13 Signaling Protocols for Multimedia Communication

Literature:

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

Ulrich Trick, Frank Weber: SIP, TCP/IP und
Telekommunikationsnetze, Oldenbourg, 4. Auflage 2009

Signaling

- *Signaling (Signalisierung, Zeichengabe)* originates from circuit-switched networks
- Signaling = Protocols of the Control Plane
  - User-to-Network Signaling: From end system to network interface
  - Network-to-Network Signaling: From one network node to another network node
  - End-to-End Signaling: From one end system to another end system
- Examples:
  - Call setup in ISDN
  - Call setup in ATM (Q.2931)
  - Resource reservation in IP networks (RSVP)

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 for Multimedia Conferences

- Session control:
  - Unit managing participants of a (conference) session
  - Management of involved connections
  - Monitoring of quality
- Signaling:
  - In particular call control:
    - How does a participant set up/join/tear down a session?
  - Negotiation of capabilities among clients
  - Adaptation to network traffic situation
  - Advanced features (like multiple calls, intelligent forwarding)

Network Architecture Option 1: Skype Based

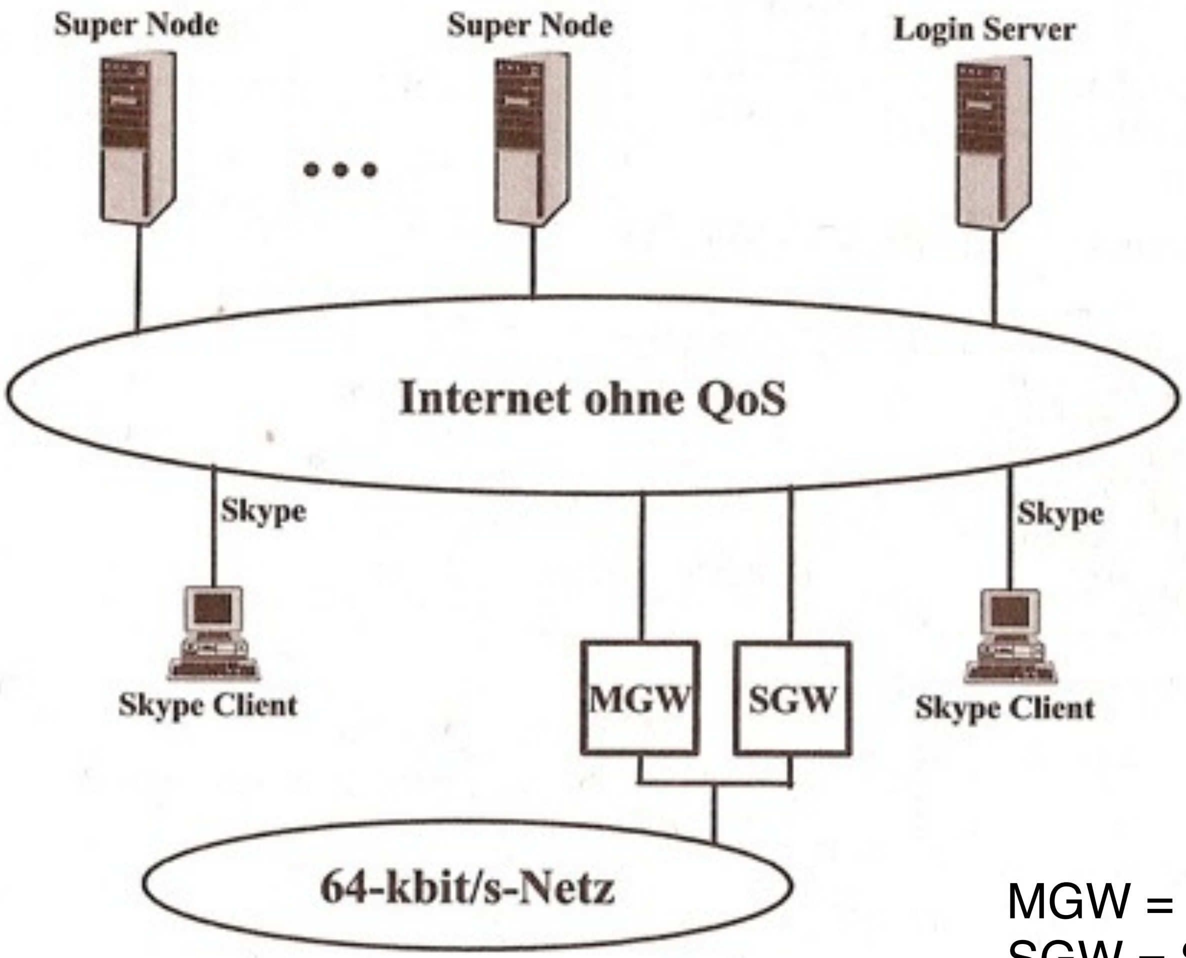


Bild 3.6: Skype für die Session-Steuerung

MGW = Media Gateway
SGW = Signaling Gateway

Trick/Weber

Skype Based Architecture

- Based on KaZaA peer-to-peer file sharing architecture
- Central *Login Server* for authentication
- Many *Super Nodes* form distributed database for user profiles
  - Powerful client computers with fixed address
- Steps in a Skype session:
  - User logs in (Login Server)
  - Client searches for Super Nodes and connects to a Super Node
  - Client gets address of communication partner from Super Node and establishes direct (peer-to-peer) communication link
  - Voice transmission: via UDP, adaptive between 24 and 128 kbit/s
    - » Predictive codecs: iSAC (LPC based), SILK (hybrid predictive/synth.)
  - Encryption of transmitted data
    - » Using AES 256 bit, key exchange through RSA
- Signaling and detailed architecture fully proprietary
- Alternative: New open standards based on WebRTC

Network Architecture Option 2: H.32X Based

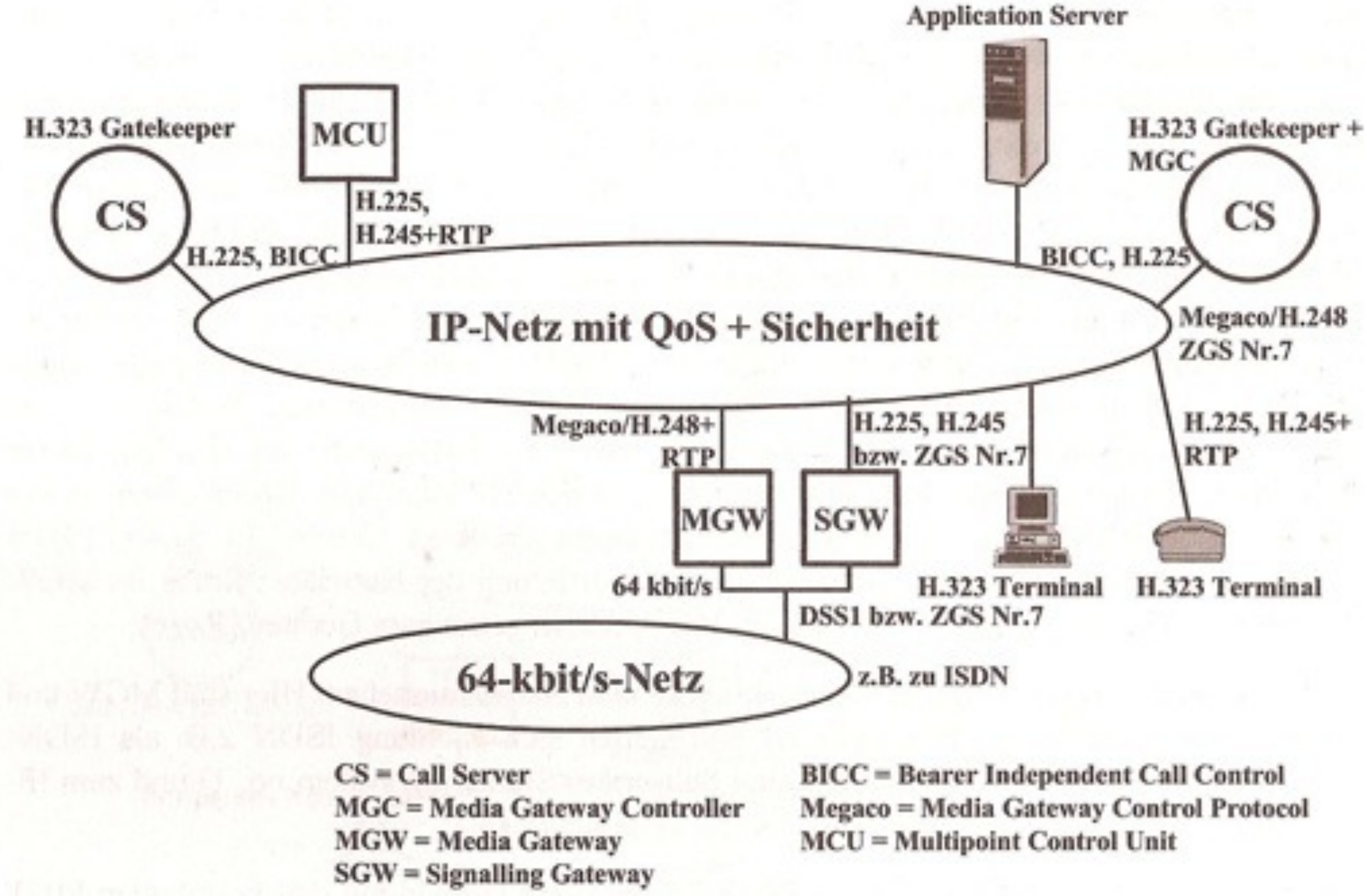


Bild 3.4: Protokolle und Netzarchitektur für Next Generation Networks mit H.323 für die Session-Steuerung

