Multimedia Networking
Communication Protocols

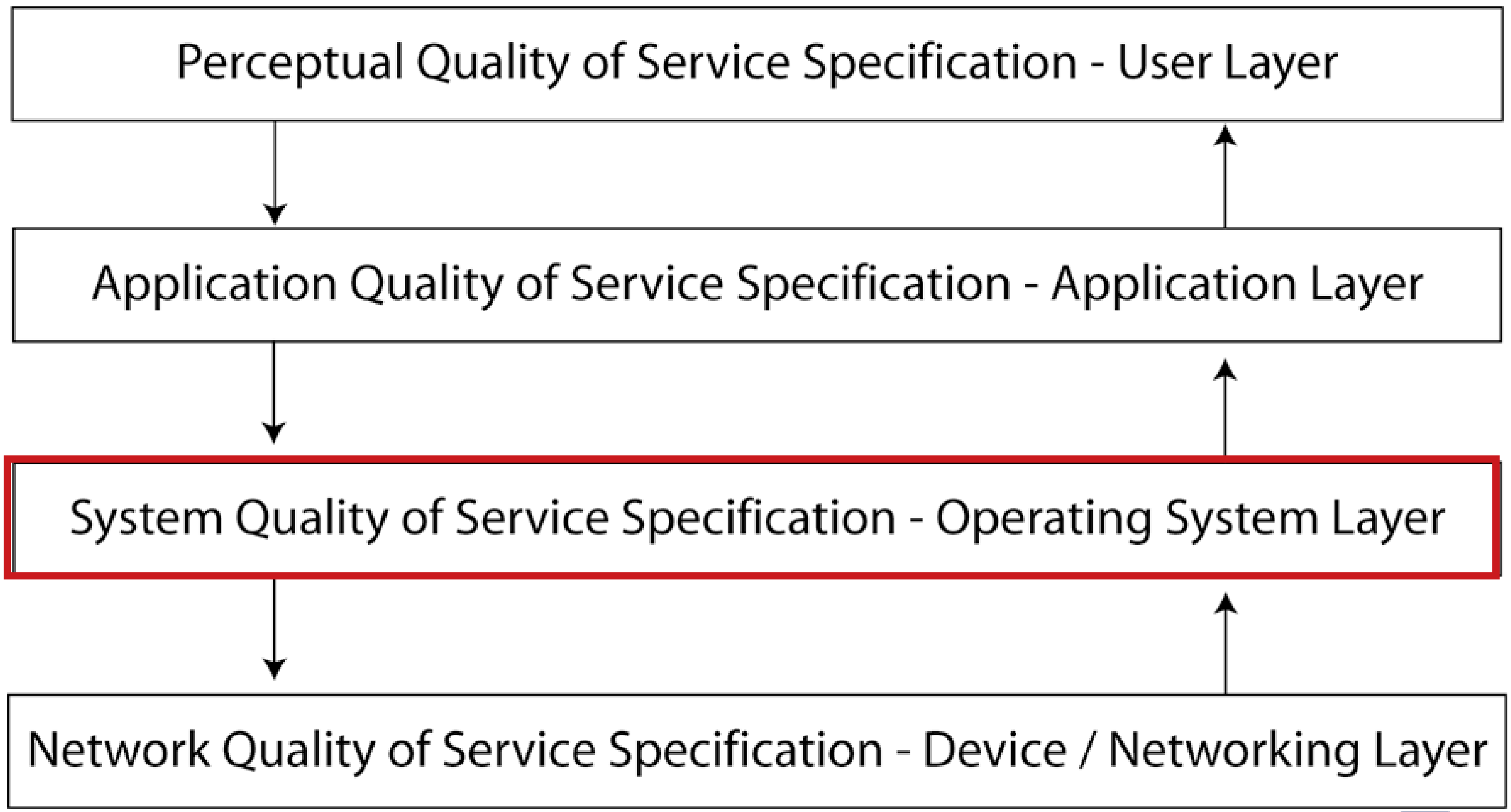
Thomas C. Schmidt
t.schmidt@ieee.org
HAW Hamburg

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Agenda

- Multimedia Communication Requirements
  - Signalling Demands
- Legacy VoIP/VCoIP: H.323
- The Internet Multimedia Protocol Suite
- Session Initiation Protocol

QoS – Layered Model



Multimedia Communication Across IP Networks

Information about Media Transport need to be shared between partners and sometimes with the network.

- Provide application-specific framing.
- Communicate media-specific intelligence & metadata.
- Place media signalling information in network transport.

Signalling Demands

Media Types can be announced by MIME, but in Real-Time Communication demands remain:

- Session Information - Application based connection handling
- Session Negotiation - Dialogs need media agreement
- Timer Information - Partners need a clock tick
- Coding Details - Time/context dependent metadata
- Time-dependent Stati - Communication may adapt to user or network needs
- Address Information - Matching users to devices
- Session Announcement - Advertising sessions

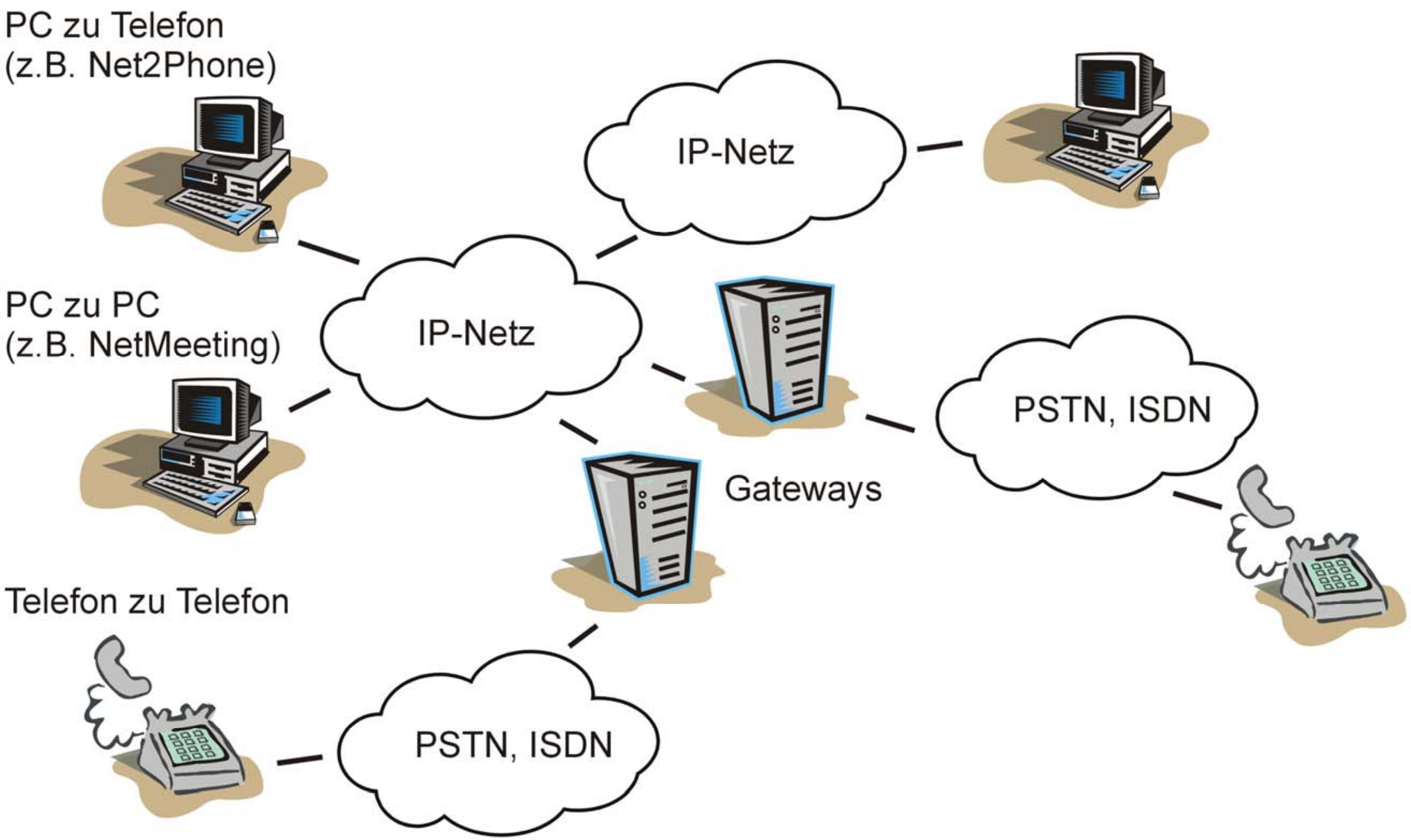
Agenda

- Multimedia Communication Requirements
- Legacy VoIP/VCoIP: H.323
  - Basic Components
  - Signalling Protocols
  - Common Scenarios
- The Internet Protocol Suite
- Session Initiation Protocol

H.323
Voice & Video over IP

- ITU-T Recommendation for Voice/Video conferencing over IP
  - Currently H.323 Version 4 (November 2000)
- Transfers digital telephony onto IP Layer
- Main functionalities
  - Bearer-Control-Function
  - Registration, Admission, Status (RAS)
  - Call Signalling
  - Gateway Service to PSTN
- Widely implemented architecture, though legacy protocols in use

H.323 Interconnects



H.323 System Components

- Terminal

  H.323 client, either IP-phone, VCoIP station or software

- Gatekeeper

  Directory Service for user-address translation, signalling service, supplementary services, bandwidths control

- Gateway

  Gateway services between IP and PSTNs

- Multipoint Conference Unit

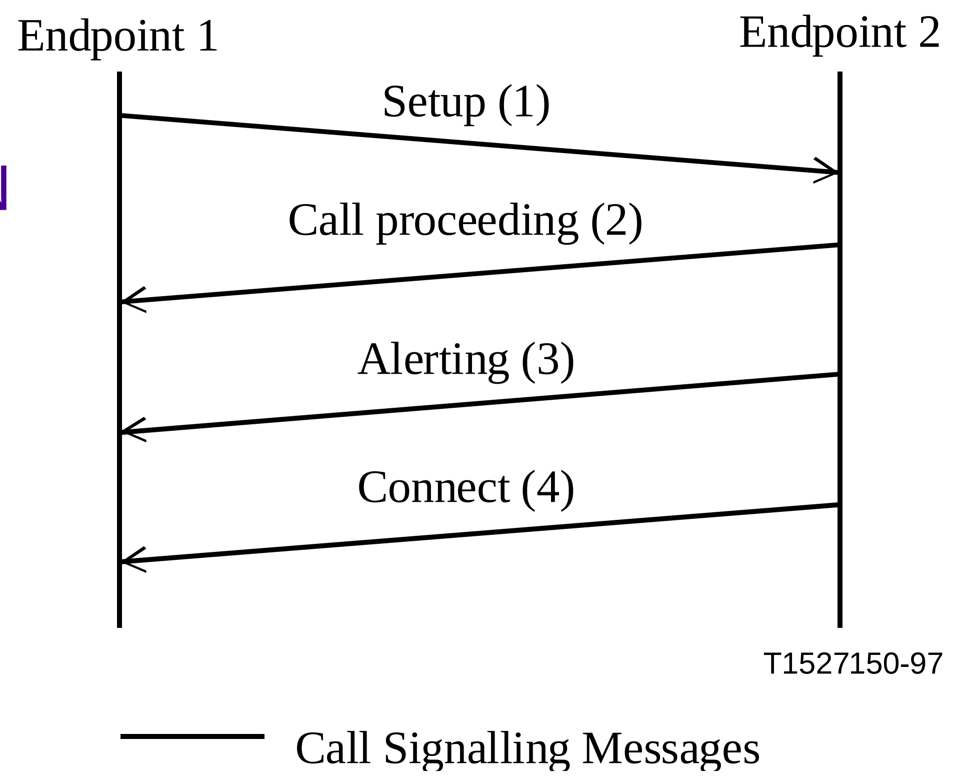
  Reflector server for group communication

<table>
<tr><td colspan="6">H.323-Standard</td><td colspan="2">ISO-OSI-Reference</td></tr>
<tr><td>Video Codecs</td><td>Audio Codecs</td><td colspan="4">Management/ Control</td><td rowspan="3">7 -
6 -
5 -</td><td rowspan="3">Application</td></tr>
<tr><td>H.26x</td><td>G.7xx
GSM
6.10</td><td rowspan="2">RTCP</td><td rowspan="2">RAS</td><td rowspan="2">Signalling
H.225.0</td><td rowspan="2">H.245</td></tr>
<tr><td colspan="2">RTP</td></tr>
<tr><td colspan="4">UDP</td><td colspan="2">TCP</td><td>4</td><td rowspan="4">Transport</td></tr>
<tr><td colspan="6">IP</td><td>3</td></tr>
<tr><td colspan="6">LLC / MAC – IEEE-802.x</td><td>2</td></tr>
<tr><td colspan="6">Fiber, Twisted Pair, ...</td><td>1</td></tr>
</table>

