation. As soon as Alice receives the challenge, the UAC SHOULD respond either with the user credentials or use the credentials from a keyring. The username part of the credentials SHOULD match with the “userinfo” part of the URI in the To: header field in the REGISTER request. As soon the Digest credentials have been inserted into the Proxy-Authentication fields, the REGISTER request needs to be re-sent to the registrar. The Digest authentication process is highlighted in Figure 5.1 indicated by the number
2. As soon as the registration is accepted by the registrar, the UA SHOULD keep the TLS connection established and reuse it for further communication between the UA and proxy server. This requires the registrar and the proxy server to be the same server. More details are provided in RFC3261[10] section 26.3.

These two security services ensure the authentication of the registrar and proxy to Alice and by establishing the TLS tunnel creates a confident connection for SIP signalling. Alice is also authenticated via the Digest Authentication to the registrar and proxy. The communication between Alice and the registrar/proxy is protected against spoofing, man-in-the-middle, DoS and replay attacks.

**Interdomain Requests**

In the next scenario, as described in RCF3261[10] section 26.3, Alice wants to initiate a SIP session to bob@xyz.com who is not in the same administered SIP domain. The two proxies in Figure 5.1 on the previous page are setup to act as a local outbound proxy. Alice is already registered to the local proxy as described above and SHOULD reuse the existing TLS connection for sending the INVITE message and SHOULD reuse the cached credentials for the INVITE to avoid multiple authentication prompts to the user.

After the proxy receives the INVITE, the server verifies the credentials to authenticate the message received from Alice. During the inspection of the Request-URI domain part, in this case xyz.com, the proxy SHOULD make the routing decision as to whether this message should be handled locally or routed to a different proxy. If Alice wants to communicate with alex@abc.com, the already established TLS connection during registration of Alex SHOULD be reused for local proxying of Alice requests to Alex indicated by number 3 in Figure 5.1 on the preceding page. Because both local UAs have established a secure and authenticated connection to the proxy, they can trust that the local domain communication is secured (authentication, confidentiality and replay protection).

In this example, the Request-URI indicates the external domain xyz.com and the abc.com local outbound proxy SHOULD establish a TLS connection to the xyz.com proxy. Both session participants are servers and SHOULD possess site certificates and SHOULD conduct mutual TLS authentication by verifying each other’s certificate, matching the domain of the proxies. In general, the mutual authentication of proxy servers together with TLS, significantly reduces the threat of a DoS. After a successful certificate verification, the TLS tunnel can be established between the two servers and the INVITE request can be forwarded, indicated by number 4 in Figure 5.1 on the previous page.

The proxy server at xyz.com SHOULD compare the domain part in the SIP From: header field with the offered certificate from the abc.com proxy. The xyz.com proxy MAY have a security policy in place so that if a mismatch of the domain in the certificate doesn’t match with the domain received in the proxied message, it has to be rejected.
5.1 UC Threats, Attacks, and Risk Mitigation

This domain verification prevents unauthenticated SIP message processing and a potential DoS attack, without this verification there is some similarity to a SMTP open relay. This secure channel between the two proxies doesn’t guarantee to xyz.com that Alice is authenticated or what kind of security policy is implemented on the abc.com proxy at this stage. It requires a certain level of trust between the two organisations.

As soon as the INVITE message has been accepted by the xyz.com proxy, the existing TLS channel with the associated user agent for bob@xyz.com SHOULD be identified and proxied through this channel to Bob. The INVITE request received by Bob via the secured TLS connection, which has been authenticated during the registration process, ensures that the From: header field was not tampered with.

During the forwarding process between the two proxies, they SHOULD add a Record-Route SIP header field, to ensure future communication between the two UAs in this dialogue will pass through the same secure channel for the lifetime of the SIP dialogue. In case the proxies don’t add the Record-Route header, the UAs would try to pass future messages directly end to end between Alice and Bob without using the established security channel. More details are provided in RFC3261[10] section 26.3.

There are several limitations identified with the security services in this section, but they are out of scope of this document and can be reviewed in RFC 3261 [10] section 26.4.

5.1.5 Media Security

How to establish an authenticated, confidential and integer secure communication channel for SIP signalling is covered in 5.1.4 on page 66. It’s equally important to establish a secure media session to limit the risks, for example of eavesdropping. For RTP[19] and RTCP a new profile RTP/SAVP has been introduced and defined in The Secure Real-time Transport Protocol (SRTP) - RFC3711[40] to provide confidentiality, message authentication and replay protection for the media session. Other goals of SRTP are the flexibility to upgrade to new cryptographic transforms, low bandwidth overhead with limited packet expansion, low computational costs together with a small footprint, and independence to network transport protocols.

There are different key management protocols available for SRTP e.g. MIKEY, DTLS-SRTP, ZRTP. In this document the SDES SDP security descriptions are described, which are defined in Session Description Protocol (SDP) Security Descriptions for Media Streams -RFC4568[82]. SRTP uses symmetric keys and ciphers, which are negotiated with SIP/SDP during the offer/answer exchange. The default cipher used is Advanced Encryption Standards (AES)[83] in Segmented Integer Counter or f8 mode. The NULL cipher is also defined in case encryption of RTP is not required. Figure 5.2 on the next page shows as an example the SDP extract for the m=video part of a SIP INVITE offer with SRTP. The SDP profile RTP/SAVP for SRTP, together with the attribute a=crypto:1 - a=crypto:4 is used to specify the offered encryption and message authentication schemes.
e.g. AES in counter mode with 256 bit HMAC SHA1 with 80-bit. The inline: key has a master key with a salting key concatenated to it; both are base-64 encoded. The $2^{31}$ specifies the lifetime of the keys. The plain text key in the SIP/SDP header needs to be authenticated and encrypted, since tampering and access to security parameters must be protected against unauthorized parties.[40, 25, 42, 84]