Trick/Weber

Network Architecture Option 3: SIP Based

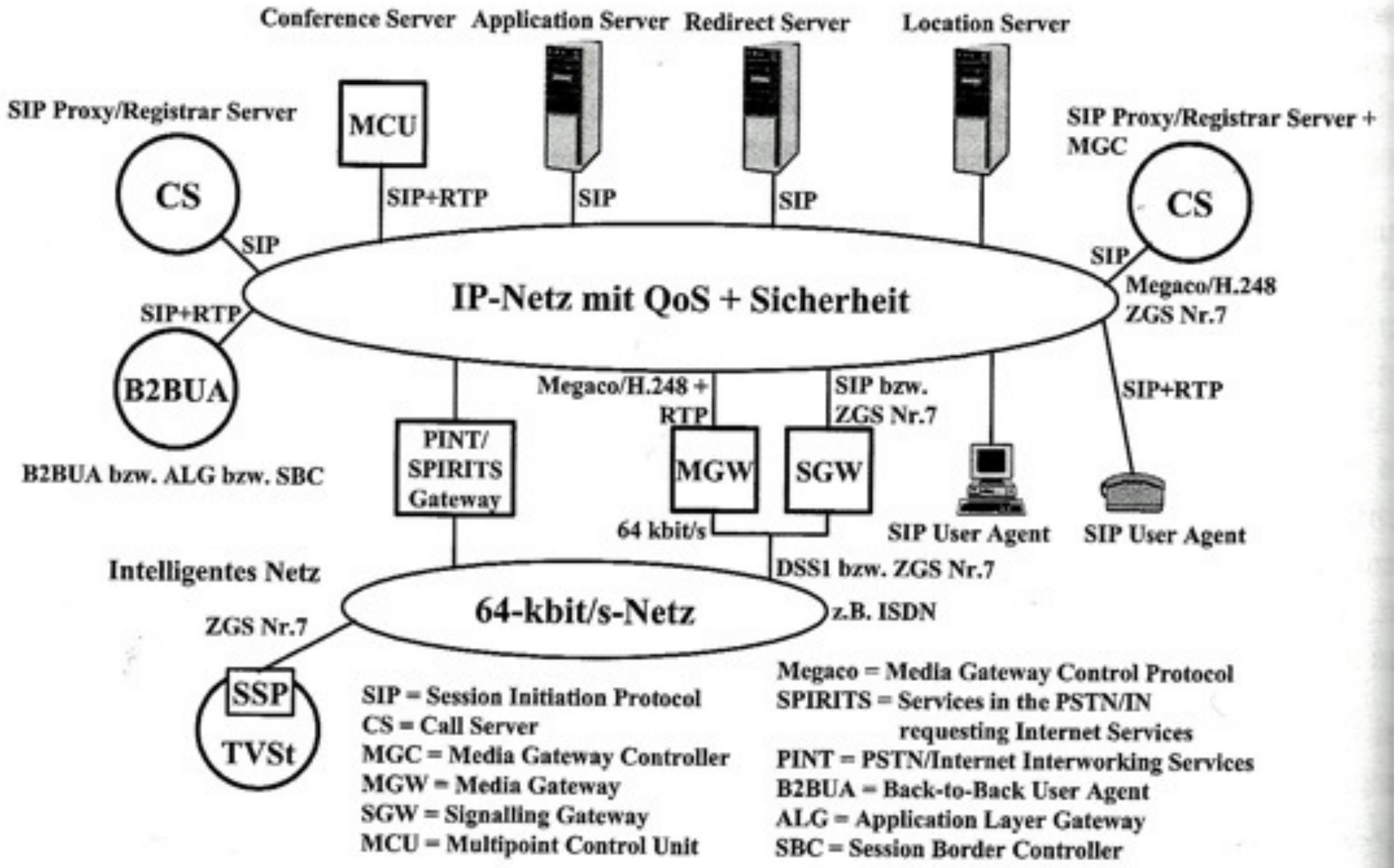


Bild 3.3: Protokolle und Netzarchitektur für Next Generation Networks mit SIP für die Session-Steuerung

Trick/Weber

13 Signaling Protocols for Multimedia Communication

Literature:

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

Ulrich Trick, Frank Weber: SIP, TCP/IP und Telekommunikationsnetze, Oldenbourg, 4. Auflage 2009

SIP - The Context

- SIP = *Session Initiation Protocol*,
  standardized by IETF *(Internet Engineering Task Force)*
  - Signaling protocol independent of underlying network technology
  - Text-based client/server protocol, 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)
- Developed based on proposals by Mark Handley and Henning Schulzrinne, 1999
- Related other protocols:
  - SDP = *Session Description Protocol*
  - SAP = *Session Announcement Protocol*
  - SCCP = *Simple Conference Control Protocol*
  - RTSP = *Real Time Streaming Protocol*
  - RTP = *Real Time Transport Protocol*