H.323 – Umbrella Standard

H.225 Signalling

- o IP-Correspondent of ISDN Signalling (Q.931)
- o Simulates a circuit switched network by managing bidirectional logical channels



H.245 Conference Control

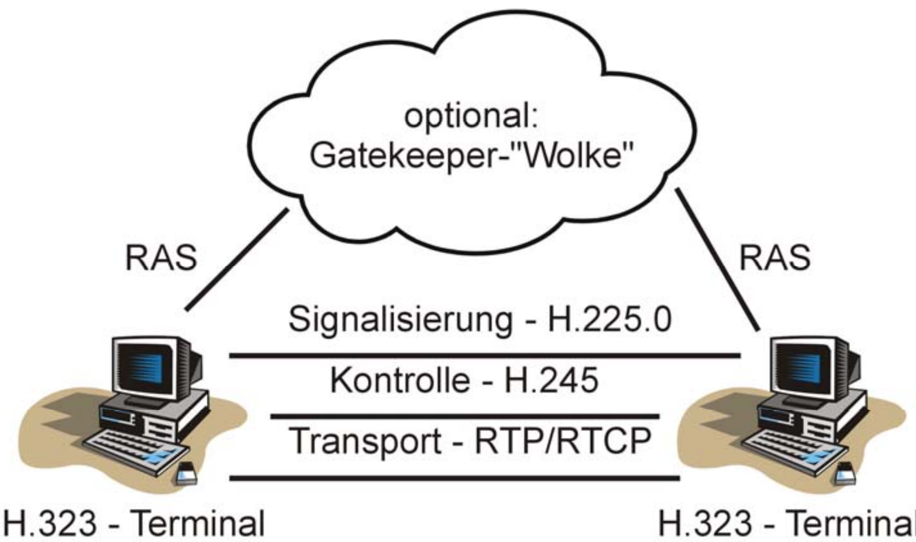
- o Legacy protocol to coordinate conferencing parties from different networks (IP, PSTN, ATM, ...)
- o Negotiates possible modes for media exchange (codecs)
- o Configures media streams (including transport addresses)
- o May carry user input from DTMF ...
- o Defines multipoint conferences
- o Initiates privacy mechanisms (H.235)
- o Provides channel maintenance messages

H.323 Signalling:

Direct-routed call

1. Caller – Gatekeeper (RAS)
   - Admission Request (ARQ)
   - Admission Confirm/Reject (ACF/ARJ)

   ⇒ destCallSignalAddress
2. Caller – Callee (H.225.0)
   - setup
3. Gatekeeper – Callee (RAS)
   - ARQ – ACF/ARJ
4. Callee – Caller (H.225.0)
   - connect
5. Caller – Callee (H.245)
   - Control Channel Established



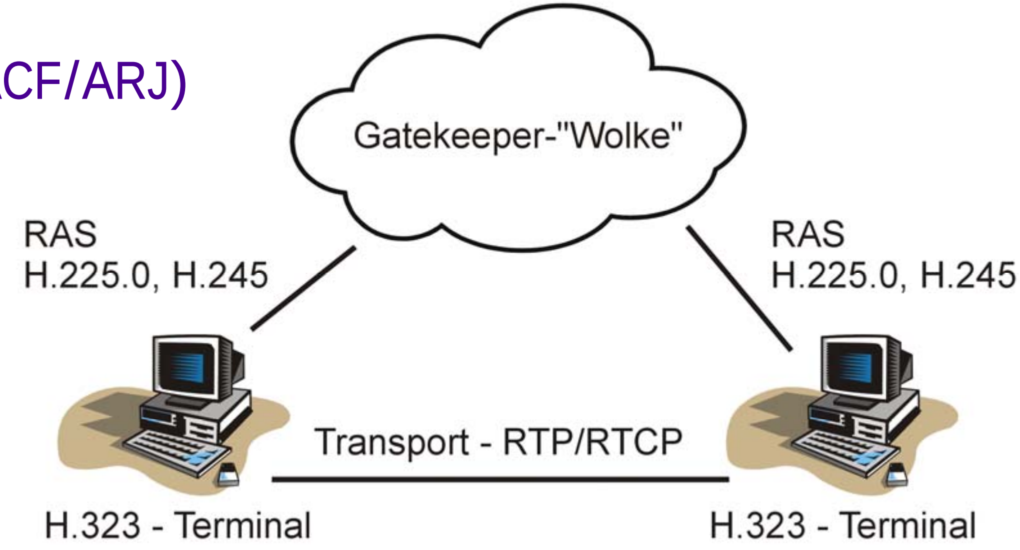
RAS signalling remains optional:

Direct routing works without Gatekeeper

H.323 Signalling:

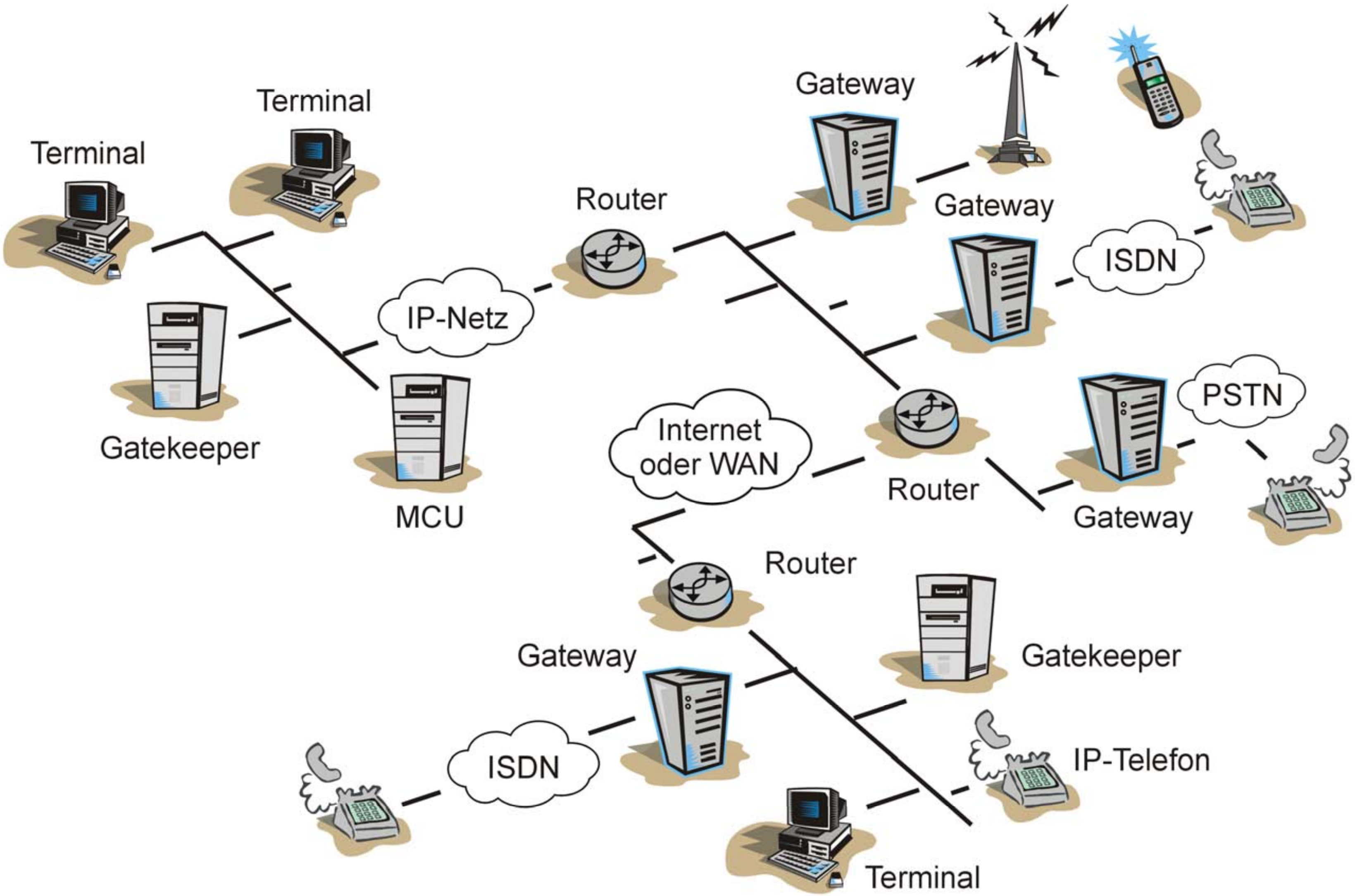
Gatekeeper-routed call

1. Caller – Gatekeeper
   - Admission Request (ARQ)
   - Admission Confirm/Reject (ACF/ARJ)
   - setup
2. Gatekeeper – Callee
   - setup
   - ARQ - ACF/ARJ
   - connect
3. Gatekeeper – Caller
   - connect
4. Caller – Gatekeeper - Callee
   - Control Channel Established (H.245)



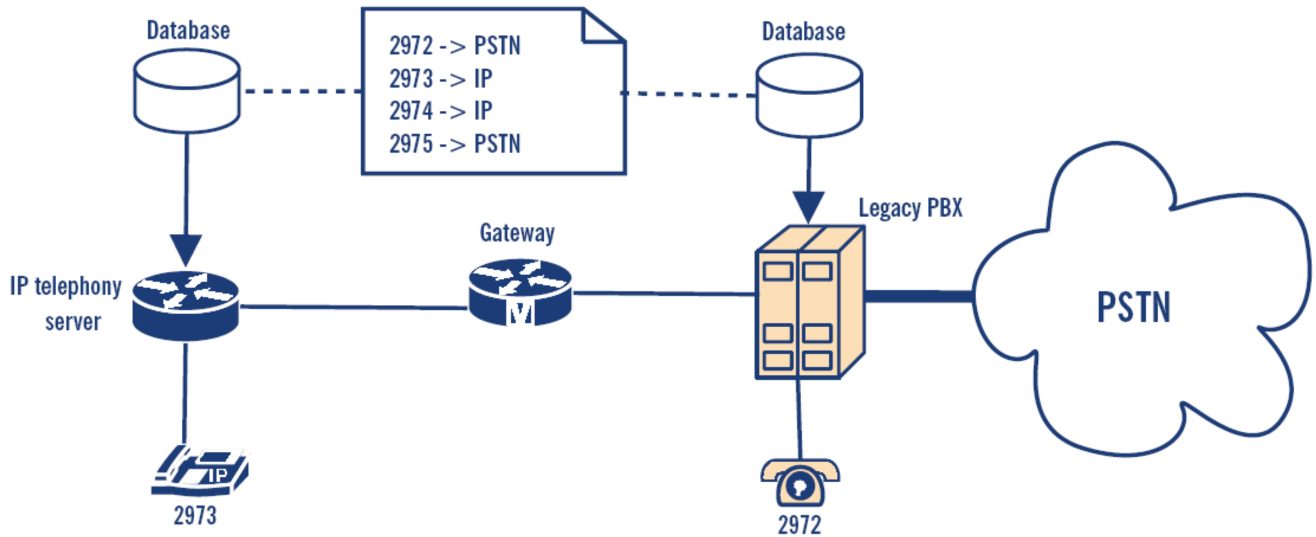
Call control requires operational Gatekeeper