SRTP generates, with AES in counter mode, an encrypted key stream which is XORed with the plain text media payload. This architecture allows the parallel CODEC processing of media and encryption without increasing the processing delay significantly. With the use of Multiple Synchronization sources (SSRC) for the SRTP sessions (RTP and RTCP), a single SRTP master key is used in both directions and for multiple media streams. The master key and salt is used to derive the session encryption key, session authentication key, and session salt key. The SRTP header is identical to the RTP header, but the payload is encrypted, padding added (if required), and an OPTIONAL authorization tag covering the SRTP header and payload is added. An OPTIONAL Master Key Identifier - MKI can be added to use for key management for the purpose of re-keying, but most implementations don’t use it.[40, 25, 42]

```plaintext
/* some header details are omitted

m=video 16582 RTP/SAVP 116 109 110 111 96 34 31 b=TIAS:1024000
a=crypto:1 AES_CM_256_HMAC_SHA1_80 inline:7hUIgCzL7GmaaRuRRf/WqmohuIPex0NL9VtxBtjTsiaoeyjzMCkK+zcC/9mWA==|2^31
a=crypto:2 AES_CM_256_HMAC_SHA1_32 inline:eVGNyZq/DMyR/inKXmk6aqgT]Du0SSHOxFhVs8kzep1m7jME620NAhk6Nhj/CA==|2^31
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:5lEVN8gC3aqw5FA+e9bqaWxvJvdZOXV/Kw/AQQIKj
|2^31
a=crypto:4 AES_CM_128_HMAC_SHA1_32 inline:3sXBChNRmJUlaOERO6zikDXHQFZVYojwBKIKZz7y
|2^31
a=rtpmap:116 vnd.polycom.lpr/9000
a=fmtp:116 V=2;minPP=0;PP=150;RS=52;RP=10;PS=1400
a=rtpmap:109 H264/90000
a=fmtp:109 profile−level−id =428020; max−mbps=490000; max−fs =8192; sar−supported =13; sar=13
a=rtpmap:110 H264/90000
a=fmtp:110 profile−level−id =428020; packetization−mode=1; max−mbps=490000; max−fs =8192; sar−supported =13; sar=13
a=rtpmap:111 H264/90000
```

Figure 5.2: Example SRTP RTP/SAVP profile and a=crypto in SIP/SDP INVITE

Normal security policies would require an end-to-end encryption of SDP for example with S/MIME, but in practice many implementations use the hop-per-hop transport over TLS, which exposes the keys to each SIP proxy that forwards the INVITE and/or 200 OK message. There are several limitations around SDES keying e.g. SIP forking, redirection and forwarding which are described in Requirements and Analysis of Media Security Management Protocols - RFC5479[85] and introduced new approaches such as Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP) - RFC5764[86] and ZRTP: Media Path Key Agreement for Unicast Secure RTP - RFC6189[87].
5.2 Real Life Examples

The discussion points from the previous section highlight some of the threats and possible attacks on UC infrastructures. The introduced security services, when implemented correctly, can help to reduce the risk. This section demonstrates some real life attack examples.

5.2.1 Anatomy of an SIP Attack

With the growing deployment of UC systems and the exposure to the Internet of related services such as SIP proxies and registrars, the misuse and defrauding of these services is increasing. The easy accessibility via the Internet allows registration attempts and the subsequent step to initiate a toll call via SIP to an international premium number or to gain “free” telephone calls. The attacker gains a financial benefit and the account owner is caused financial damage.

A group of network and security researchers in Germany conducted a three year analysis on VoIP attacks with the use of a SIP Honeynet[88]. During the period December 2009 to November 2012 they collected via their Honeynet network over 90 million SIP messages and after some analysis identified 5684 attack samples. The team recognized four attack stages within the collected data:

1. SIP Scan
In the SIP standard[10], the method OPTIONS is required (MUST) to be implemented by every UA to query the supported capabilities of another UA or proxy server before, for example, an INVITE message is sent. This behaviour is used by an attacker to identify “listening” SIP UAs on the Internet, by using automated tools like vmap from the SIPvicious tool box[89]. If the OPTION method is blocked the attacker can alternatively use the REGISTER message.

2. Extension scan
After potential SIP hosts are identified, the attacker tries to register with several extensions e.g. 100 - 9999 without a password. If the extension exists on the server, it responds with a 403 FORBIDDEN, indicating there was no password provided. If the extension doesn’t exist, the server should respond with 404 NOT FOUND. With this method the attacker can build a list of existing extensions hosted on this server. For this step a tool such as svwar[89] can be used.

3. Registration
With the collected list of existing extensions from step 2, a tool like svcrack[89] can be used to “guess” the password of the extension by using a dictionary or brute force attack (trying all possible combinations of characters as the password) against the SIP server with REGISTER requests. As soon as successful registration confirmation is received from the attacked SIP server the extension and password combination is stored.
4. **Toll Fraud**

The attacker is in possession of the extension and password, and is able to register. Now it’s possible to call a premium international number set up by the attacker, and receive the toll from the telephone service provider. Without proper countermeasures or detection systems, the victim will only notice the fraud when they receive their next invoice from the telephone service provider, which could be days or even weeks later. Only when they see the invoice will they see the costs created by the attack. It’s also possible for the attacker to use the victim’s proxy server after registration, to run attacks on other targets, thus obfuscating the source of the attack.

The outcome of the study[88] could be used as input patterns and calibration for specialised UC intrusion detection/prevention systems. To mitigate this kind of attack, mutual TLS encryption together with strong passwords which are changed regularly for device Digest authentication should be used.