Main Features & Components of SIP

- SIP Proxy Servers for forwarding of control messages
  - Including “redirect” and “location” servers
- Support of user, terminal and service mobility
- Gateways to traditional networks (e.g. telephone networks)
  - Including services of the so-called “Intelligent Network” (IN), i.e. advanced network features
- Status observation for users and terminals (e.g. online/offline, busy/free)
- Service creation and execution tools
  - Call Processing Language CPL
  - XML-Scripts in SIP server
  - SIP-Java-Servlets

- In the following: Focus (first) on audio connections = “IP telephony”

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

- Text-based peer-to-peer protocol
- Modelled after HTTP
  - *Header* contains connection parameters and service information
  - *Body* contains 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)

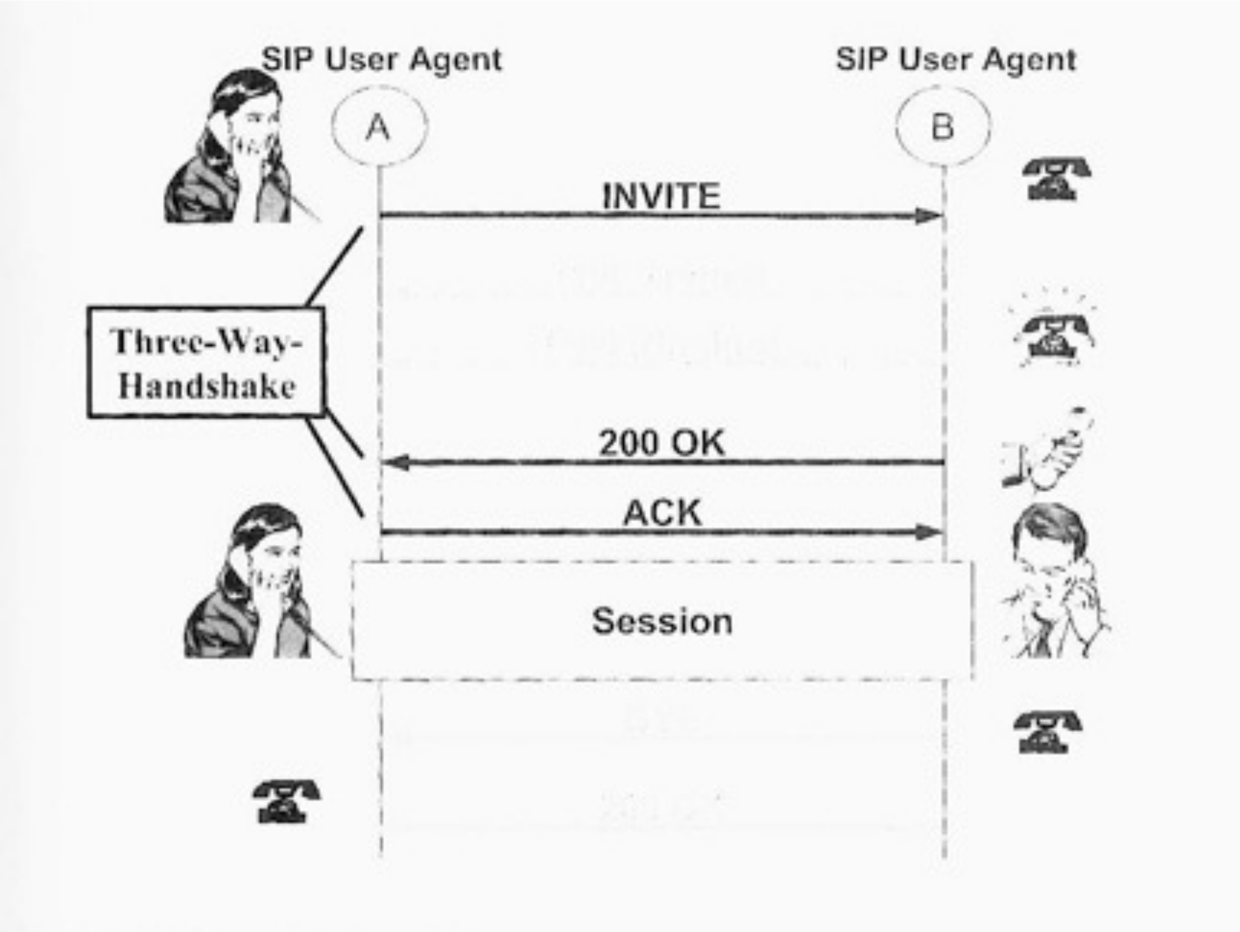


Bild 5.2: SIP Three Way Handshake

Trick/Weber

Example: SIP Message

```
INVITE sip:john@domain.com SIP/2.0
```
Start Line

```
VIA:SIP/2.0/UDP 169.130.12.5
Call-ID:187602141351@worchester.bell-telephone.com
From:<sip:a.g.bell@bell-telephone.com>
To:T.A.Watson<sip:watson@bell-telephone.com>
CSeq:1 INVITE
```
General Header