H.323 Scenario



H.323 – Basic Configuration

- Setting up Devices, a Dial-Plan + Routing at Gatekeeper/PBX

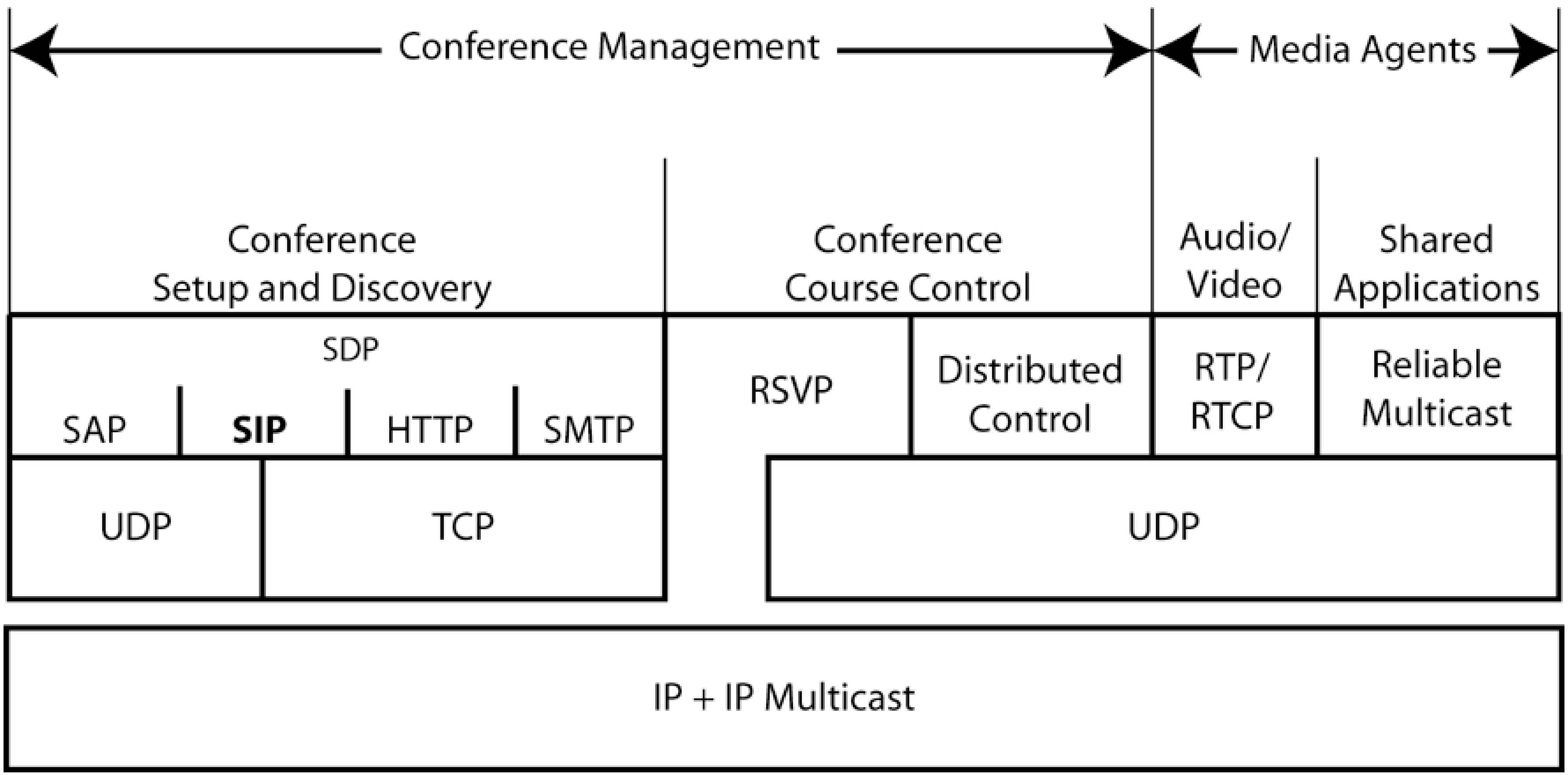


- Configuring Interfaces + Services at Gateway
- Setting up Security (H.235 – rarely implemented)

Agenda

- Multimedia Communication Requirements
- Legacy VoIP/VCoIP: H.323
- The Internet Multimedia Protocol Suite
  - Real-Time Media Transport
  - Session Description
  - Session Negotiation and Announcement
- Session Initiation Protocol

Multimedia Communication: The Internet Protocol Suite



Real-time Transport Protocol

RTP/RTCP (V2, RFC 3550, Schulzrinne et al 2003)

- End-to-end transmission of real-time data
- RTP identifies and synchronises data streams
- RTCP transmits controls to allow for adaptation

Sessions

- Identify parties, sort and order packets

Timestamps

- Decorate packets with temporal alignment

Media-specific Signalling

- Extendable profiles according to media requirements

A Typical Application Scenario

Voice or Video Conference

- Two party (IP unicast) or group (IP multicast)
- Transport of media data: RTP packets within UDP
- RTP provides timing, ordering and identification
- Media specific encodings carried within RTP: e.g. frame type, layers, adaptive schemes
- Audio and video as two separate RTP streams
- Resynchronisation of streams (mixing) and transcoding (translation)
- Privacy via SRTP profile
- RTCP reports on receivers and reception quality

RTP Entities

- Transport Address

  Combination of network (IP) address and port as defining an endpoint

- RTP media type

  Any collection of payload types within a single RTP session

- RTP session

  One communication between a pair of transport addresses

- RTP multimedia session

  A set of RTP sessions among a common group of participants

- Mixer

  An intermediate system receiving RTP packets while changing formats or packet combinations

RTP Entities (2)

- Synchronisation source (SSRC)

  Source of a synchronised RTP stream, identified by the SSRC id

- Contributing source (CSRC)

  Source of a synchronised RTP stream contributing to a combined stream produced by a mixer, identified by the CSRC id

- Translator

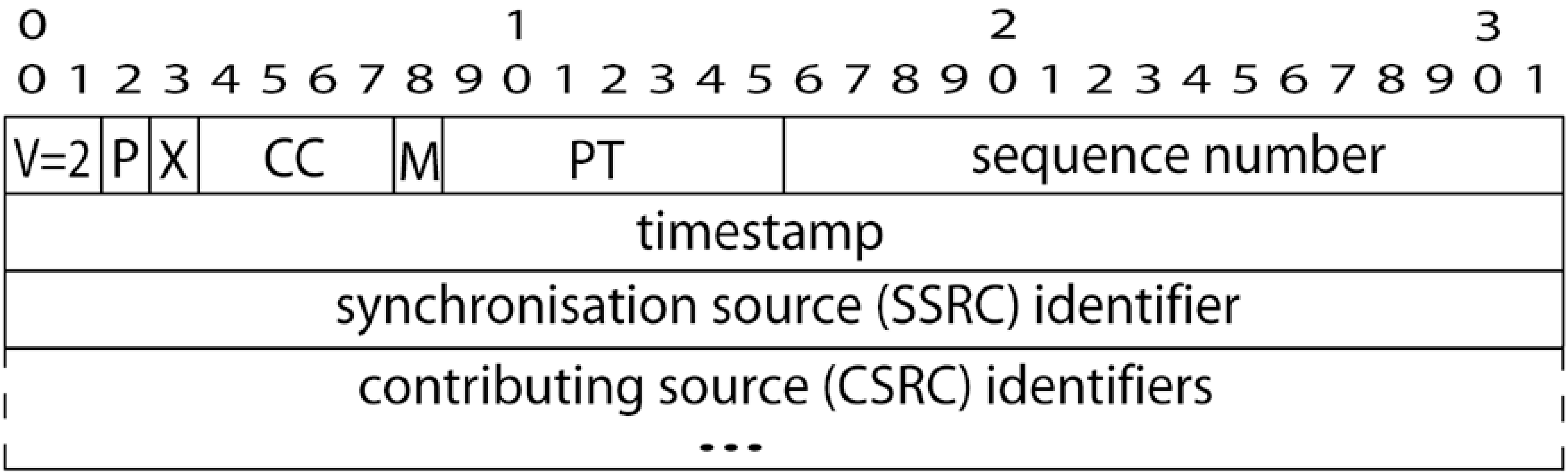
  An intermediate system forwarding RTP packets without changing SSRC, but possibly modifying payloads

- Monitor

  An application receiving RTCP packets for diagnostics

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

RTP Fixed Base Header



Version(V): 2 bit
Padding(P): 1 bit
Extension(X): 1 bit
CSRC count (CC): 4 bit
Marker (M): 1 bit

Payload Type(PT): 7 bit
Sequence Number: 16 bit
Timestamp: 32 bit
SSRC: 32 bit
CSRC: 0 to 15 items, 32 bits each

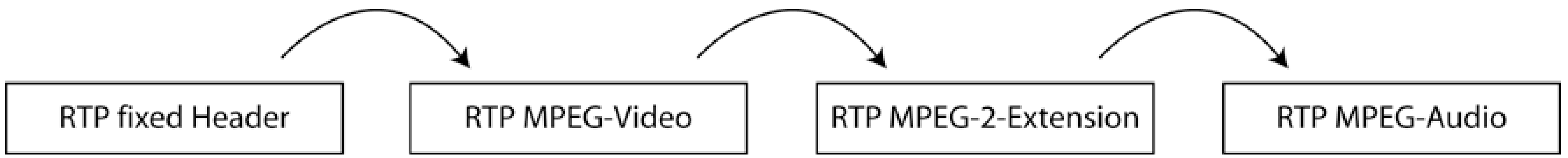
RTP & Media Encoding

RTP is intentionally left open to further media specifications and data interpretation within Profiles:

- o Payload Type – Identifies format and interpretation of the RTP payload (Audio/Video: RFC 3551)
- o Marker – Interpretation of the Marker is defined by a profile, e.g. marking frame boundaries
- o Extension Headers – May be defined in Profiles to carry additional, specific information

RTP Profiles: Header Chain

RTP allows for the encoding of media-specific information by possible (a chain of) Extension Headers.



- The extension bit indicates a following RTP header.
- The payload type indicates the profile of extension header type *and* of the payload data.

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

RTP MPEG Extension Header

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

MBZ	T	TR	AN	N	S	B	E	P	FBV	BFC	FFV	FFC

MBZ: For future use

Type (T): MPEG-2 set to 1

Transport Reference (TR): Temporal Reference of current picture within GOP(0-1023)

ActiveN (AN): Set 1, if N-Bit is used to signal changes in picture header

N: New-Picture-Header

S: Sequence-Header-Present

B: Beginning-of-Slice

E: End-of-Slice

P: Picture Type

FBV:
BFC:
FFV:
FFC:
MPEG-2-Vector-Identifier

nces

Real-time Transport Control Protocol

- RTCP provides feedback to the all members of the RTP session by a periodic transmission of control packets using the same distribution as data (e.g., multicast).
- RTCP feedback reports on
  - reception statistics on quality, i.e., loss, delay, jitter
  - faults to diagnose network problems
  - distribution properties, i.e., receivers of the session
- RTCP facilitates flow control & adaptive coding, but also multicast session surveillance
- RTCP reports adapt to network capacities and session members

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

RTCP Packet Types

Sender Report: transmit and receive statistics from active senders

- Delay, Jitter, Packet Loss, NTP timestamp, ...

Receiver Report: transmit and receive statistics from passive receivers

- Delay, Jitter, Packet Loss, ...

Source Description Items:

- Cname, Name, Email, Phone, Location, Tool, Note, ...

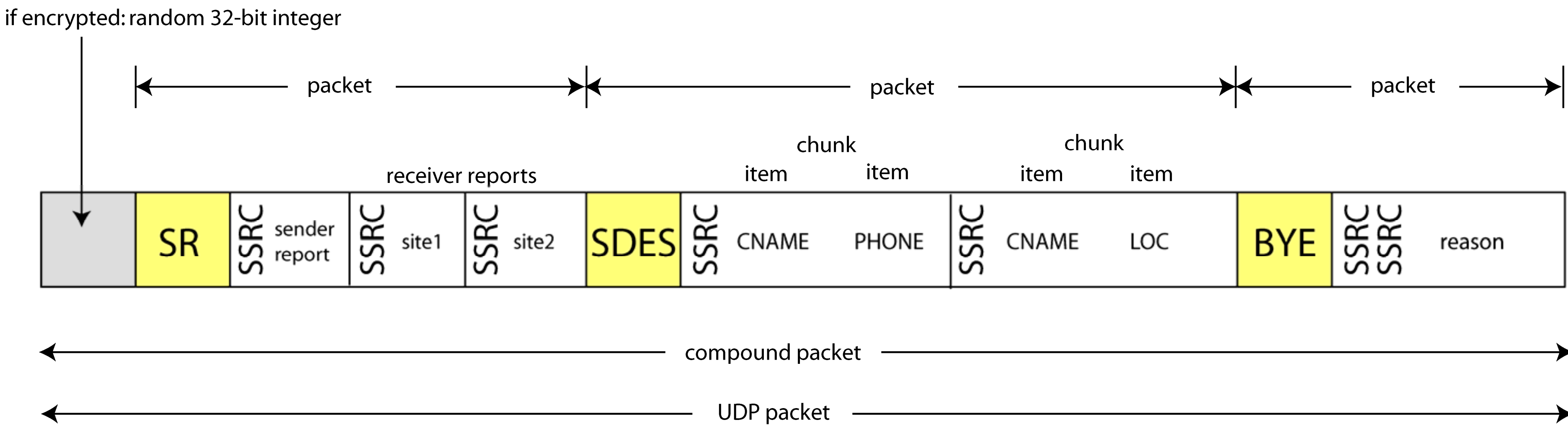
Bye: Leave Session

Application Specific Functions

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

RTCP Compound Packaging

For efficiency reasons RTCP reports can be concatenated to form a compound packet.