5.2.2 **ISDN Gateway Misuse**

The next example demonstrates a common problem, seen by the author several times. This is a problem caused by incorrect UC infrastructure security configuration even though the right components are in place and can result in telephone invoices for thousands of Euros. Unfortunately many customers are still using ISDN for voice or video calls in meetings. If video calls and the “special” ISDN 64 kbit/s B channel bounding required to provide the bandwidth for video, a specific ISDN E1/T1 card is built into the MCU directly. For voice only calls, a SIP/PBX gateway can be used, causing a similar problem illustrated in Figure 5.3 and described as follows:

![Figure 5.3: Example ISDN Gateway Misuse](image)

1. **SIP Scan**

   Organisation abc.com provides a business to business SIP video dial in option via a firewall and a dedicated SBC (see Figure 5.3 point 1). That means any
5.2 Real Life Examples

unauthenticated SIP UA is allowed to send a SIP INVITE message to the SBC and this message is processed by the SBC. A potential attacker could find out the SIP dial in point by conducting Internet research on the corporate home page, e-mail signatures, social engineering, or simply by querying the SIP service DNS records of the organisation’s domain name.

2. Virtual Meeting Room (VMR) Scan
The attacker already knows a valid virtual meeting room (VMR) number (similar to an extension) learned from an e-mail signature or by guessing. A SIP INVITE can then be sent to this SIP URI e.g. 10123@abc.com by the attacker. If this is a valid VMR, the call ends up on the MCU illustrated in Figure 5.3 on the facing page point 2.

3. ISDN Misuse
The attacker is now connected to the infrastructure of abc.com and could use DTMF tones to try to dial out via ISDN from the MCU, “inviting” somebody via ISDN into the same VMR. The number used for dialling could be a premium international number, maintained by the attacker, generating a telephone toll to abc.com indicated by Figure 5.3 on the preceding page point 3.

There are several options to reduce the risk of such ISDN service misuse. Turning off ISDN is generally not acceptable for various reasons e.g. security policy(!), business need, availability of appropriate network services. There is still a misleading security feature of ISDN such as caller identifier, which can easily be spoofed. Another problem with ISDN, is that it doesn’t allow IP based security services such as mutual TLS authentication or the usage of certificates. Allowing only authenticated users to send SIP INVITES is also not feasible, because this will prevent the communication with external organisations which are not managed by abc.com. A mutual TLS authentication between the abc.com SBC, business partners, and specific UAs on the Internet introduces the problem of certificate management and the required setup before a call can be established.

A more practical solution is the use of different PINs for each meeting, each with more than 5 digits, to be used as authentication during the join process of the user before the connection gets established with the VMR on the MCU. The PINs shouldn’t be included in meeting invites and a different PIN exchange mechanism with the participants should be used. In addition or instead of PINs a chairperson password can be used to enable all meeting participants only after one participant has provided the chairperson PIN when joining the VMR. It’s the chairperson’s responsibility to make sure only authorised participants are on the call. Another option, is to allow such unauthorised Internet calls only to a specific static VMR with specific security settings and limited services e.g. no ISDN/PSTN dial out.
5 Security Considerations for UC

5.3 UC Security Recommendations from Vendors

In addition to available standards and security services, each UC vendor provides some kind of recommendations and guides for their part of the UC “ecosystem” (products and services). This section of the document introduces some examples from well known industry vendors. Most of these recommendations apply in general for IT assets e.g. use a firewall, and some are more specific to UC.

The two guidelines Polycom Recommended Best Security Practices for Unified Communications[90] and Cisco Video and TelePresence Architecture Design Guide[91] recommend the following:

- Place all UC infrastructure components and endpoints behind a firewall and use a SBC for firewall traversal. Polycom’s SBC enterprise solution product is called RealPresence Access Director - RPAD and the deployment guide provides comprehensive direction.[92] Cisco is promoting their solution called VCS.[26]

- Conduct periodical security assessment vulnerability scans (internal and external) on UC components, update security configurations and apply software updates accordingly.

- Change default passwords and configuration items (as required)

- Turn off unused features/open ports, to reduce the amount of attack vectors. For example turn off H.323 when only SIP is used for signalling.

- For system management turn off protocols that are not required e.g. Telnet, and use secure protocol versions when they are used e.g. SNMP v2c or v3 instead of v1 (which uses plain text “passwords”), SSH, and HTTPS for management interface access.

- Disable Auto Answer on video endpoints. Auto answer is the ability for the video endpoint to automatically connect incoming calls without asking the user for permission. If enabled, this could allow somebody to call into a physical meeting room, where the endpoint is located, and listen/watch conversations without the knowledge of the people in the room.

- Use VPN connections for branch locations and remote access with video endpoints to provide a similar security and management level as at the main site of the organisation.

- Use complex passwords for user authentication/authorisation and possibly LDAP as back end to allow the end user to use the same password as for their computer login.

- Review and monitor the UC infrastructure generated log files together with the Call Detail Records - CDRs for anomalies of the UC usage e.g. registrations, unusual ISDN gateway calls (toll fraud).

- Use on the SBC security policy compliant settings to avoid unwanted connections and registration attempts from the Internet (toll fraud).
5.4 Conclusion

- Encryption for signalling and media should be used together with device/user authentication by utilizing technologies such as TLS, SRTP/SRTCP, and digital certificates.
- Incorporate the UC architecture into a security management system such as ISO 27001.
- Mobile UC devices should be managed centrally for video configuration and endpoint security via a secure channel e.g. HTTPS as used on Polycom’s RealConnect Resource Manager - RPRM[51] or Cisco’s TMS[52].
- Use a virtual meeting room - VMR for business to business or external guest user calls. Don’t allow call establishment from unknown sources directly to an endpoint inside the organisation.
- Separate the UC traffic on OSI layer 2 from data traffic by using Virtual LANs - VLANs and restrict communication from and to this VLAN.

