13 Signaling Protocols for Multimedia Communication*

13.1 Signaling and Sessions

13.2 SIP Basics

13.3 Instant Messaging Protocols

13.4 WebRTC

Literature:
Stephan Rupp, Gerd Siegmund, Wolfgang Lautenschlager: SIP – Multimediale Dienste im Internet, dpunkt.Verlag 2002


* = Chapter 13 is not mandatory for minor subject students!
Signaling

- Signaling (Signalisierung, Zeichengabe) originates from circuit-switched networks
- Signaling = Protocols to control connections (network control plane)
  - User-to-Network
  - Network-to-Network
  - End-to-End
- Session control:
  - Managing participants and connections of a (conference) session
  - Monitoring of quality
- Call control:
  - Set up/join/tear down
  - Negotiation of capabilities
  - Adaptation to network traffic situation
Call and Session Signaling in H.32X Standards

- **H.323**: ITU-T standard “Visual Telephone Terminals over Non-Guaranteed QoS Service LANs”
- **H.324**: ITU-T standard “Terminal for Low Bit-Rate Multimedia Communication”
- **H.225**
  - Call signaling and RAS (Registration, Admission, Status) over non-QoS networks
  - Additional protection and recovery mechanisms on top of H.320
- **H.245**
  - Control protocol for multimedia
  - Information exchange about terminal capabilities (e.g. codecs, ports)
  - Negotiation of logical channels between terminals
  - Can be “tunneled” through H.225 (firewalls)
Network Architecture Option 1: H.32X Based

Source: Cisco
Network Architecture Option 2: Skype Based

Figure 1: Skype Network

Source: fortinet.com
Network Architecture Option 3: SIP Based

Source: tml.tkk.fi
13 Signaling Protocols for Multimedia Communication

13.1 Signaling and Sessions

13.2 SIP Basics

13.3 Signaling for Instant Messaging

13.4 WebRTC

Literature:
Stephan Rupp, Gerd Siegmund, Wolfgang Lautenschlager: SIP – Multimediale Dienste im Internet, dpunkt.Verlag 2002

SIP – Components and Features

- SIP = *Session Initiation Protocol*, by IETF
  - Mark Handley and Henning Schulzrinne, 1999
    (MMUSIC WG = Multiparty Multimedia Session Control)
    - Independent of underlying network technology
    - Similar to HTTP
    - Covers broad range from traditional telephony to multimedia conferencing
    - Peer-to-peer style architecture:
      » Client contains *User Agent* (UA) in client and server roles (UAC, UAS)
- Proxy, Redirect and Location Servers for forwarding of control messages
- User, terminal and service mobility
- Gateways to traditional networks (e.g. telephone networks)
- Status observation for users and terminals (e.g. online/offline, busy/free)
Addressing in SIP

- SIP supports various address formats including addresses based on phone numbers
  - ITU standard for international phone number format: E.164
- Email style addresses:
  - sip:Heinrich.Hussmann@ifi.lmu.de
- IP-based addresses:
  - sip:hussmann@141.84.8.6
- Phone number style addresses:
  - sip:+49-89-2180-4650@net2phone.com
- Mapping of E.164 telephone numbers to IP domain names
  - +49-89-2180-4650 is mapped to domain name 0.5.6.4.0.8.1.2.9.8.9.4.E164.arpa
- IP-based addressing of terminals is a potential problem
  - Many large sites use NAT (network address translation)
SIP Messages

• Modeled after HTTP
  – Header: Connection parameters and service information
  – Body: Description of connection (using Session Description Protocol SDP)

• Requests:
  – From client (agent) to server (agent)
  – INVITE, BYE, OPTIONS, STATUS, CANCEL, ACK, REGISTER, ...

• Responses:
  – Status information, e.g.
    » Informational: 100 Trying, 180 Ringing, 181 Call is forwarded, ...
    » Success: 200 OK
    » Redirection: 300 Multiple Choices, 301 Moved Permanently, ...
    » Client Error: 400 Bad Request, 404 Not Found, 486 Busy Here, ...
    » Server Error: 500 Internal Server Error, 504 Gateway Timeout, ...
Call Setup by Three-Way Handshake

- Direct connection establishment between two SIP terminals (user agents)
## Example: SIP Message