RTP Programming (C++)

Choose/bind RTP stack (no standardized API)

- Example: JRTPLIB – http://research.edm.uhasselt.be/~jori/page/index.php?n=CS.Jrtplib

Create session: (specify port)

```
RTPSession sess; status=sess.Create(5000);
```

Send RTP Data: (specify address, payloadtype, mark, timestamp increment)

```
sess.AddDestination(addr,5000);
sess.SendPacket("1234567890",10,4,false,13);
```

Receive RTP Data:

```
if (sess.GotoFirstSourceWithData()) {
    do  {
          RTPPacket *pack;
          pack = sess.GetNextPacket();
          // process packet
          delete pack;
        } while (sess.GotoNextSourceWithData()); }
```

RTP Programming (Java)

(One) RTP stack is part of the Java Media Framework 2 (http://java.sun.com/products/java-media/jmf)

JMF RTP API is built of the following components:

Session Managers: Maintains session participants, streams & statistics

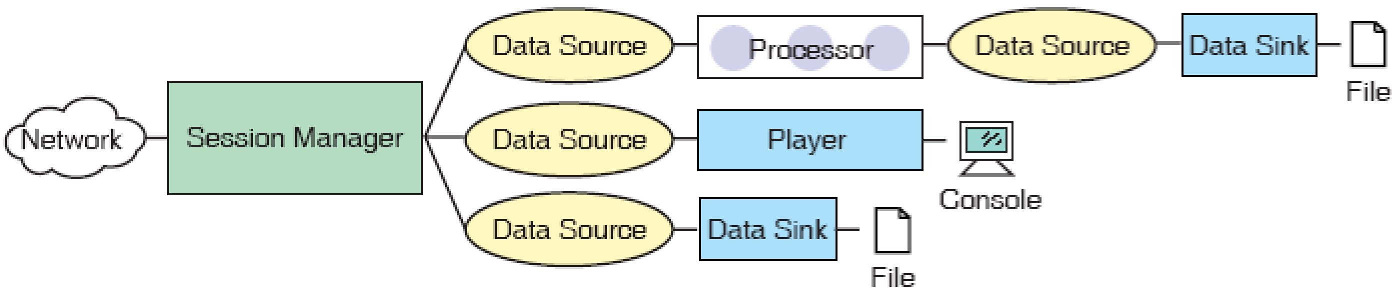
RTP Events/Listeners: Report on sessions, send/receive streams & remote participants

RTP Data: Predefined audio & video formats (extensible), transport protocol independent data handlers with input and output data streams

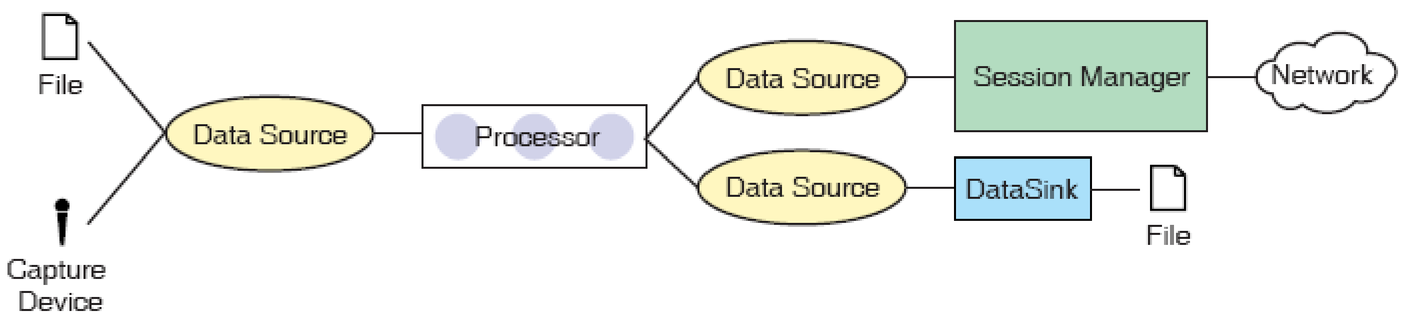
RTP Controls: Formats, sessions, buffers, statistics ...

RTP Programming (Java)

RTP Reception



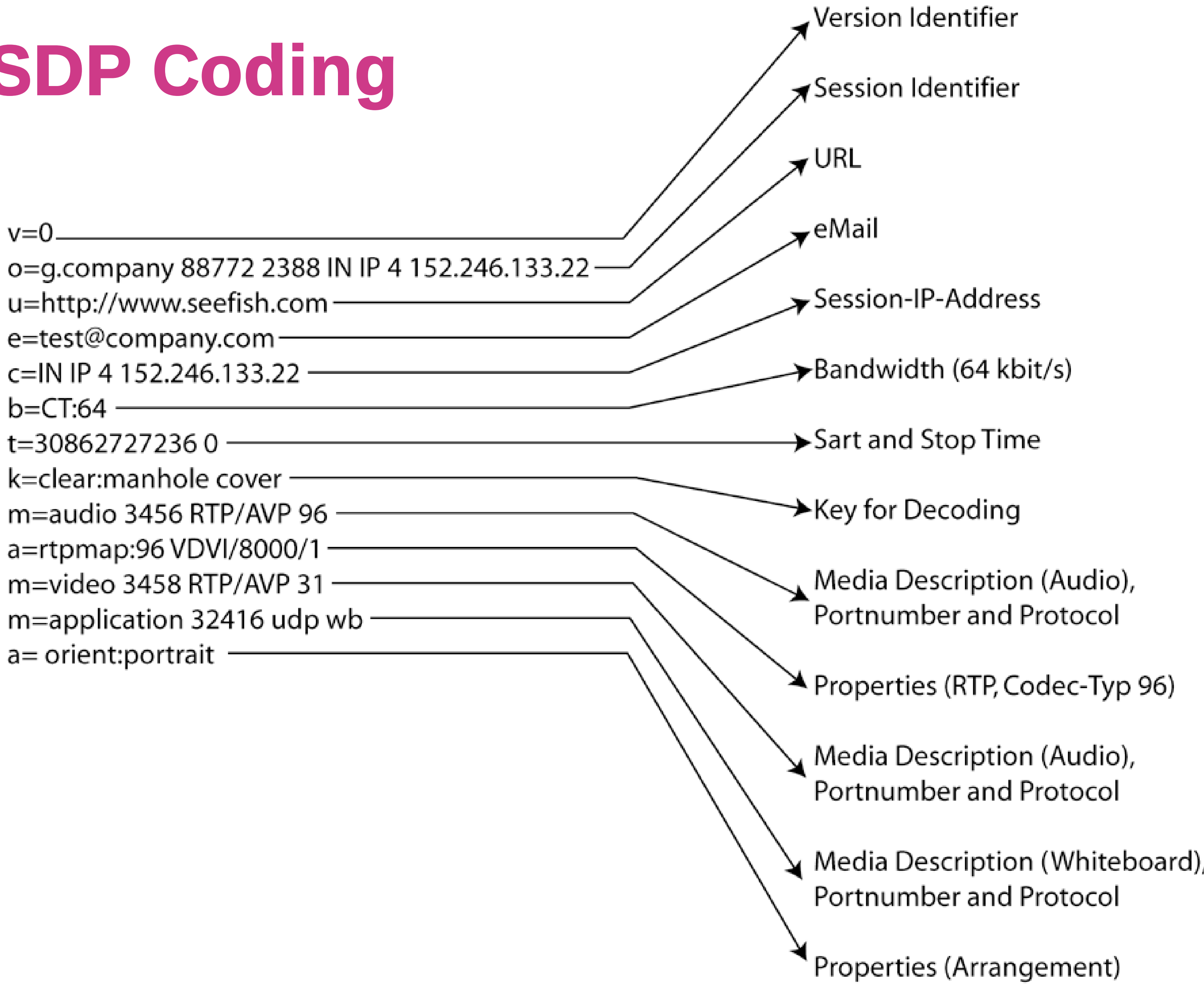
RTP Transmission



SDP
Session Description Protocol

- o IETF RFC 4556 (Handley et. al., MMusic)
- o General description of multimedia sessions:
  - Media details
  - Transport addresses & properties
  - User / session metadata
- o Focuses the purposes
  - Session announcement (e.g. via SAP)
  - Session invitation
  - Real-time streaming
  - Within MIME, e.g., in emails or http
  - SDP is only a format, independent of its actual transport

SDP Coding


v=0
o=g.company 88772 2388 IN IP 4 152.246.133.22
u=http://www.seefish.com
e=test@company.com
c=IN IP 4 152.246.133.22
b=CT:64
t=30862727236 0
k=clear:manhole cover
m=audio 3456 RTP/AVP 96
a=rtpmap:96 VDVI/8000/1
m=video 3458 RTP/AVP 31
m=application 32416 udp wb
a= orient:portrait
Version Identifier
Session Identifier
URL
eMail
Session-IP-Address
Bandwidth (64 kbit/s)
Sart and Stop Time
Key for Decoding
Media Description (Audio), Portnumber and Protocol
Properties (RTP, Codec-Typ 96)
Media Description (Audio), Portnumber and Protocol
Media Description (Whiteboard), Portnumber and Protocol
Properties (Arrangement)

SDP Parameters

Parameter	m/o	Name	Meaning
a	o	Attributes	Additional properties (SDP-extension)
b	o	Bandwidth	Necessary bandwidth
c	o	Connection Information	More information on media stream
e	o	Email Address	Email address of the „owner“
i	o	Session Information	Additional information in text format
k	o	Encryption Key	Security key for media streams
m	m	Media	Name and address of the media stream
o	m	Owner	Initiator (owner) of a session
p	o	Phone Number	Telephone number of the „owner“
r	o	Repeat	Repetition
s	m	Session Name	Session name
t	m	Time	Session duration
u	o	URI	Identifier of session description
v	m	Version	Version of the used protocol
z	o	Time Zone Adjustment	Time zone adjustment

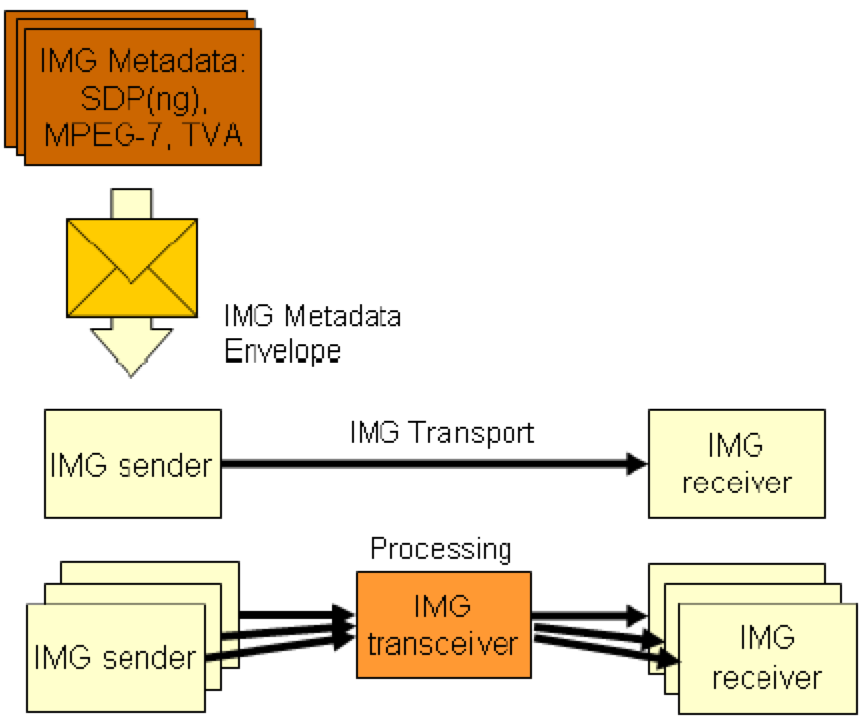
Session Announcement

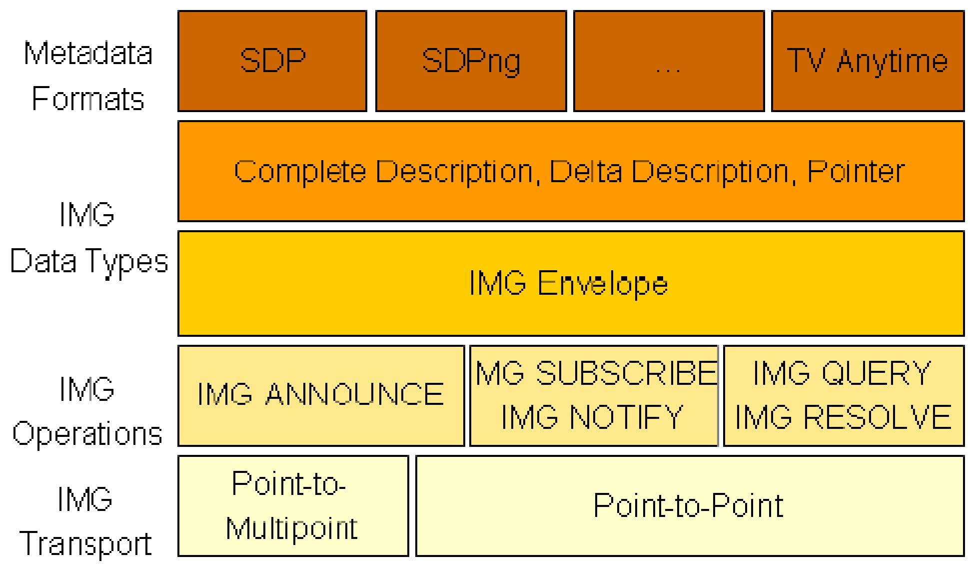
- o Simple Session Announcement via SAP
  - IETF experimental RFC 2974 (v2)
  - Periodic multicast of SDP data + optional authentication

- o Internet Media Guide Framework
  - General content description scheme derived from Electronic Program Guides (digital TV broadcasting)
  - Current standardisation effort in IETF - MMUSIC
  - Goal1: arbitrary content meta data support
  - Goal2: interoperation of any suitable distribution mechanism (push/pull unicast, multicast, ...)