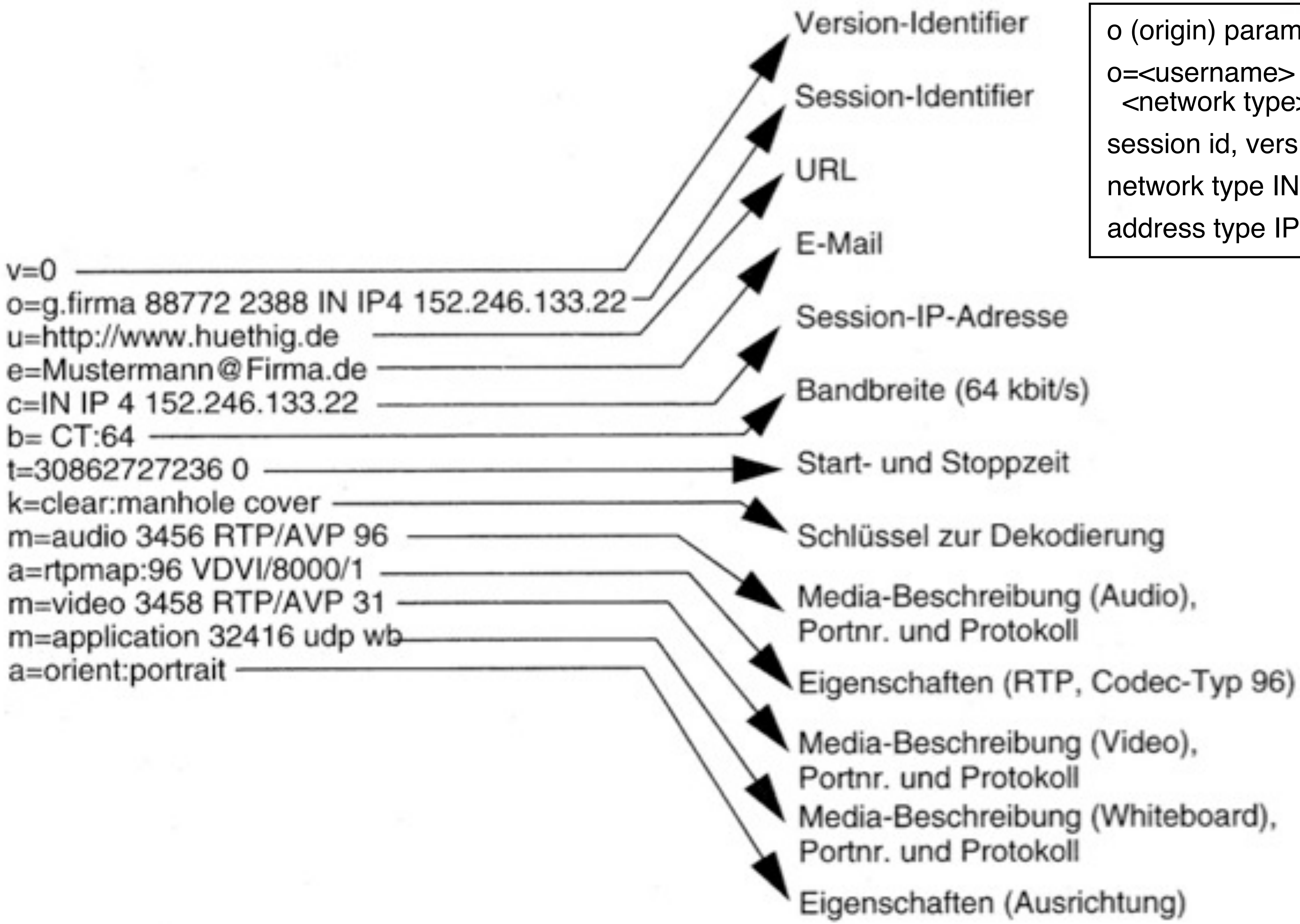
Sequence Number

```
Subject:Mr. Watson, come here
```
Request Header

```
Content-Type:application/sdp
Content-Length:885
```
Entity Header

```
v=0
o=bell 536557652353687637 IN IP4 128.3.4.5
c=IN IP4 135.180.144.94
m=audio 3456 RTP/AVP 0 3 4 5
```
Body: SDP Data