<table>
<thead>
<tr>
<th><strong>Start Line</strong></th>
<th><strong>General Header</strong></th>
<th><strong>Sequence Number</strong></th>
<th><strong>Request Header</strong></th>
<th><strong>Entity Header</strong></th>
<th><strong>Body: SDP Data</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INVITE sip: <a href="mailto:john@domain.com">john@domain.com</a> SIP/2.0</strong></td>
<td><strong>VIA: SIP/2.0/UDP 169.130.12.5</strong></td>
<td></td>
<td><strong>Call-ID: <a href="mailto:187602141351@worchester.bell-telephone.com">187602141351@worchester.bell-telephone.com</a></strong></td>
<td><strong>From: <a href="">sip:a.g.bell@bell-telephone.com</a></strong></td>
<td><strong>Subject: Mr. Watson, come here</strong></td>
</tr>
<tr>
<td></td>
<td><strong>To: T.A.Watson<a href="">sip:watson@bell-telephone.com</a></strong></td>
<td></td>
<td></td>
<td><strong>Content-Type: application/sdp</strong></td>
<td><strong>Content-Type: application/sdp</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Content-Length: 885</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>v=0</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>v=0</strong></td>
</tr>
<tr>
<td></td>
<td><strong>o=bell 536557652353687637 IN IP4 128.3.4.5</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>o=bell 536557652353687637 IN IP4 128.3.4.5</strong></td>
</tr>
<tr>
<td></td>
<td><strong>c=IN IP4 135.180.144.94</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>c=IN IP4 135.180.144.94</strong></td>
</tr>
<tr>
<td></td>
<td><strong>m=audio 3456 RTP/AVP 0 3 4 5</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>m=audio 3456 RTP/AVP 0 3 4 5</strong></td>
</tr>
</tbody>
</table>
Session Description Protocol (SDP)

ο (origin) parameter:
ο=<username> <session id> <version>
<network type> <address type> <address>

session id, version: NTP timestamp
network type IN = Internet
address type IP4 or IP6

v=0
ο=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000

Example from RFC 4566
SDP Media Description and Attributes

• Media description (m)
  – Media type (e.g. audio)
  – Used port number
  – User data transport protocol
    » e.g. RTP/AVP = Real-Time Transport Protocol, Audio/Video Profile
  – List of available formats/codecs
    » "99" in previous example, may be a list of options

• Attribute description (a)
  – Codec details for all mentioned media formats
  – E.g. from "rtpmap" in RTP/AVP standard (IETF RFC 3551)
Example for Multiple Media Formats

m=audio 2410 RTP/AVP 0 8 3 4
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:4 G723/8000

- Communication partner announces the codecs/formats which are locally supported
- Standardized list of RTP-Codecs in RTP/AVP standard, excerpt:

<table>
<thead>
<tr>
<th>Payload type</th>
<th>Encoding name</th>
<th>Media type</th>
<th>Clock rate</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PCMU (µ-law)</td>
<td>A</td>
<td>8000</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>reserved</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>reserved</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>GSM</td>
<td>A</td>
<td>8000</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>G723</td>
<td>A</td>
<td>8000</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>DVI4</td>
<td>A</td>
<td>8000</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>DVI4</td>
<td>A</td>
<td>16000</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>LPC</td>
<td>A</td>
<td>8000</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>PCMA (a-law)</td>
<td>A</td>
<td>8000</td>
<td>1</td>
</tr>
</tbody>
</table>
Codec Negotiation by Offer&Answer

• In connection establishment dialogue (3-way handshake):
  – Partner A sends *offer* (list of supported codecs) as SDP part of *INVITE*
  – Partner B selects appropriate options and specifies them as SDP part of *OK*

• Example:
  – Offer:
    \[ m=audio 2410 \text{ RTP/AVP 0 8 3 4} \]
  – Answer:
    \[ m=audio 2468 \text{ RTP/AVP 0 3} \]

• Analogous negotiation for multiple media channels
  – E.g. audio + video
  – E.g. chat, possibly encrypted
  – E.g. file transfer
Example: Parallel Call Forking (e.g. Call Center)

Continued in right column
13 Signaling Protocols for Multimedia Communication

13.1 Signaling and Sessions
13.2 SIP Basics
13.3 Signaling for Instant Messaging
13.4 WebRTC

Literature:
www.ietf.org/impp
www.xmpp.org
Signaling for Instant Messaging

• Proprietary protocols for specific services!
• Several efforts for standardization, two important examples:
  • SIMPLE
    (SIP for Instant Messaging and Presence Leveraging Extensions)
    – RFCs 3428, 3856, 3863, 4479, ... and many drafts
    – Messaging as extensions of the SIP protocol
    – No multimedia support, just text messages
  • XMPP (Extensible Messaging and Presence Protocol)
    – Standardized form of XML-based streaming and presence protocols
devolved by the “Jabber” community (since 1999)
    – IETF standardization, most important: RFCs 6120-6122 (2011)
    – Presence, authentication, encryption, multi-user chat, privacy blocking
    – Support from commercial IM applications currently decreasing
      » e.g. Facebook Chat XMPP Interface (2010-2015),
      Google Talk (ended Feb 2015)
XMPP

- Based on generic transport protocol for XML streams over the Internet
- Idea:
  - Two-way exchange of XML files of potentially infinite length
  - Transmission of discrete semantic units (*XML stanzas*)

```
<stream>
  <presence>
    <show/>
  </presence>
  <message to='foo'>
    <body/>
  </message>
  <iq to='bar'>
    <query/>
  </iq>
  ...
</stream>  iq = info/query
```
XMPP Example

C: <xml version='1.0'?><stream:stream
    to='example.com'
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'
    version='1.0'>

S: <xml version='1.0'?><stream:stream
    from='example.com'
    id='someid'
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'
    version='1.0'>

... encryption, authentication, and resource binding ...