Internet Media Guides

- Abstract meta-data types: Complete, Delta, Pointer (URI to meta data)
- Packaging in flexible envelopes
- Additional distribution “Transceiver” for proxying, combining, filtering, personalisation ...





SDP Offer / Answer Model (RFC 3264)

Objective:

Provide a mechanism by which two parties arrive at a common view of a multimedia session using SDP.

Offer:

Send SDP message with 0 to n media streams `m=“”´, which the offerer is willing to send or receive (including transport binding).

Answer:

Reply with a counter matching SDP message, containing all offered media streams, correspondently marked as ‘sendrecv’/ ‘send/recvonly’ or ‘inactive’.

Multicast:

Provides a single view of a unidirectional stream (direct matching).

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Agenda

- Multimedia Communication Requirements
- Legacy VoIP/VCoIP: H.323
- The Internet Multimedia Protocol Suite
- Session Initiation Protocol
  - SIP Architecture & Components
  - SIP Messages
  - SIP Extensions: Events & Presence
  - SIP Conferencing
  - Further Functions

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

SIP - Session Initiation Protocol

- IETF RFC 3261+ (Schulzrinne et al 2002)
- Signalling control protocol for multimedia sessions
- Main functionalities support
  - Call setup: ringing & establishment
  - Call handling: sustaining, transferring & termination
  - User location: discovery of user presence
  - User availability: discovery of user's call willingness
  - User capabilities: determination of media parameters for use
- Increasing number of implementations for VoIP, conferencing, presence and messaging services

SIP Protocol

- End-to-end application protocol transported via UDP or TCP
- Designed to establish, modify and terminate stateful multimedia communication (sessions/conferences/instant messaging ...)
- Signalling component, not an architecture like H.323, operates in combination with
  - RTP/RTCP for media transport
  - SDP for session description
  - SAP for session announcement
  - Gateway Control Protocol for PSTN gateway control
- Extendable, but minimal implementation requirements
- Security mechanisms and transport layer encryption - SIPS

SIP Components

- SIP Addresses: URIs

  Telephone numbers, sip:user@domain, sip:phone_number@host, ...

- SIP Messages

  HTTP-like transactions: sip://<request-URI> request $\rightarrow$ response

- User agent server / SIP Server

  Receives session requests, may perform service registering & control, AAA, proxying, location services, ...

- User agent client / SIP Client

  Initiates a session

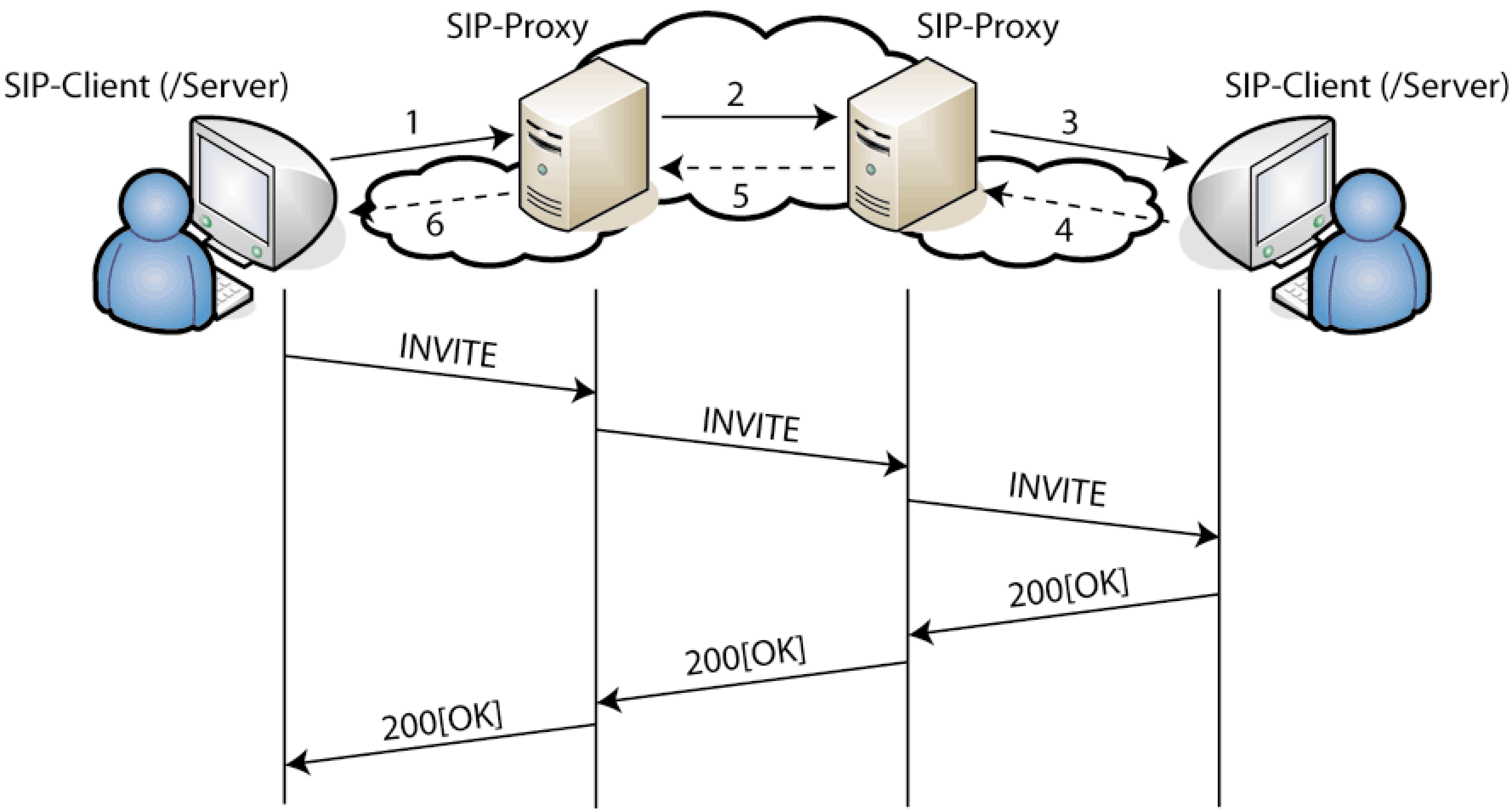
- SIP Protocol

  Peer-to-peer protocol between UACs and UASs

SIP Protocol (contd.)

- SIP is a multi-layered application protocol
  - Upper layer: Transaction user
  - Third layer: Transaction process layer
  - Second layer: Transport layer
  - Low layer: Syntax & encoding
- Interactions between components are transactional
  - Every request needs at least one response
  - A SIP dialog is a P2P relationship between two User Agents that persists for some time
- SIP participants form an overlay
- Media traffic is in parallel to SIP traffic
  - Media session parameters are included in the SDP

SIP Session Initiation: User Transaction Layer



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

SIP Messages

Inspired by SMTP encoding: Text style & extension headers, borrows: `To, From, Date` and `Subject` header

- Generic Message:

```
Request-Line / Status-Line
*message-header

[message-body]
```

- Request (Request-Line):

```
Method Request-URI SIP-Version
```

- Response (Status-Line):

```
SIP-Version Status-Code Reason-Phrase
```

- Methods: `INVITE, ACK, CANCEL, BYE, REGISTER, +++`

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

SIP Message Example: Call Initiation

Transaction ID

```
INVITE sip:snoopy@dog.net SIP/2.0
Via: SIP/2.0/UDP pc.brown.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Snoopy <sip:snoopy@dog.net>
From: Charlie <sip:charlie@brown.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.dog.net
CSeq: 314159 INVITE
Subject: Tales from the Red Baron …
Contact: <sip:charlie@sun17.brown.com>
Content-Type: application/sdp
Content-Length: 142

(Charlie's SDP not shown)
```

Member ID

Session ID

Response: Call Acceptance

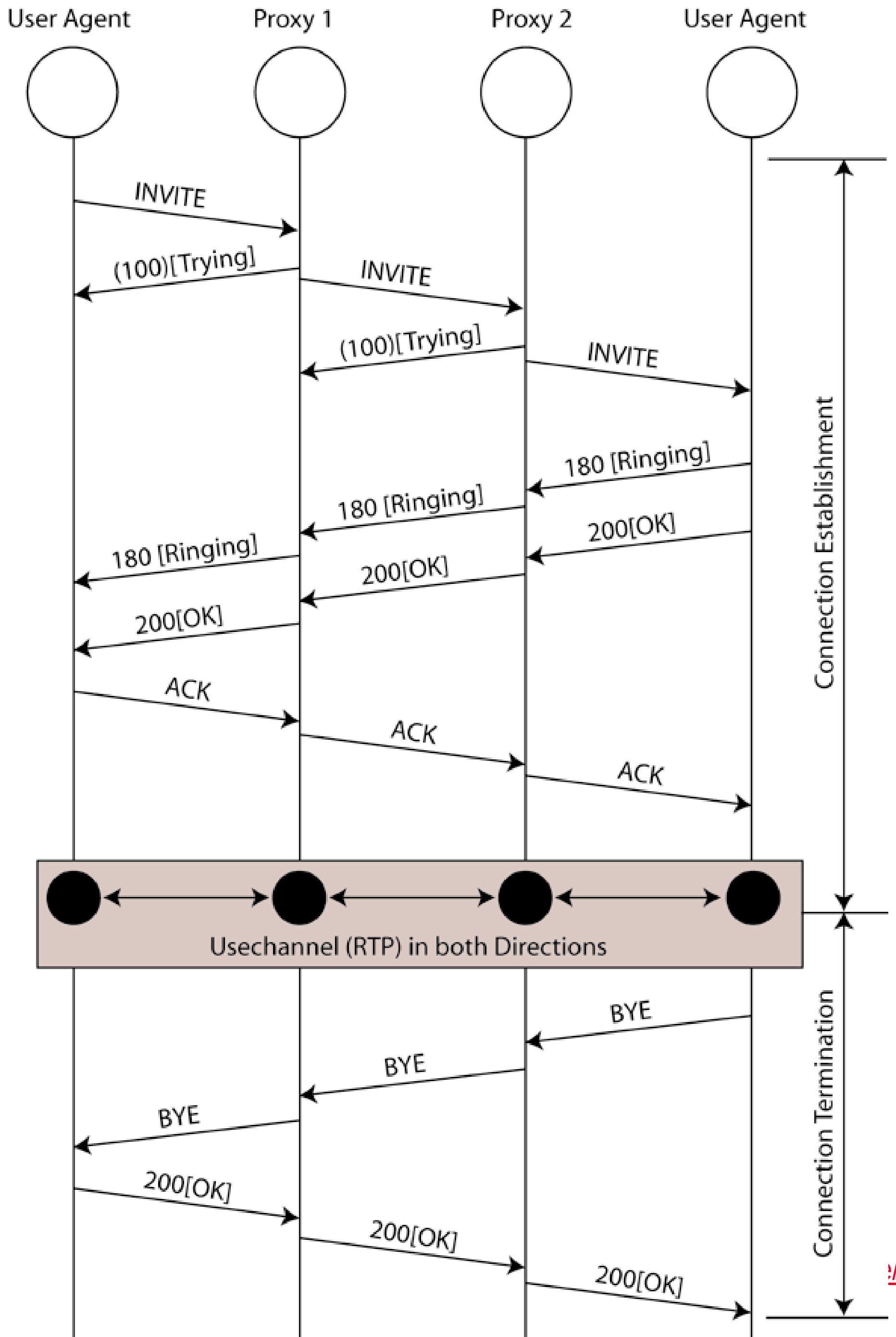
```
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy.peanuts.org;branch=z9hG4bK77ef
 ;received=192.0.2.2
Via: SIP/2.0/UDP pc3.brown.com;branch=z9hG4bK776asdhds
 ;received=141.22.13.122
To: Snoopy <sip:snoopy@dog.net>;tag=a79e45
From: Charlie <sip:charlie@brown.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.dog.net
CSeq: 314159 INVITE
Contact: <sip:RB.Snoopy@airterm.dog.net>
Content-Type: application/sdp
Content-Length: 148

(Snoopy's SDP not shown)
```