SDP Information



o (origin) parameter:

o=<username> <session id> <version> <network type> <address type> <address>

session id, version: NTP timestamp

network type IN = Internet

address type IP4 or IP6

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
    - » "96" 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
```

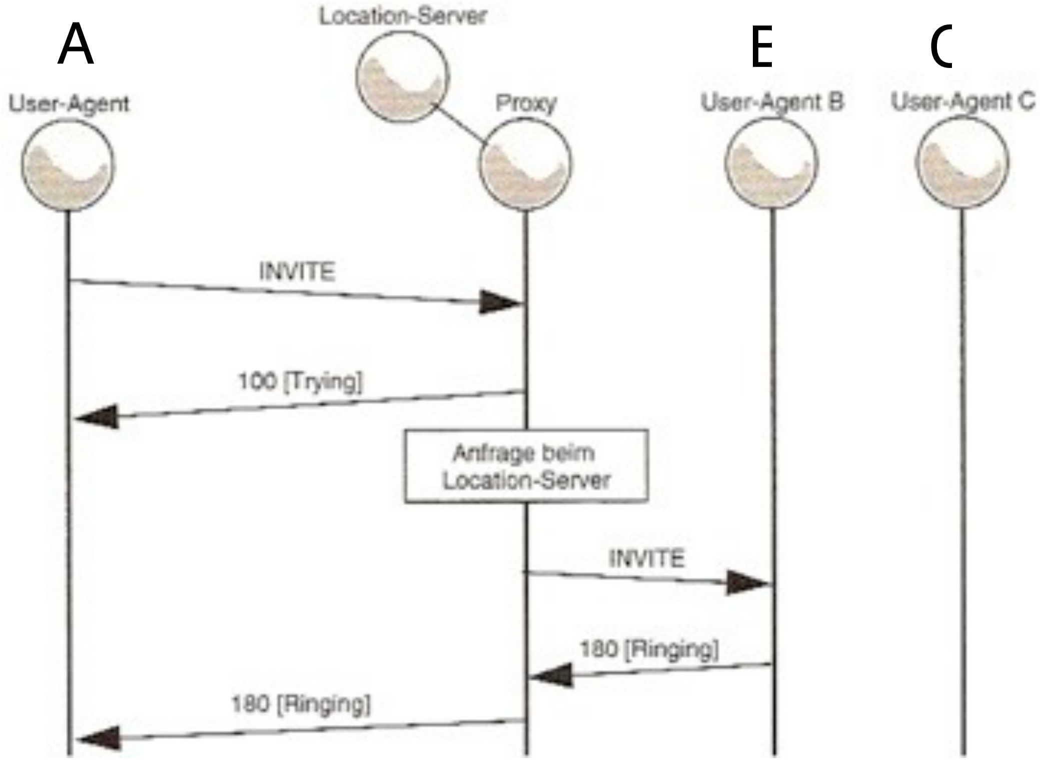
- Communication partner announces the codecs/formats which are locally supported
- Standardized list of RTP-Codecs in RTP/AVP standard, excerpt:

Payload type	Encoding name	Media type	Clock rate	Channels
0	PCMU (μ-law)	A	8000	1
1	reserved	A		
2	reserved	A		
3	GSM	A	8000	1
4	G723	A	8000	1
5	DVI4	A	8000	1
6	DVI4	A	16000	1
7	LPC	A	8000	1
8	PCMA (a-law)	A	8000	1

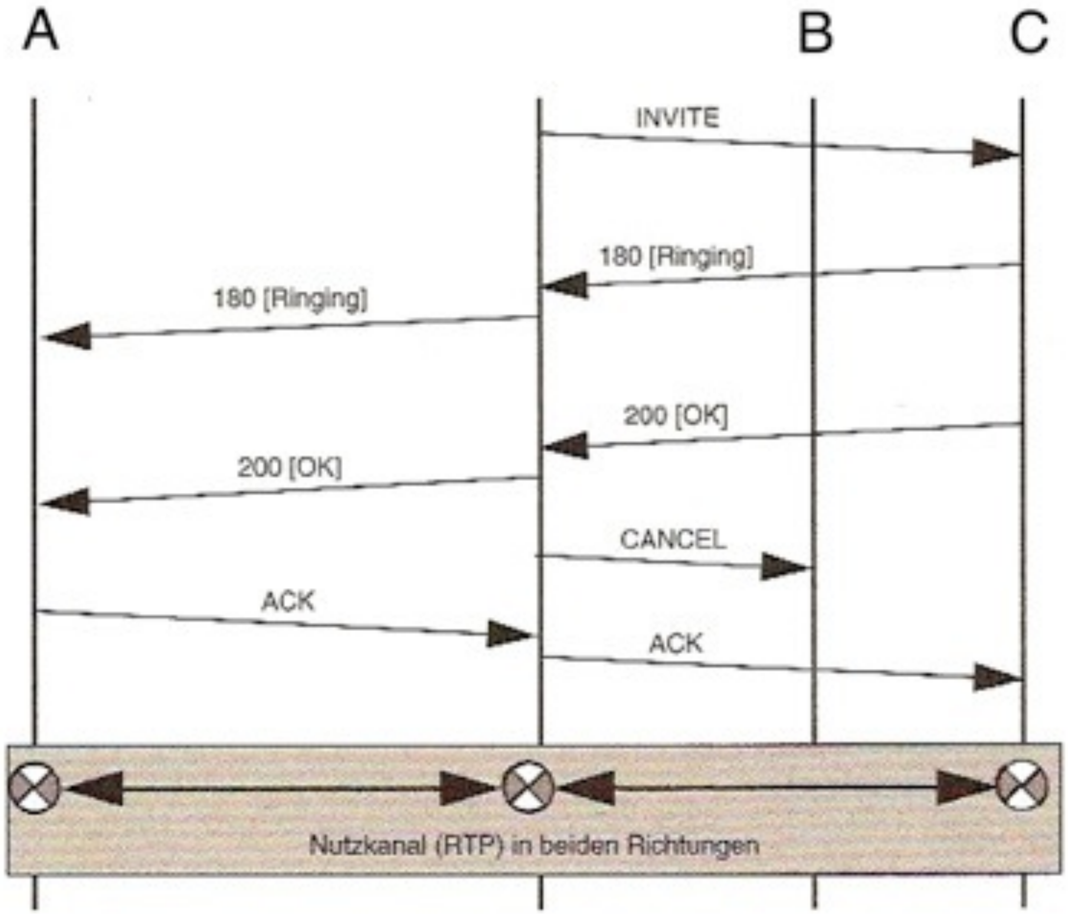
Codec Negotiation by Offer&Answer

- 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 RTP/AVP 0 8 3 4`
  - Answer:
    `m=audio 2468 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

Literature:

www.ietf.org/impp
www.xmpp.org

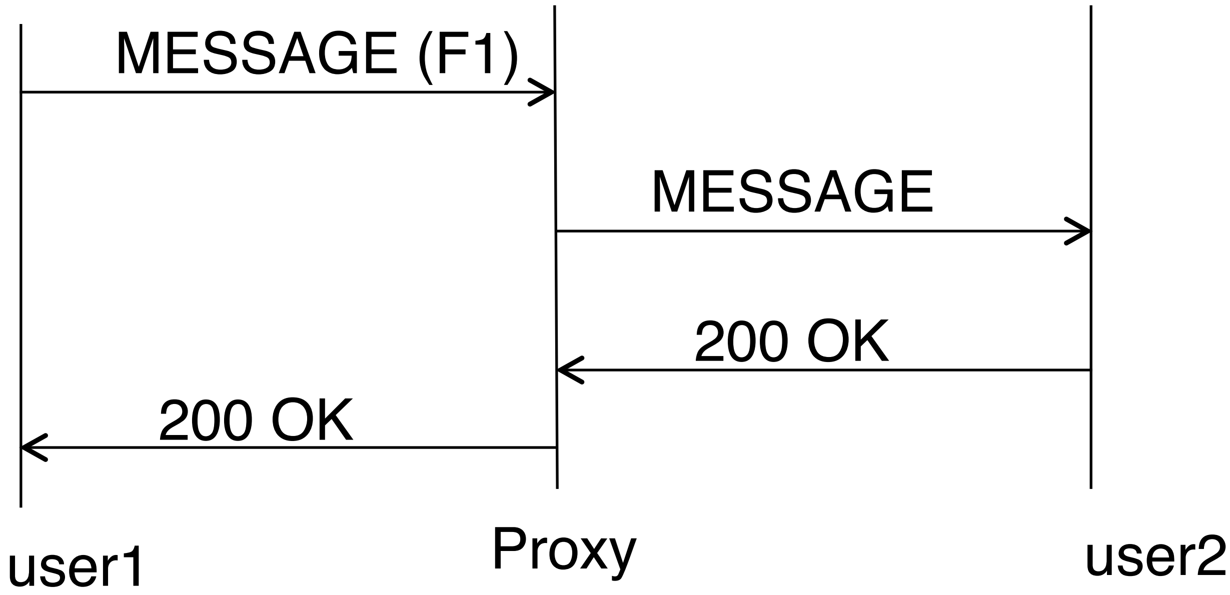
Instant Messaging (IM)

- Exchange of text information between clients in real-time
- Usually combined with *presence information*