5.4 Conclusion

In this chapter we discussed vulnerability issues that can cause organisations significant monetary losses and/or loss of reputation. There is to a certain level, an overlap of generic security requirements with other information technology areas. These items such as changing default passwords, usage of strong passwords etc. should be covered by an Information Security Management System (ISMS) like for other information technology assets. In addition there are some UC specific security requirements such as eavesdropping, toll fraud with ISDN/PSTN services, which are described in the vendor recommendations 5.3 on the facing page, the SIP standard 5.1.4 on page 66 and explained by real life examples. These threats need to be addressed, by for example, usage of PINs and VMRs, limitation of ISDN/PSTN gateway services and the implementation of an SBC to control the in and outbound UC Internet communication.

The SIP standard provides security services which needs to be implemented correctly in the right combination from registration to de-registration of endpoints. In the listed SIP standard security services in Table 5.1.4 on page 66, not the most appropriated security services are MUST e.g. mutual TLS, they are indicated as SHOULD or MAY and show a compromise. Media security is equally important and depends on a secured signalling communication path to limit attack vectors e.g. on key exchange. The available security services surrounding UC are sufficient to provide an acceptable security level, together with the vendor recommendations, those “tools” provide a good guidance and should be followed to enforce an organisations security policy and limit the vulnerability of an UC architecture.

Now we have discussed the theory, it’s important to take a look in the next chapter 6 on page 77 at the relevance of our research with regards to real life examples.
6 Case Studies

This Case Studies chapter demonstrates the protocols and security services provided so far in this document, using real products, mainly from Polycom. Most of the introduced features and configurations should be possible to replicate with other vendor’s products.

After an introduction of the test environment, some simplified “real life” examples are demonstrated, with a focus on firewall traversal and the security services used. The intention of the test cases is to set up the environment “as securely as possible” with the provided features, which is rather difficult to measure. Also, vendor security recommendations from 5.3 on page 74 are used where applicable and where useful for demonstration. Sometimes those configuration and security services are not practical and are not used or are replaced by “weaker” services in real projects for other particular reasons. One reason could be the burden of certificate management. Instead, pre shared keys are used for authentication, which introduces the requirement for key management, and the potential risk of key sharing among multiple devices. Specific UC settings, for example video resolutions, are not focused on unless it is relevant to security.

6.1 Test Environment

The test environment represents a small and simplified common enterprise UC deployment and it is used in this project for tests and analysis primarily focused on security. Figure 6.1 on the next page shows a fictitious organisation headquarters (HQ) with network segments separated by a firewall. These segments are: Inside 192.168.1.0/24, DMZ-private 192.168.254.0/24, DMZ-public 81.X.X.120/29. The communication from the Internet goes first to DMZ-public terminated by the SBC and with new sessions established by the SBC from DMZ-private to the Inside security zone. All communication from the Inside in the opposite direction to the Internet takes the same path but in reverse. In the Appendix 3 on page 103 the relevant Firewall HQ settings and rules are documented.

The branch and HQ locations both use different Internet connections and the Inside networks are separated by a firewall. For direct communication between those two sides the Internet is used, without any further “direct protected” IP connectivity e.g. MPLS WAN, VPN. In the branch location firewall, only NPAT is configured for all outbound traffic, which represents a classical Internet Service Provider (ISP) Customer-premises equipment (CPE) setup with a single dynamic public IP and no inbound port forwarding.

As a mobile user, an Android smartphone client is used with 4G ISP connectivity. Because in this environment, most of the traffic is encrypted, the packet capture is con-
Figure 6.1: Project Test Environment

ducted directly on the individual components e.g. SBC, for further analysis. There is a dedicated Wireshark capture instance running on the two HQ DMZs. For automatic SIP service registration, service records are specified on the public DNS server (for configuration details see Appendix 2 on page 102). Some general specifics of the environment are listed below:

1. Only IPv4 and SIP are used and analysed.
2. If possible, certificates, together with TLS are used and mutual authentication is turned on.
3. All relevant components (roomsystems, SIP Proxy Server, SBC, etc.) have certificates installed.
4. The SIP Proxy Server uses the following settings after installation (beside basic configuration items such as IP addresses, NTP server etc.):
   a) Accepts the two local SIP domains mobile-lab.local and mobile-lab.at only.
   b) Permits SIP TLS only on port 5061 and SIP port 5060 is turned off.
   c) Enforces Digest device authentication.
6.2 SIP Registration and Management Interface

d) Mutual TLS authentication with endpoints based on certificates.

Further details about the specific icons, components, software versions and functions are displayed in Appendix 1 on page 101.

6.2 SIP Registration and Management Interface

To ensure device authentication and encryption with the directly registered endpoints, the SIP Proxy - DMA is configured as shown in Figure 6.2 to allow only mutual TLS authenticated clients together with Digest pre-shared key authentication.

![Figure 6.2: SIP Proxy Settings](image)

On the roomsystem1 side, the certificate settings are shown in Figure 6.3 and forces the endpoint to validate the server certificate from the SIP Proxy and to verify the certificate from the web client used to manage the device via HTTPS/TLS. There are two certificates installed; server for the management web interface and client for the communication with the SIP Proxy. For roomsystem2 and roomsystem3, which are a more recent endpoint generation called Polycom GroupSeries, the settings are similar in the administration interface and are listed in Appendix 4 on page 105ff.

![Figure 6.3: roomsystem1 - HDX Certificate Settings](image)
roomsystem1 is configured in stand alone mode, not managed by the Video Resource Manager. Hence all settings need to be done directly on the system without centralised management. Figure 6.4 highlights the SIP settings of roomsystem1 to use TLS and device authentication with pre shared keys. The other two Polycom GroupSeries roomsystems are in dynamic mode, managed and provisioned from the Video Resource Manager. The equivalent roomsystem2 and roomsystem3 settings are listed in Appendix 4 on page 105ff.