Proxy Transaction

Init. Transact

New Member ID

Same Session



Basic SIP Session Handling

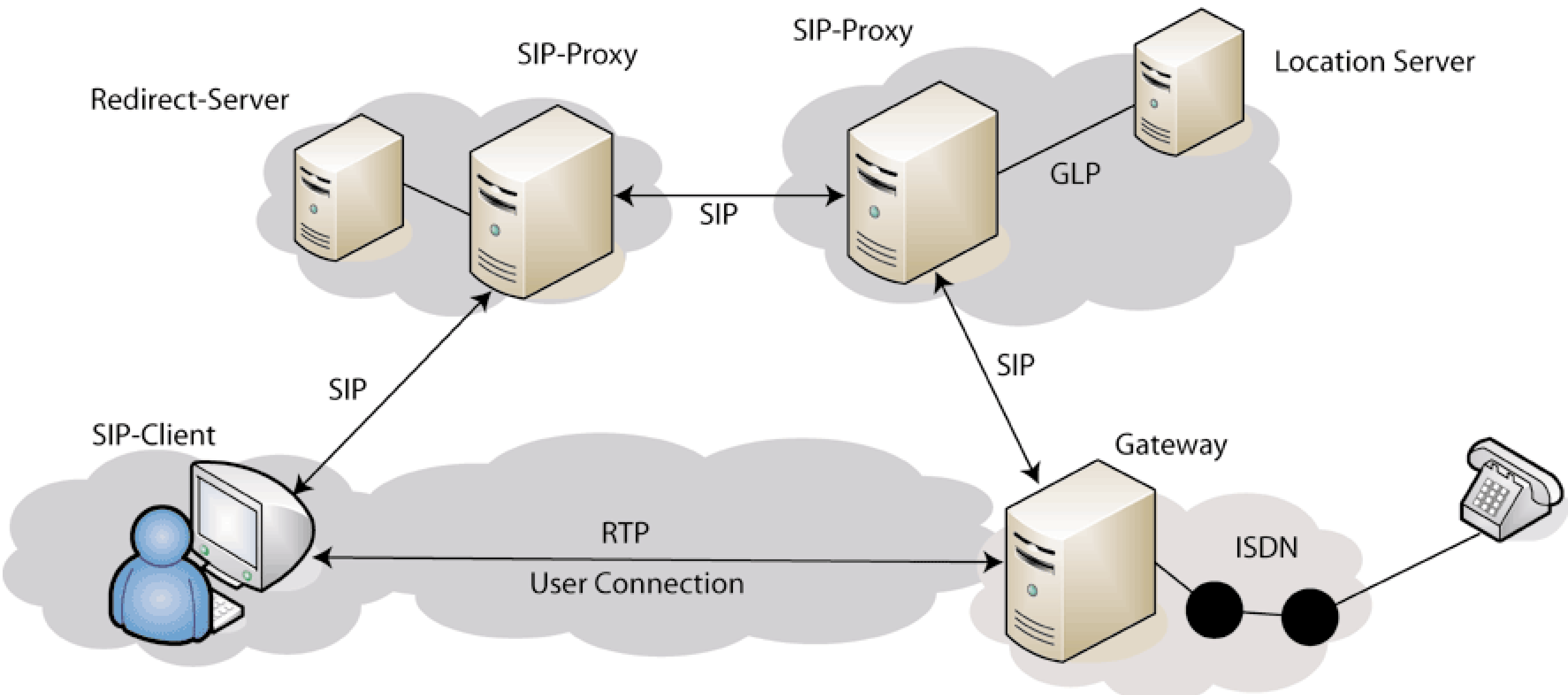
Registering with a Proxy

- A SIP Proxy server is an infrastructural entity for call routing based on presence information
- UAC may register with ‘their’ Proxies:

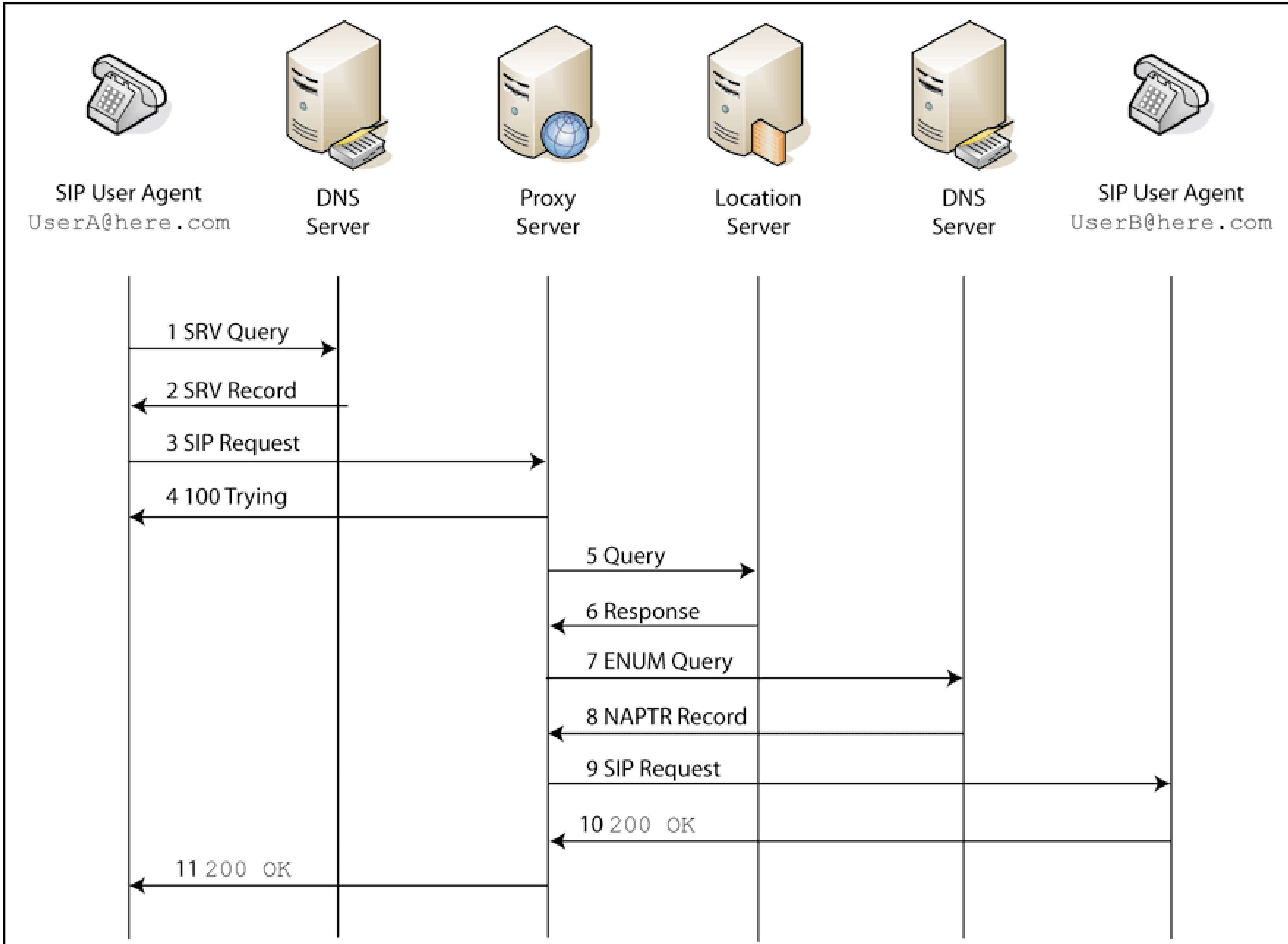
```
REGISTER sip:registrar.dog.net SIP/2.0
Via: SIP/2.0/UDP 141.22.8.8:5060;branch=z9hG687b
Max-Forwards: 70
To: Snoopy <sip:snoopy@dog.net>
From: Snoopy <sip:snoopy@dog.net>;tag=7654
Call-ID: 147@141.22.8.8
CSeq: 44 REGISTER
Contact: <sip:RB.Snoopy@airterm.dog.net>;expires=3600
Content-Length: 0
```


Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

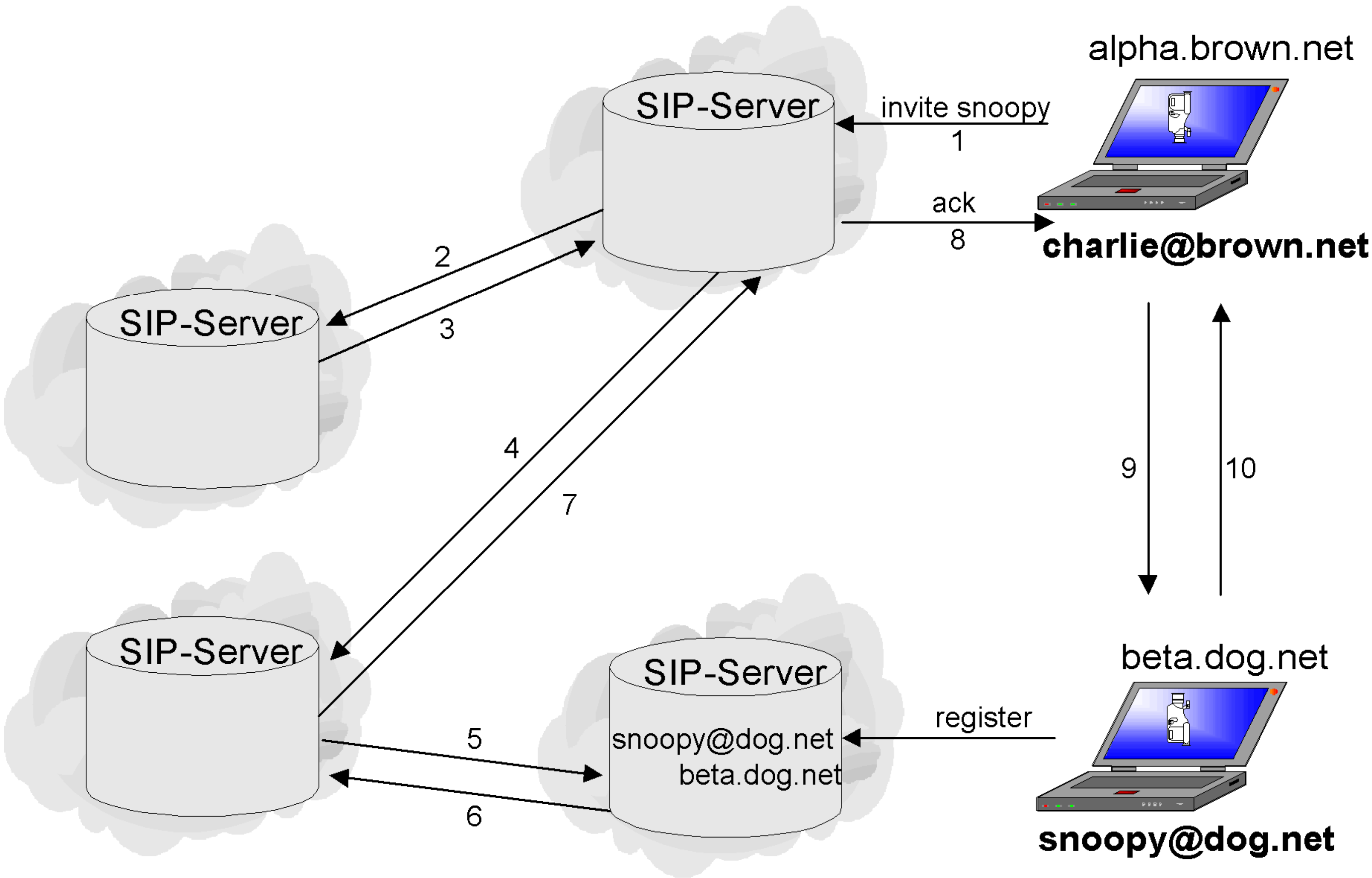
SIP Redirect and Gateway Services



SIP Address Resolution



SIP Locating Users/Servers



Extending SIP

SIP's functionality can be easily extended by adding new 'Request-Response dialogs':

1. Define new Request Methods

   `Examples: JOIN, SUBSCRIBE, MESSAGE, …`

2. Define appropriate Response Status-Lines

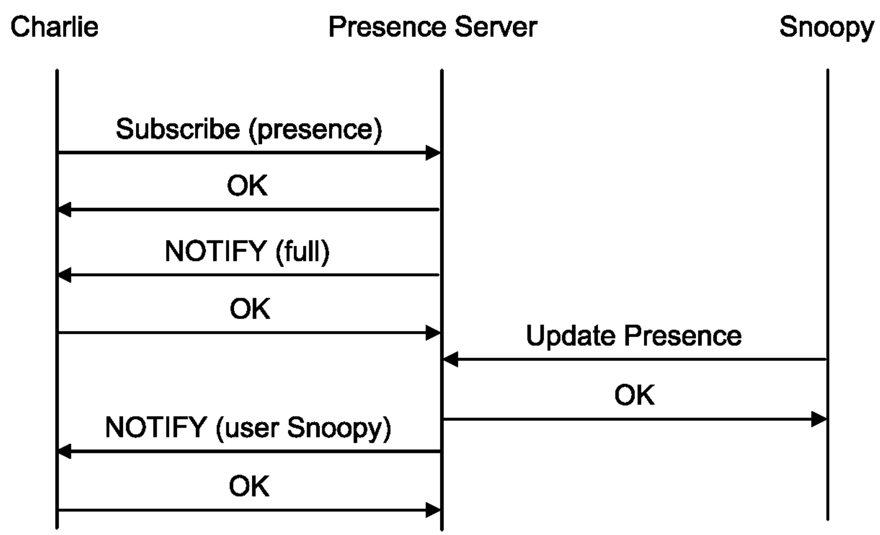
   `Examples: MOVED, TURN, …`

3. Define call sequence behaviour

Numerous RFCs and I-Drafts around

SIP Event Packages

- States of SIP services can be extended to event-type notifications (RFC 3265)
- Event information are encoded in XML as “Event Packages”
- New methods: `SUBSCRIBE` and `NOTIFY`
- Many new functions, e.g.,
  - Invite dialog state
  - Feature key events
  - Updating IMGs
  - Conferencing
  - Push-to-talk
  - Presence