- Traditionally computer-based, but may be used on other devices
- Modern clients often integrated with audio/video conferencing
- History:
  - 1970s: Terminal-based messaging (e.g. Unix “talk”)
  - Commercial GUI-based systems: ICQ (1996), AOL Instant Messenger (1997)
  - Many incompatible systems: Yahoo, MSN, Excite, ...
  - 2000: Open-source protocol “Jabber”, developed into XMPP
  - Current: Multi-protocol clients, e.g. Adium, Digsby, Pidgin, Trillian, iChat, ...
- Architecture:
  - Many clients, few servers
  - Device-based or network-based (server-based)
  - Centralized servers (e.g. ICQ) vs. decentralized servers (e.g. Jabber)

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
  – Currently no multimedia support, just text messages
- XMPP (Extensible Messaging and Presence Protocol)
  – Standardized form of XML-based streaming and presence protocols developed by the “Jabber” community (since 1999)
  – IETF standardization 2002–2004: RFCs 3920-23
  – Quite complete, covers e.g. authentication and encryption, multi-user chat, privacy blocking
  – Increasing support from commercial IM applications
    » e.g. Google Talk, Apple iChat, Facebook Chat XMPP Interface (2010)

SIMPLE Example (1): Message

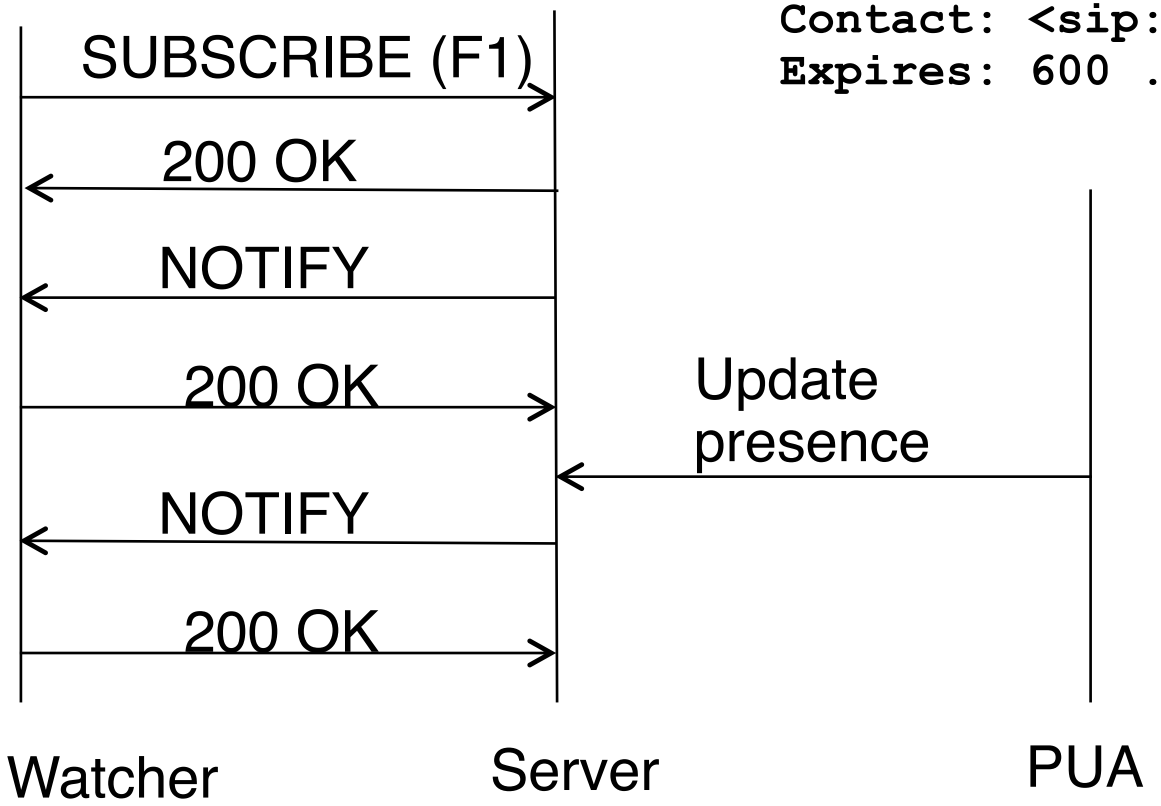


F1:

```
MESSAGE sip:user2@domain.com SIP/2.0
Via: SIP/2.0/TCP user1pc.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:user1@domain.com;tag=49583
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 18
Watson, come here.
```

SIMPLE Example (2): Presence

```
SUBSCRIBE sip:resource@example.com SIP/2.0
Via: SIP/2.0/TCP watcherhost.example.com;...
To: <sip:resource@example.com>
From: <sip:user@example.com>;tag=xfg9
...
Event: presence
Accept: application/pidf+xml
Contact: <sip:user@watcherhost.example.com>
Expires: 600 ...
```



PIDF (RFC 3863):
Presence Indication
Data Format

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

XMPP Example

C ⟷ S

```
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'>
C:     <body>Art thou not Romeo, and a Montague?</body>
C:   </message>
S:   <message from='romeo@example.net'
              to='juliet@example.com'
              xml:lang='en'>
S:     <body>Neither, fair saint, if either thee dislike.</body>
S:   </message>
C: </stream:stream>
S: </stream:stream>
```

Source: RFC 3920