![Figure 6.4: roomsystem1 - HDX SIP Settings](image)

In Figure 6.5 the registration message from roomsystem3 to the SIP Proxy Server is listed. Two noteworthy header details are the used protocol TLS on port 5061 and the Authorization: details, indicating Digest authentication. Details regarding the Digest SIP implementation are covered in Section 22 in RFC 3261[10].

```
REGISTER sip:mobile-lab.local SIP/2.0
Via: SIP/2.0/TLS 192.168.1.153:5061; branch=z9hG4bK1067525402-824; rport=5061
Max-Forwards: 70
Allow: INVITE, BYE, CANCEL, ACK, INFO, PRACK, COMET, OPTIONS, SUBSCRIBE, NOTIFY, MESSAGE, REFER, REGISTER, UPDATE
From: "roomsystem3" <sip:roomsystem3@mobile-lab.local>; tag=plcm_643951213-824; epid = 8213100FDF6ACV
To: <sip:roomsystem3@mobile-lab.local>
Call-ID: 643949876-824
CSeq: 6 REGISTER
Authorization: Digest username="roomsystem3@mobile-lab.local", realm="mobile-lab.local", nonce="MTQ1NDcwMTc3Mzg3Mg==", uri="sip:mobile-lab.local", response="e0012d19c1301d67c045f0b98ae89659", algorithm=MD5, cnonce="3e191ee9", qop=auth, nc = 00000005
/* further details omitted
```

![Figure 6.5: roomsystem3 REGISTER](image)
6.2 SIP Registration and Management Interface

The respond SIP 200 OK message in Figure 6.6 from the SIP Proxy Server represents a successful registration for roomsystem3. Both messages are retrieved from the SIP Proxy Server internal log files to be able to read them in unencrypted form.

SIP /2.0 200 OK
CSeq: 5 REGISTER
Call-ID: 643949876−824
From: "roomsystem3" <sip:roomsyststem3@mobile−lab.local>;tag=plcm_643951213−824;epid =8213100FD6ACV
To: <sip:roomsyststem3@mobile−lab.local>;tag=a9b
Via: SIP /2.0/TLS 192.168.1.153:5061;branch=z9hG4bK1050660084−824;rport=5061
Contact: <sip:roomsyststem3@192.168.1.153:5061;transport=tls>;+sip.instance="urn:uuid:d474c94c−baff−552d−a565−bde48321c35d">;expires=150
Date: Fri, 05 Feb 2016 19:56:18 GMT
Content−Length: 0

Figure 6.6: roomsyststem3 REGISTER OK

The firewall in the branch location only supports NAPT and no further firewall traversal functionalities. roomsystem2 is configured in dynamic-mode, as described in 4.4 on page 44, using the SBC at the HQ location for registration, provisioning and presence information illustrated in Figure 6.7.

Figure 6.7: roomsystem2 - Registration in dynamic mode

The SBC provides firewall traversal functionality for four protocols: SIP, XMPP (pres-
ence), HTTP/TLS (provisioning), and LDAP (address book) to and from roomsystem2. Communication requests are received on the Internet side of the SBC and applies (recommended) access control functions (ACL) e.g. SIP header field checks, authentication and certificate validation of external systems based on a rule set defined by an organisation’s security policy. After a successful validation, new connections are established (proxied) to the HQ location infrastructure components e.g. SIP proxy and Resource Manager. The SBC acts as a B2BUA. The configuration details for the SBC SIP and Access Proxy settings are shown in Appendix 5 on page 107 and further information can be found in the Polycom RPAD deployment guide[92].

As indicated in section SIP Security Services Implementation 5.1.4 on page 66, it’s critical to ensure authentication, authorization and confidentiality starting from the registration process, and to reuse the established communication channels e.g. TLS. From the experience of the author, observed in many implemented projects, often only encryption is enabled, together with “out-of-the-box” self signed certificates. This misleads the implementer into thinking they have a “secure system”, because it indicates that everything is encrypted on the endpoint or infrastructure. In reality, the system is vulnerable, for example to man-in-the-middle, integrity and information leakage attacks. It is recommended (see 5.3 on page 74) that pre shared key Digest authentication is used, together with TLS. Many administrators don’t “like” the additional required pre shared key management or for mutual authentication with TLS certificate management.
6.3 SIP Based Firewall Traversal Calls via SBC

6.3.1 Call Scenario: roomsystem2 → roomsystem3 (P2P)

The first call scenario describes roomsystem2, located in the branch location and registered at the HQ infrastructure via the SBC, initiating a point to point (P2P - direct call) to roomsystem3 located at the HQ. Both endpoints are registered as described in 6.2 on page 79 and the call flow is illustrated in Figure 6.8 below.

![Call Scenario Diagram](image_url)

Figure 6.8: Call Scenario: roomsystem2 → roomsystem3 (P2P)

The SBC receives the SIP INVITE from roomsystem2 and permits the “internal” (endpoints administered by the same organisation) call by proxying the call to the SIP proxy based on its ACL settings. Only registered and authenticated external endpoints should be allowed to initiate direct inbound P2P calls. Unknown participants, not managed by the organisation, should only be allowed to dial into a VMR protected with a PIN as described in 5.3 on page 74. The SIP proxy verifies its dial plan and locates roomsystem3 as a local registered device and proxies the request to roomsystem3. After the RTP media is negotiated via SIP/SDP, the media stream is established directly via the SBC acting as a media relay. On the HQ side, only the SBC DMZ-PRIVATE IP address (192.168.254.125) is exposed to the infrastructure. In Figure 6.9 on the following page the initial phase of the SIP signalling is demonstrated, extracted from the SIP proxy log files.
Figure 6.9: Call Scenario: roomsystem2 → roomsystem3 (P2P) signalling
6.3 SIP Based Firewall Traversal Calls via SBC

6.3.2 Call Scenario: mobile and roomsystem2 → VMR (P2M)

The following scenario demonstrates two calls initiated from roomsystem2 and mobile.android dialling into VMR 1001 managed by the SIP Proxy and hosted on the MCU (see Figure 6.10). Both endpoints are registered on and managed by the HQ infrastructure, authenticated and using encryption for signalling and media as described in 5.1.4 on page 66. Appendix Figure 7 on page 106 shows the call statistic of roomsystem2 including the encryption protocol used.