SIP Presence Event Package

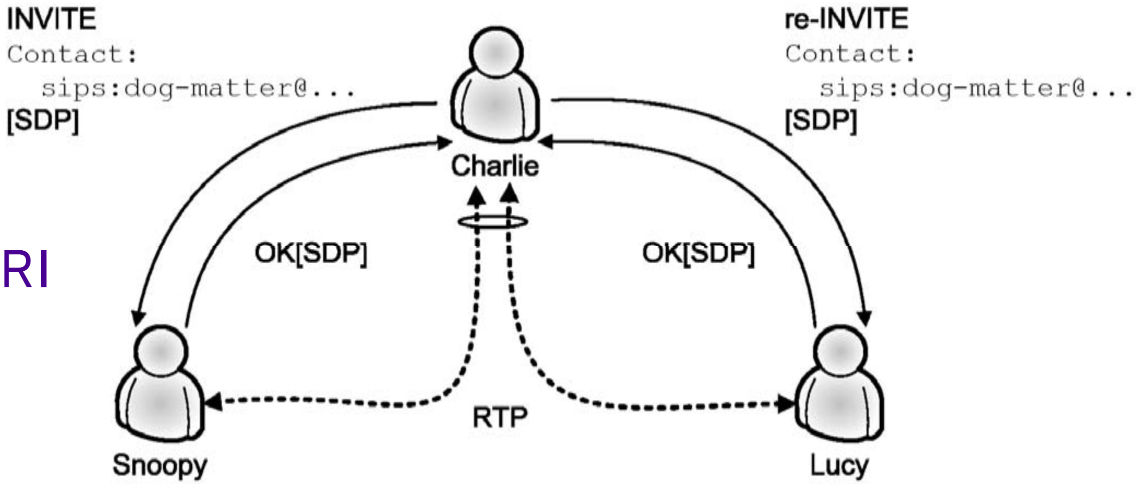
- Indication of online availability for community use – 'Buddy List' with prioritised contact info
- Conveys rich presence information on Activities (`playing`), Mood (`confused`), Place (`noisy in aircraft`), Relationships (`friend`), clear text Note ...
- Presence Information Data Format (PIDF, RFC 3863) - can be extended by personal attributes
- Commonly combined with Instant Messaging:
  - Short individual messages using the `MESSAGE` method
  - Session-based messaging using the MSRP protocol

Conferencing with SIP

- Support of multi-party sessions is a vital core function
- Conference: Instance of a multi-party conversation
- Many flavours of conferencing:
  - Centralized versus distributed
  - Ad hoc versus scheduled
  - Tightly versus loosely coupled
- Rich application domain:
  - Audio-/ videoconferencing
  - Distributed gaming (MMOGs)
  - Presence & Instant Messaging services
  - Foreseen as part of the IMS (MBMS)

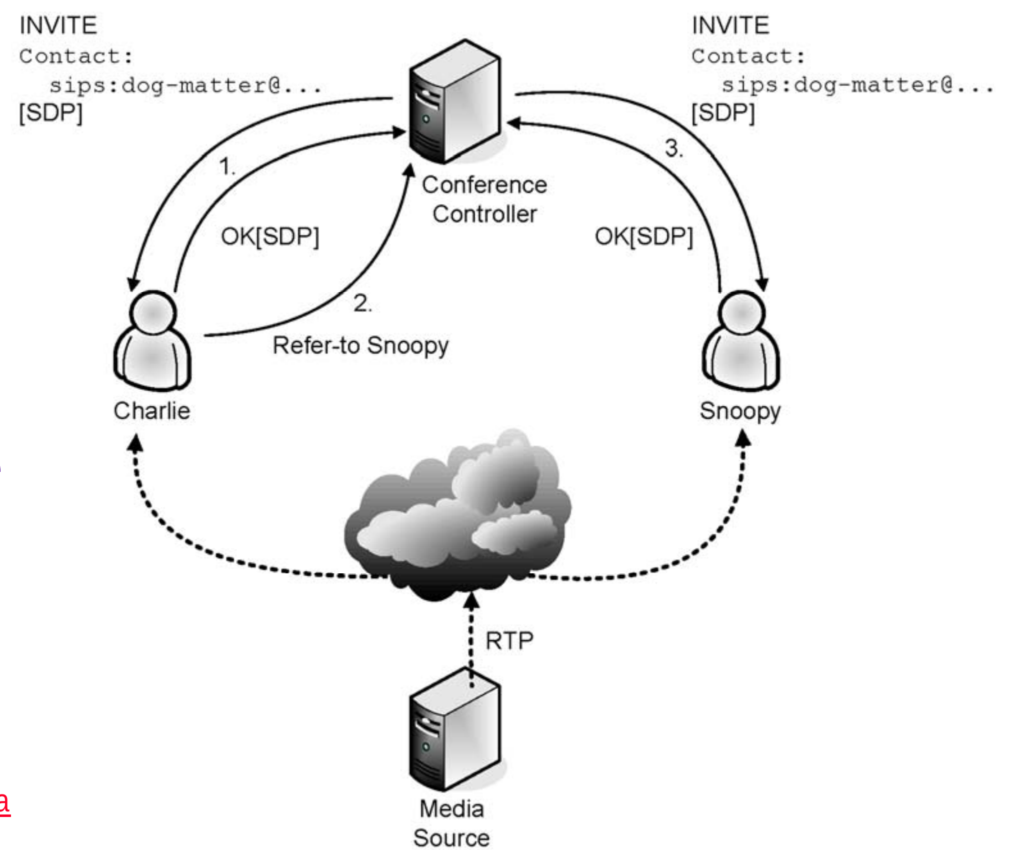
3-Way Conference

- Typical Scenario: Two parties in a call extend conversation to a 3rd member (ad hoc)
- Could be handled implicitly by application, but
  - No explicit group context (wiretapping!)
  - No way to switch relaying party
- SIP introduces conference Focus in Contact header
  - explicit conference URI
  - `isfocus` tag

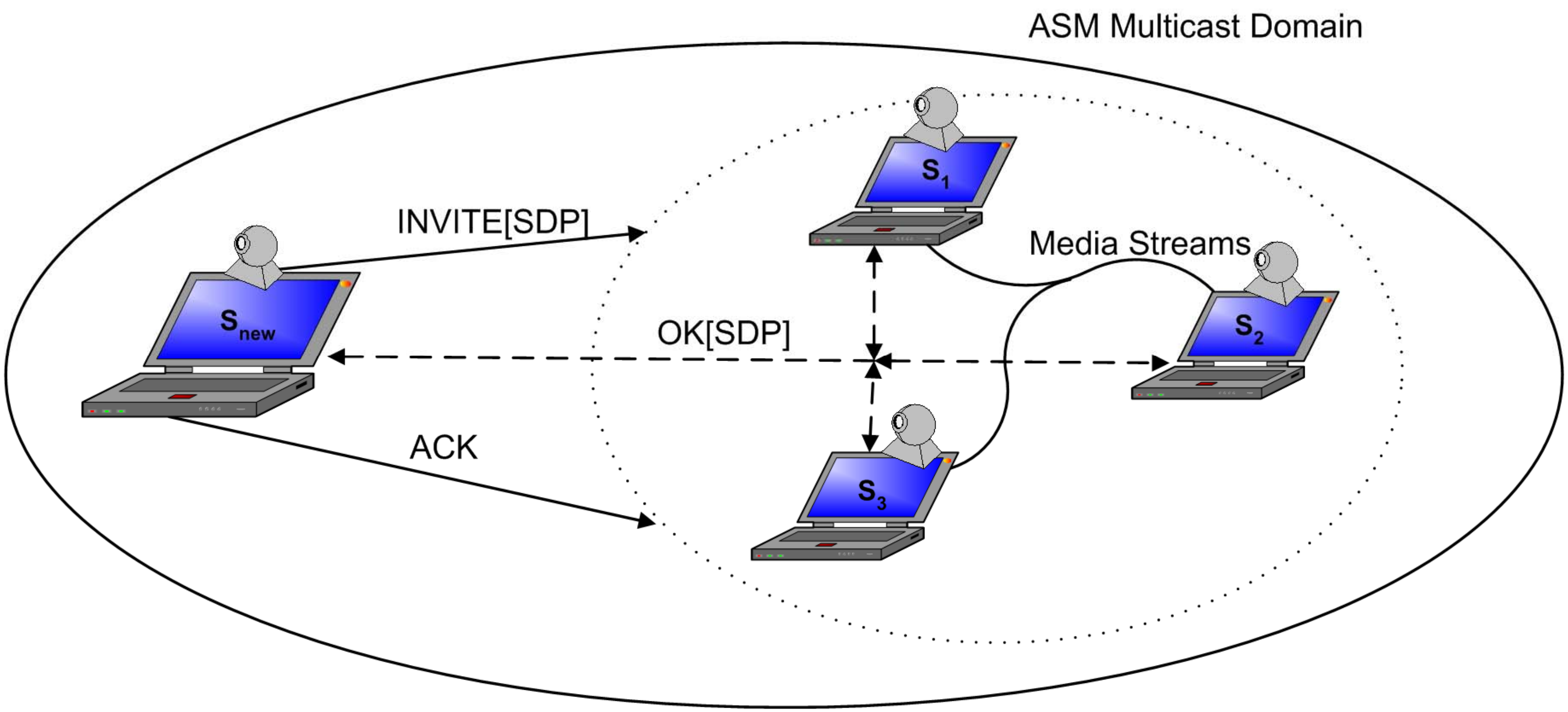


Large Scale Conferences

- Conference control by a dedicated conference controller or via multicast signalling
- Media distribution decoupled, typically by multicast or a (strong) MCU
- Additional functions
  - REFER - 3rd party invite
  - conference event states
  - floor control



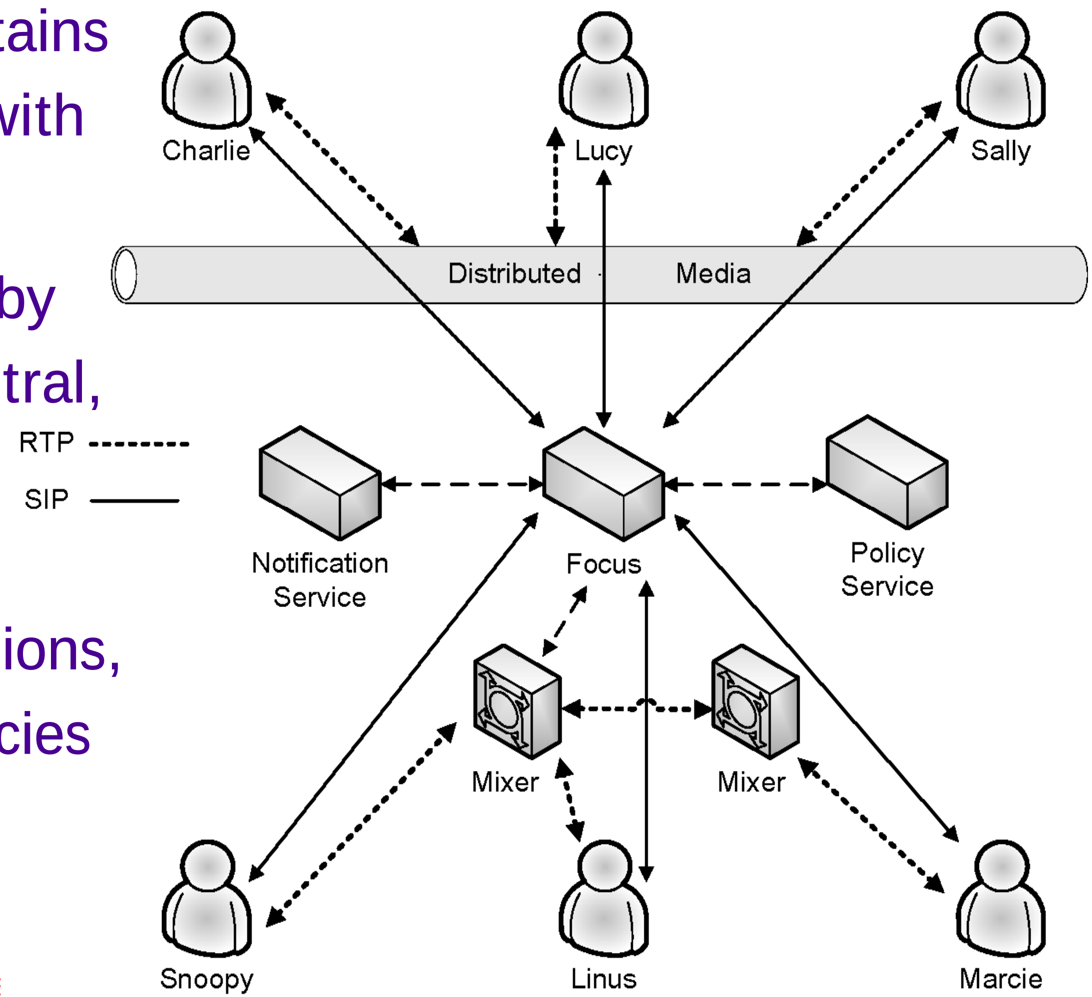
SIP via Group Communication



- S_{new} sends its `INVITE` to (*,G)
- All group members answer to (*,G)
- Out-of-Band agreement on addressing & SDP