C: <message from='juliet@example.com'
            to='romeo@example.net'
            xml:lang='en'>
    <body>Art thou not Romeo, and a Montague?</body>
</message>

C: <message from='juliet@example.com'
            to='romeo@example.net'
            xml:lang='en'>
    <body>Neither, fair saint, if either thee dislike.</body>
</message>

S: </stream:stream>
13 Signaling Protocols for Multimedia Communication

13.1 Signaling and Sessions

13.2 SIP Basics

13.3 Signaling for Instant Messaging

13.4 WebRTC

Literature:
www.webrtc.org
www.html5rocks.com/en/tutorials/webrtc/basics/

Web Real Time Communications: Why and How?

• State of the art of multimedia conferencing around 2011:
  – Proprietary applications (e.g. Skype)
  – Proprietary plug-ins (e.g. Google Talk plug-in for Hangouts)
  – Difficult to integrate with Web applications

• Goal:
  – Voice, video and data communication in (almost) real-time
  – Based on standard browser software only (no plug-ins)

• Driver:
  – Google acquisition of company "GIPS"
  – Open standard alliance “WebRTC” driven by Google, incorporating W3C and IETF

• State of the art 2016:
  – Support in current browser versions of Chrome, Firefox, Opera, Edge
  – Used e.g. within WhatsApp, Facebook Messenger, TokBox, appear.in, Unify Circuit, …
Try it out: https://apprtc.appspot.com/
WebRTC Main APIs

- **MediaStream** *(aka getUserMedia)*
  - Obtain synchronized media streams
  - Mainly from camera and microphone

- **RTCPeerConnection**
  - Audio and video connection establishment
  - Arbitrary choice of signaling protocol (e.g. SIP, XMPP, AJAX requests)
  - Media capability negotiation (offer, answer) based on SDP
  - Encryption
  - Bandwidth management

- **RTCDataChannel**
  - Peer-to-peer exchange of application data
  - Similar to (and compatible with) WebSockets
Example: getUserMedia

costants = {
    audio: false;
    video: true
};
navigator.mediaDevices.
    getUserMedia(
        constraints,
        gotIt);
function gotIt(stream) {
    var videoTracks = 
        stream.videoTracks;
    ... 
};
Internet Connectivity Establishment Framework (ICE)

- Establishment of connections between peers in presence of address translations (NAT)
  - First option: direct
  - Second: Obtain public IP address through STUN (Session traversal utilities for NAT)
  - Third: Use relay server with TURN (Traversal using relays around NAT)

Source: html5rocks.com
14 Visions and Outlook

14.1 Innovation and Prognoses

14.2 Trends and Visions
Inheritance Lines:
Example iPad

- Alan Kay: DynaBook 1968
- Go Technology EO 1993
- Wellner: Multitouch gestures 1991
- Touchpad gestures 1993
- Fingerworks 2005
- Apple Newton 1993
- Microsoft Tablet PC 2002
- iPad 2010
- Univ Toronto (Lee, Buxton, Smith): Multitouch Tablet 1985
- iPhone 2007
Prognosis for 2014 from the Year 1964

August 16, 1964

Visit to the World's Fair of 2014

By ISAAC ASIMOV

• “By 2014, electroluminescent panels will be in common use. Ceilings and walls will glow softly, and in a variety of colors that will change at the touch of a push button.”

• “Gadgetry will continue to relieve mankind of tedious jobs. Kitchen units will be devised that will prepare "automeals," heating water and converting it to coffee; toasting bread; frying, poaching or scrambling eggs, grilling bacon, and so on.”

• “Robots will neither be common nor very good in 2014, but they will be in existence. General Electric at the 2014 World's Fair will be showing 3-D movies of its "Robot of the Future," neat and streamlined, its cleaning appliances built in and performing all tasks briskly.”

• “The appliances of 2014 will have no electric cords, of course, for they will be powered by long-lived batteries running on radioisotopes.”

• “Much effort will be put into the designing of vehicles with "Robot-brains" - vehicles that can be set for particular destinations and that will then proceed there without interference by the slow reflexes of a human driver.”

• “Communications will become sight-sound and you will see as well as hear the person you telephone. The screen can be used not only to see the people you call but also for studying documents and photographs and reading passages from books.”
Work-Study Position in Munich
During Lecture Season with an option to continue during the time between terms

Scope of Work
As a software company delivering communication and collaboration solutions for enter-
prises we responsible-minded manage our software inventory, e.g. with respect to soft-
ware licensing and vulnerability management. For our software inventory database we
want to support the various stakeholders within the company with role-dependent workflows
applying user-centric design.

Your Working Environment
As an applicant to this position you will work in an international context and get valuable
insights to advanced management of software regarding the integration of 3rd party soft-
ware or Open Source Software (OSS) and leading edge software vulnerability manage-
ment.

Qualification Profile
- Good knowledge of Linux and mySQL and related development tools
- Profound knowledge in PHP using the PHP programming framework Yii 2.0
- Team spirit, personal initiative, and self-organization
- Fluent spoken and written English

Interested?
We are flexible allowing you to combine studying with practical experience. Please send