![Call Scenario Diagram](image)

Figure 6.10: Call Scenario: mobile.android and roomsystem2 → VMR (P2M)

The SIP signalling and media is proxied via the SBC, and ACLs can be enforced. In this scenario as many participants can join the conference as required, regardless of their signalling and media type as long as enough bandwidth and MCU resources are available. Of course the used technology and transcoding needs to be supported by the infrastructure, especially the MCU. A PIN can be assigned on the SIP proxy for this VMR to add additional security, authentication of the participants and not the device. This VMR concept also allows the invitation of external participants, not managed by the organisation, and enables collaboration with them.
In Figure 6.11 the initial SIP signalling between SBC, SIP Proxy and MCU is highlighted. The by mobile.android initiated call is a separate “call leg” and not shown here.

Figure 6.11: Call Scenario: roomsystem2 → roomsystem3 (P2P) signalling
6.3.3 Call Scenario: roomsystem2 → mobile.android (P2P)

In this scenario, which is very similar to the previous one, the main difference is that the call is a point to point (P2P) call, roomsystem2 calling mobile.android. All the registration, management and signalling goes back to the HQ infrastructure and the SIP Proxy is involved to locate mobile.android as a local registered endpoint, both located outside of the HQ on the Internet and in the branch office. Because it’s a P2P call, no MCU resources are required.

![Call Scenario Diagram]

The signalling is proxied (see Figure 6.13 on the next page for more details) from the SBC to the SIP Proxy and the SIP Proxy sends a new SIP INVITE back to the SBC with the SIP URI:

```
INVITE sip:mobile.android-45078746275119@192.168.254.125:5071;
transport=tls SIP/2.0
```

All the call media (RTP) traffic is proxied by the SBC directly between the two participants and doesn’t reach the HQ infrastructure, hence the media traverses the HQ...
Internet connection twice. Only the SIP signalling part reaches the SIP Proxy and can enforce a centralised security policy for registered endpoints.

Figure 6.13: Call Scenario: roomsystem2 → mobile.android (P2P) signalling

6.4 Conclusion

This chapter demonstrates the implementation of standards and vendor recommendations into real products and proves that more than the “must” security services can be integrated and provide an architecture that is still usable and administrable. It’s important to implement the available security services correctly, beginning at the registration process. A negative example is the usage of encryption for confidentiality, but without enforcing authentication between the two communication parties.
7 Conclusions and Key Findings

7.1 Introduction and Overview

Field studies\cite{11, 12} show that technologies such as IP Telephony, IM/Presence and Audio/Web conferencing, are deployed but not integrated together to form a proper Unified Communication (UC) architecture. Most of these technologies are distributed on organization’s premises and are generally managed individually by external 3rd parties\cite{11}, and as such have limited integration points. UC is becoming a settled IT area with a steady growth and the trend is to reduce costs and to move the infrastructure to “cloud” hosted services. New technologies such as WebRTC, which transforms the web browser to a UC interface embedded into web applications are supporting this trend 3.4 on page 21.

Some UC building blocks are not deployed widely enough yet, such as video conferencing, perhaps due to the network and UC infrastructure requirements for real time traffic. The special need on bandwidth, RTT, Jitter and QoS on network infrastructure is an often underestimated factor for the success and adoption rate for UC architectures. In addition there are some UC specific components such as MCUs, call control and endpoints required which are specially built for UC and add some level of complexity. The possible complications surrounding the interoperability between different vendors is still a challenge and the importance of forums such as the Unified Communication Interoperability Forum (UCIF) are significant in making UC “as easy as a telephone call”. It is important to have a focus on these requirements, for further details see 2.3 on page 10 and 3.2 on page 16, during the design and implementation of such UC environment to build a robust, easy to use and at the same time secure UC solution.

7.2 Protocols and Standards

In chapter 4 on page 25 Protocols and Related Standards, one of the main areas focused on, is that of SIP \cite{10} as a signalling protocol and the affiliated protocols such as SDP\cite{18}, RTP\cite{19}. SIP itself doesn’t provide services, instead it provides a framework of primitives to implement different services independently of the used transport protocol. This flexibility is perhaps one of the biggest success factors for SIP, but on the other hand it comes with a “jungle” of RFCs and extensions.
7 Conclusions and Key Findings

Firewalls, NAT and the Traversal Problem

Firewalls are still an integral factor for enforcing security policies at network perimeters, for example between private organisation internal LANs and the Internet. The two main problems for SIP and UC in general are traversing firewall packet filtering and Network Address Translation (NAT), getting SIP itself passing the firewall/NAT and the second most challenging issue is to get the media session to pass through. These problems are described in detail in section 4.6 on page 51. ALGs (see 4.5 on page 46), usually built into firewalls, are not a reliable solution, because they need to be kept updated with standard changes, don’t cover all UC aspects e.g. content sharing sessions and can have a negative impact on performance.

UC traffic will not work across firewalls and NAT boundaries without special solutions. NAT causes the following major issues on UC: the SIP Via: and Contact URI header field and the SDP information attributes c= and m= lines contain private IP addresses. There are workarounds for SIP, but not for the media session see 4.6 on page 51 for more details. The best solution for NAT traversal is to use hole punching in combination with ICE and in case that’s not successful fall back to TURN as described in section 4.6 on page 51. A very interesting outcome of the Ford study[69] is: About 82% of the NATs tested supported hole punching for UDP across the tested router/firewall and operating systems.

For service provider or larger organisation needs, relying on hole punching, ICE, STUN and TURN is not sufficient and the usage of an SBC is the right way to address the problem. There is some debate about using SBCs and their B2BUA behaviour because of the “breaking the end-to-end nature”[75] of SIP, but it solves the problem surrounding firewalls and NAT. The SBC acts on the public Internet facing side as a SIP user agent and handles incoming UA sessions, applies policies, processes the sessions, possibly relays the media and “proxies” the session into the private network as described in section 4.7 on page 58.