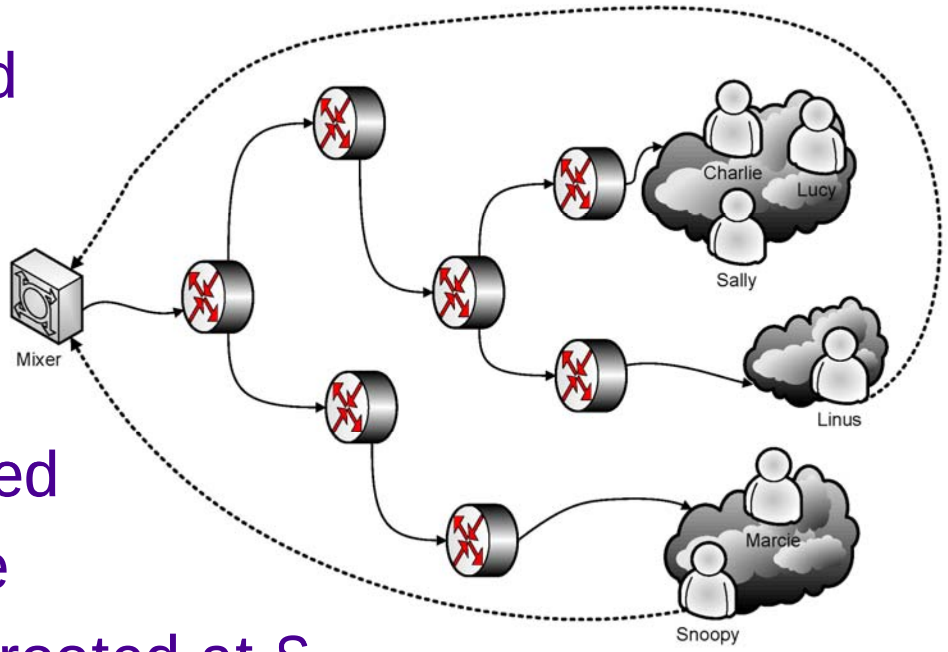
Architecture of Tightly Coupled Conferences

- Centralised focus maintains signalling relationship with all members
- Directs media streams by conducting mixers (central, cascaded) or use of multicast media
- Additional service functions, e.g., Notification & Policies



Media Distribution via SSM

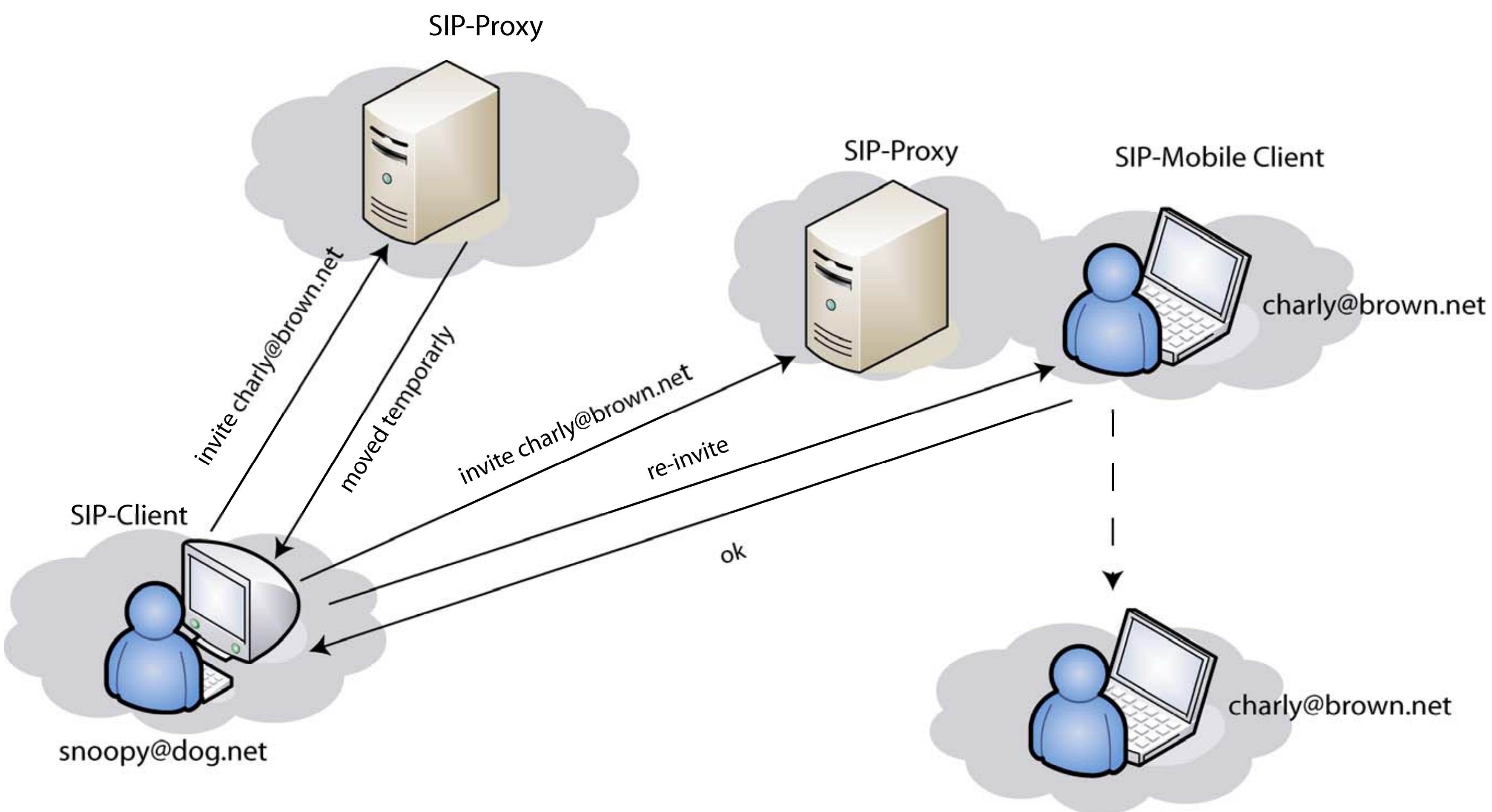
- Media distribution in a tightly coupled conference may be centralised based on SSM
- All streams are submitted to one mixer S
- Each member subscribes to (S, G)
- Media flows are distributed according along a Source Specific distribution tree rooted at S



Application Layer Mobility with SIP

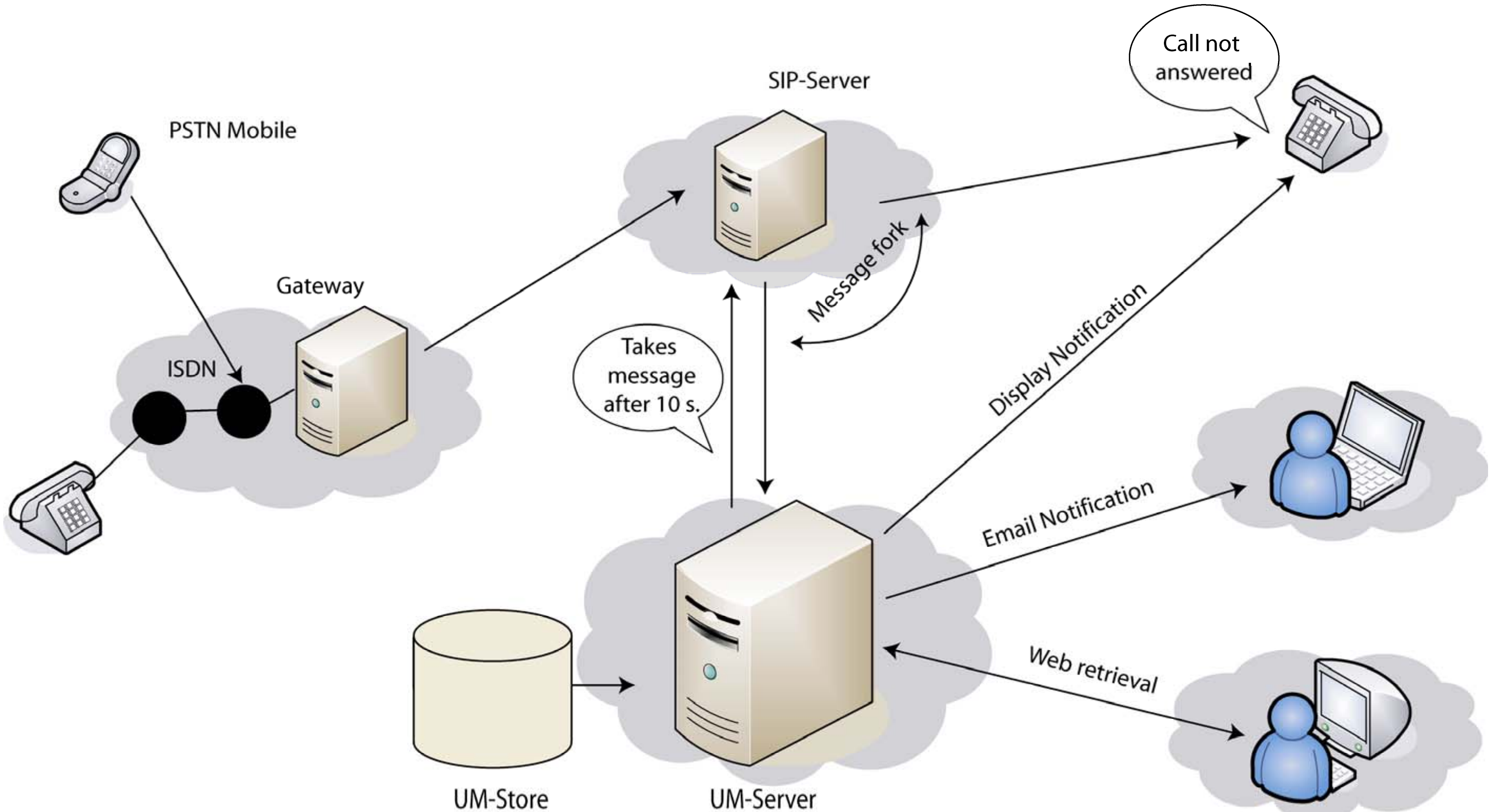
- Two types: personal & midcall mobility
- Personal mobility:
  - Multiple registration: with home and visited registrar
  - On call registrar returns “temporarily moved to”
- Midcall mobility
  - Mobile host issues a re-INVITE with its new session & contact data

SIP Mobility



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

SIP Unified Messaging



SIP Programming

A general purpose Java SIP stack is JAIN SIP (http://jain-sip.dev.java.net)

Java SIP stacks are also available from the Java Community Process

Server Side: SIP Servlet API (http://jcp.org/en/jsr/detail?id=116)

Terminal Side: SIP API for J2ME (http://jcp.org/en/jsr/detail?id=180)

Core architecture:

- One SipStack (interface) with several SipProviders, sending or receiving Request/Response messages
- SIP address factory
- SIP header factory
- SIP message factory

Many commercial C/C++ SIP stacks. Open Source Versions:
GNU: oSIP (http://www.gnu.org/software/osip)
reSIProcate (http://www.resiprocate.org/) – Minimal UAC example here

Reading

- D.B. Johnston: SIP, Artech House, 2003.
- Sinnreich, Johnston: Internet Communications Using SIP, 2nd Ed. Wiley & Sons, New York, 2006.
- Syed Ahson, Mohammad Ilyas (Ed.): SIP Handbook: Services, Technologies, and Security, CRC Press, Boca Raton, November 2008. Chapter on Group Conferencing here.
- TERENA: The IP Telephony Cookbook, March 2004, http://www.terena.org/activities/iptel/contents1.html.
- IETF Documents: www.rfc-editor.org .
- ITU Documents: www.itu.int .