7.3 Security Considerations for UC

There is a certain overlap regarding the security requirements for UC and “regular” IT areas such as authentication, authorization and confidentiality, which should be integrated and covered by an Information Security Management System (ISMS). Some more general mitigating examples are changing default passwords, conducting security penetration tests, turning off unused services, using secure protocols such as SSH and TLS for management sessions and many more listed by vendor recommendations in section 5.3 on page 74. There are also some UC specific security requirements such as using a VMR together with a PIN to authenticate a participant in a UC session or limiting access to ISDN/PSTN gateway services to avoid toll fraud. Real life examples surrounding these issues are provided and explained in section 5.2 on page 71.
It’s important to ensure that the available security services surrounding UC are implemented correctly and in the right combination, from registration through to deregistration. In section 5.1.4 on page 66 the recommended SIP standard security services are listed and Figure 5.1 on page 66 shows that not the most appropriated or secure services are MUST e.g. mutual TLS authentication and indicates a compromise of minimum requirements. It’s obvious that not only the SIP signalling needs to be protected, the media security is equally important and described in detail in section 5.1.5 on page 69. An example of incorrect UC security implementation, often seen by the author, is to use encryption services for confidentiality without ensuring authentication and authorization of communication participants upfront. The provided security services surrounding UC and described in this document are sufficient to achieve an acceptable security level, which of course depends on the specified security policy, but it’s a good foundation.

For the firewall traversal and NAT problem, a vendor recommendation and best practise is to use a SBC together with an appropriate rule set to enforce the organisations security policy without preventing the usability, to avoid unwanted connections and registration attempts from the Internet including ISDN/PSTN toll fraud. The functionality of SBCs is explained in more detail in section 4.7 on page 58 and demonstrated with examples in the Case Studies 6.3 on page 83.

7.4 Case Studies

In chapter 6 on page 77 Case Studies of this dissertation, implementation examples prove, that standard requirements from, for example SIP RFC3261[10] surrounding security can be integrated into real life products including vendor security related recommendations. The provided and recommended security services mitigate the main security concerns listed in 5.1 on page 61. A recommended approach used in this project was the usage of mutual TLS authentication with certificates, beginning at the registration process, in combination with Digest device authentication to ensure authentication, authorization and confidentiality and limiting the vulnerability against DoS attacks. For media security, encryption has been enabled and the required key and parameter exchange has been conducted by the already established and protected signalling channel. Details regarding the setup and configuration can be found at 6.2 on page 79 and 6.3 on page 83.

7.5 Future Work

The scope of this dissertation has to have some limits and there are several points which could be expanded. For example:

- Cover the alternative protocol H.323, which is significant because of the still wide deployment for signalling and compare it to SIP’s firewall traversal approaches.
- Include some additional Case Studies such as WebRTC, Microsoft SfB, because of the current industry trend for relayed instead of transcoded calls and the fast adoption rate by the market noticed by the author.
7 Conclusions and Key Findings

- Highlight the integration between different technologies such as traditional SIP/H.323 based UC architectures and Microsoft SfB and WebRTC.
- Compare the different vendor approaches for firewall traversal including advantages and disadvantages e.g. SBCs in traditional and service provider environments compared with Microsoft's Edge/Frontend Server and Federations approach.
- The protocol section could be expanded to investigate some of the alternative more exotic protocols such as SCTP or ZRTP that are less commonly used. There are some advantages with these alternative protocols, but unfortunately for the moment they are not emerging.
Bibliography


Bibliography


Appendix

1 Test Environment Components

The icons used in the figures in this document are provided by Polycom.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Component Name, Type, and Version</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Firewall HQ | Firewall HQ  
Juniper  
SSG20 - V6.3.0r21.0 |          |
| Firewall Branch Office | Firewall Branch  
Huaway  
WebGate3 | NAPT |
| SIP Proxy Server | SIP Proxy Server  
Polycom  
DMA - V6.3 | local domains are:  
mobile-lab.local  
and mobile-lab.at |
| MCU | MCU  
Polycom  
RMX - V8.6 |          |
| SBC | SBC  
Polycom  
RPAD - V4.2.2 |          |
| Video Resource Manager | Video Resource Manager  
Polycom  
Resource Manager - V9.0 | Provides provisioning (HTTPS), address book (LDAP), presence (XMPP), and management functionality |

Table 1: Test Environment Components 1 of 2
Appendix

<table>
<thead>
<tr>
<th>Icon</th>
<th>Component Name, Type, and Version</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Desktop System Icon" /></td>
<td>Desktop System&lt;br&gt;Polycom&lt;br&gt;Real Presence Desktop - V3.5</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Room System1 Icon" /></td>
<td>roomsystem1&lt;br&gt;Polycom&lt;br&gt;HDX 8000 - V3.1.9</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Room System2 Icon" /></td>
<td>roomsystem2&lt;br&gt;Polycom&lt;br&gt;Group Series 500 - V5.1</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Room System3 Icon" /></td>
<td>roomsystem3&lt;br&gt;Polycom&lt;br&gt;Group Series 500 - V5.1</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Mobile Android Icon" /></td>
<td>mobile.android&lt;br&gt;Polycom&lt;br&gt;RealPresence Mobile - V3.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Test Environment Components 2 of 2

## 2 Public DNS Server

The DNS zone file below, from the public DNS server, specifies the service records required for automatic service detection on the Internet, public facing DNS.

```plaintext
$TTL 600 ; 10 min default TTL
$ORIGIN mobile-lab.at.
@ IN SOA ns1.mobile-lab.at. admin.mobile-lab.at. ( 2015120305 ; Serial 10800 ; Refresh 3600 ; Retry 604800 ; Expire 300 ; Negative Response TTL )

; DNS Servers
IN NS ns1.mobile-lab.at.
IN NS ns2.mobile-lab.at.

; Machine Names
localhost IN A 127.0.0.1
ns1 IN A 81.X.X.122
ns2 IN A 81.X.X.122
```
3 Firewall HQ Configuration Details

Any SIP helper features are turned off on the firewall and the following access control lists for the different security zones are setup with the specified services below:

- Polycom-HTTPS-Special" protocol tcp src-port 0−65535 dst-port 8443−8443
- Polycom-HTTPS-Special" + tcp src-port 0−65535 dst-port 9443−9445
- Polycom-HTTPS-Special" timeout 5
- RDP" protocol tcp src-port 0−65535 dst-port 3389−3389
- Polycom_PD_Licensing" protocol tcp src-port 0−65535 dst-port 9333−9333
- Polycom_PD_Licensing" + tcp src-port 0−65535 dst-port 3333−3333
- SIP-TLS" protocol tcp src-port 0−65535 dst-port 5061−5061
- Media WAN Inbound" protocol udp src-port 1023−65535 dst-port 20002−30001
- Media WAN Outbound" protocol udp src-port 20002−30001 dst-port 1023−65535
- Media LAN Inbound" protocol udp src-port 40002−50001 dst-port 1023−65535
- Media LAN Inbound" + tcp src-port 16001−17000 dst-port 1023−65535
- Media LAN Outbound" protocol udp src-port 1023−65535 dst-port 40002−50001
- Media LAN Outbound" + tcp src-port 1023−65535 dst-port 16001−17000
- SBC Access Proxy WAN" protocol tcp src-port 1023−65535 dst-port 443−443
- SBC Access Proxy WAN" + tcp src-port 1023−65535 dst-port 389−389
- SBC Access Proxy WAN" + tcp src-port 1023−65535 dst-port 5222−5222
- XMPP" protocol tcp src-port 0−65535 dst-port 5222−5222
- SIP-SBC" protocol tcp src-port 0−65535 dst-port 5070−5070
- SIP-TLS-SBC" protocol tcp src-port 0−65535 dst-port 5071−5071

Figure 1: Firewall Rules: Internet → DMZ-PUBLIC
Figure 2: Firewall Rules: DMZ-PUBLIC → Internet

<table>
<thead>
<tr>
<th>ID</th>
<th>Source</th>
<th>Destination</th>
<th>Service</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>srv.mobile-lab.at</td>
<td>Any-IPv4</td>
<td>DNS HTTP ICMP-ANY</td>
<td>✓</td>
</tr>
<tr>
<td>28</td>
<td>sbc.mobile-lab.at</td>
<td>Any-IPv4</td>
<td>SIP SIP-TLS</td>
<td>✓</td>
</tr>
<tr>
<td>32</td>
<td>sbc.mobile-lab.at</td>
<td>Any-IPv4</td>
<td>Media WAN Outbound</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>Any-IPv4</td>
<td>Any-IPv4</td>
<td>ANY</td>
<td>✗</td>
</tr>
</tbody>
</table>

Figure 3: Firewall Rules: DMZ-PRIVATE → Inside

<table>
<thead>
<tr>
<th>ID</th>
<th>Source</th>
<th>Destination</th>
<th>Service</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>sbc.mobile-lab.at Management</td>
<td>ad01.mobile-lab.local</td>
<td>DNS LDAP ICMP-ANY</td>
<td>✓</td>
</tr>
<tr>
<td>27</td>
<td>sbc.mobile-lab.at Management</td>
<td>pd-rp1.mobile-lab.local</td>
<td>ICMP-ANY Polycom_PDI_licensing SNMP</td>
<td>✓</td>
</tr>
<tr>
<td>29</td>
<td>sbc.mobile-lab.at Signaling</td>
<td>rm.mobile-lab.local</td>
<td>HTTPS LDAP XMPP</td>
<td>✓</td>
</tr>
<tr>
<td>31</td>
<td>sbc.mobile-lab.at Signaling</td>
<td>dma-a-node1.mobile-lab.local SIP SIP-TLS</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>sbc.mobile-lab.at Signaling</td>
<td>Any-IPv4</td>
<td>Media LAN Inbound</td>
<td>✓</td>
</tr>
<tr>
<td>36</td>
<td>shark.mobile-lab.local</td>
<td>Any-IPv4</td>
<td>DNS HTTP HTTPS ICMP-ANY</td>
<td>✗</td>
</tr>
<tr>
<td>22</td>
<td>Any-IPv4</td>
<td>Any-IPv4</td>
<td>ANY</td>
<td>✗</td>
</tr>
</tbody>
</table>

Figure 4: Firewall Rules: Inside → DMZ-PRIVATE

<table>
<thead>
<tr>
<th>ID</th>
<th>Source</th>
<th>Destination</th>
<th>Service</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Any-IPv4</td>
<td>Any-IPv4</td>
<td>DNS HTTP HTTPS ICMP-ANY Polycom HTTPS-Special Polycom_PDI_licensing RDP SNMP</td>
<td>✓</td>
</tr>
<tr>
<td>34</td>
<td>dma-a-node1.mobile-lab.local</td>
<td>sbc.mobile-lab.at Signaling</td>
<td>Media LAN Outbound SIP-SBC SIP-TLS-SBC</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>Any-IPv4</td>
<td>Any-IPv4</td>
<td>ANY</td>
<td>✗</td>
</tr>
</tbody>
</table>
4 Endpoint Configuration and Call Details

![GroupSeries SIP Settings](image)

Figure 5: GroupSeries SIP Settings roomsystem3
Figure 6: GroupSeries Certificate Settings roomsystem3

Figure 7: roomsystem2 call statistics in VMR call
5 SBC Configuration Details

![SBC SIP Signalling settings](image)

**Figure 8: SBC SIP Signalling settings**

![SBC Access Proxy settings](image)

**Figure 9: SBC Access Proxy settings**