



# Multimedia Networking Communication Protocols

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

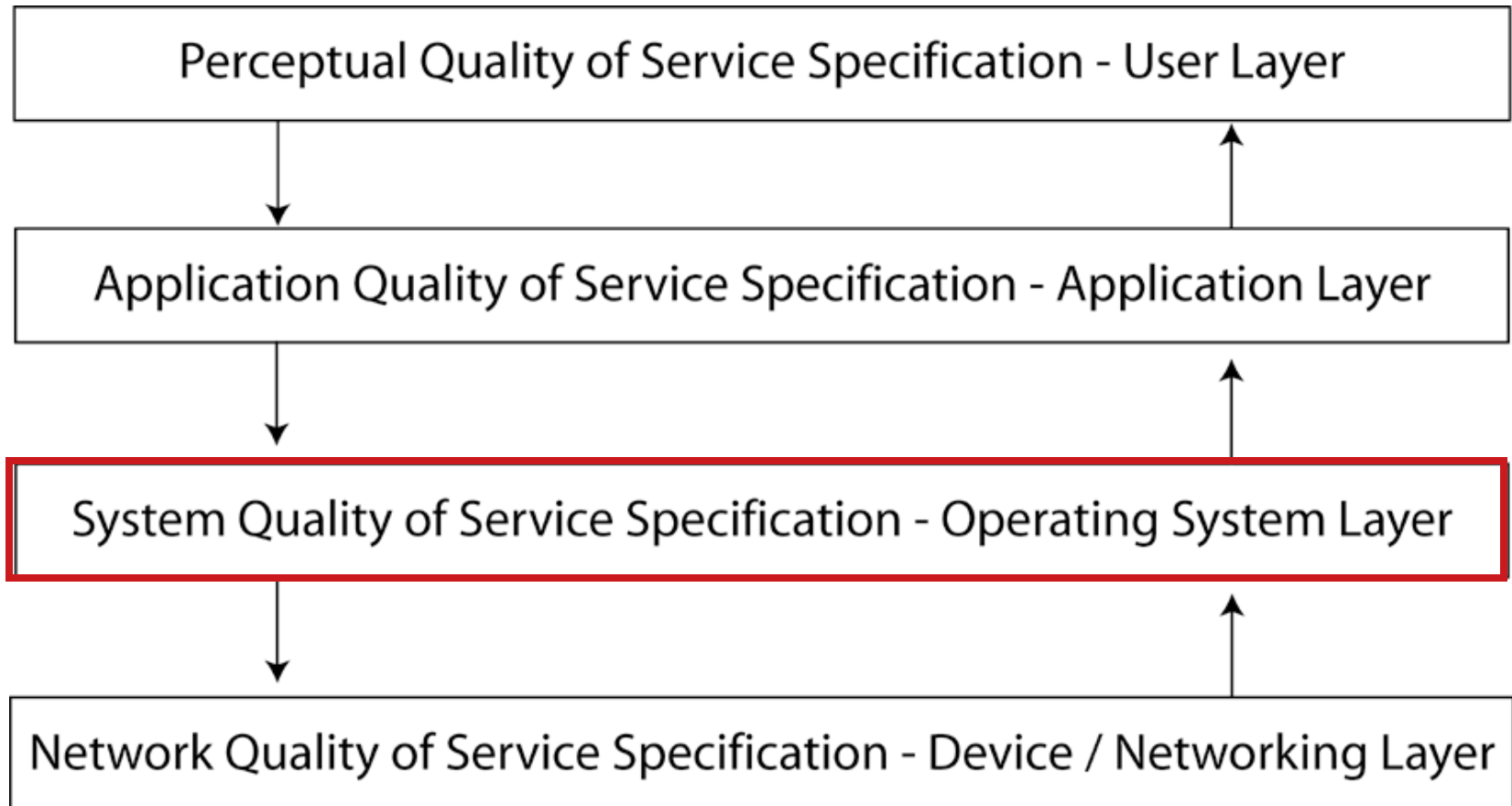


# Agenda

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

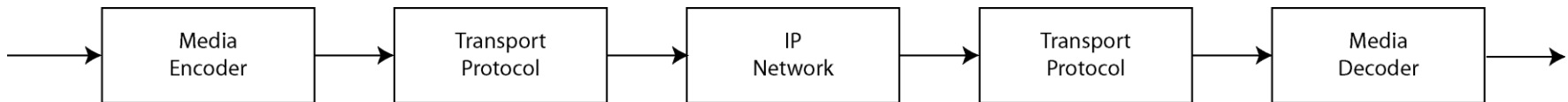


# QoS – Layered Model



# Multimedia Communication Across IP Networks

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



- o Provide application-specific framing.
- o Communicate media-specific intelligence & metadata.
- o 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

- o ITU-T Recommendation for Voice/Video conferencing over IP

  - Currently H.323 Version 4 (November 2000)

- o Transfers digital telephony onto IP Layer

- o Main functionalities

  - Bearer-Control-Function

  - Registration, Admission, Status (RAS)

  - Call Signalling

  - Gateway Service to PSTN

- o Widely implemented architecture, though legacy protocols  in use

# H.323 Interconnects

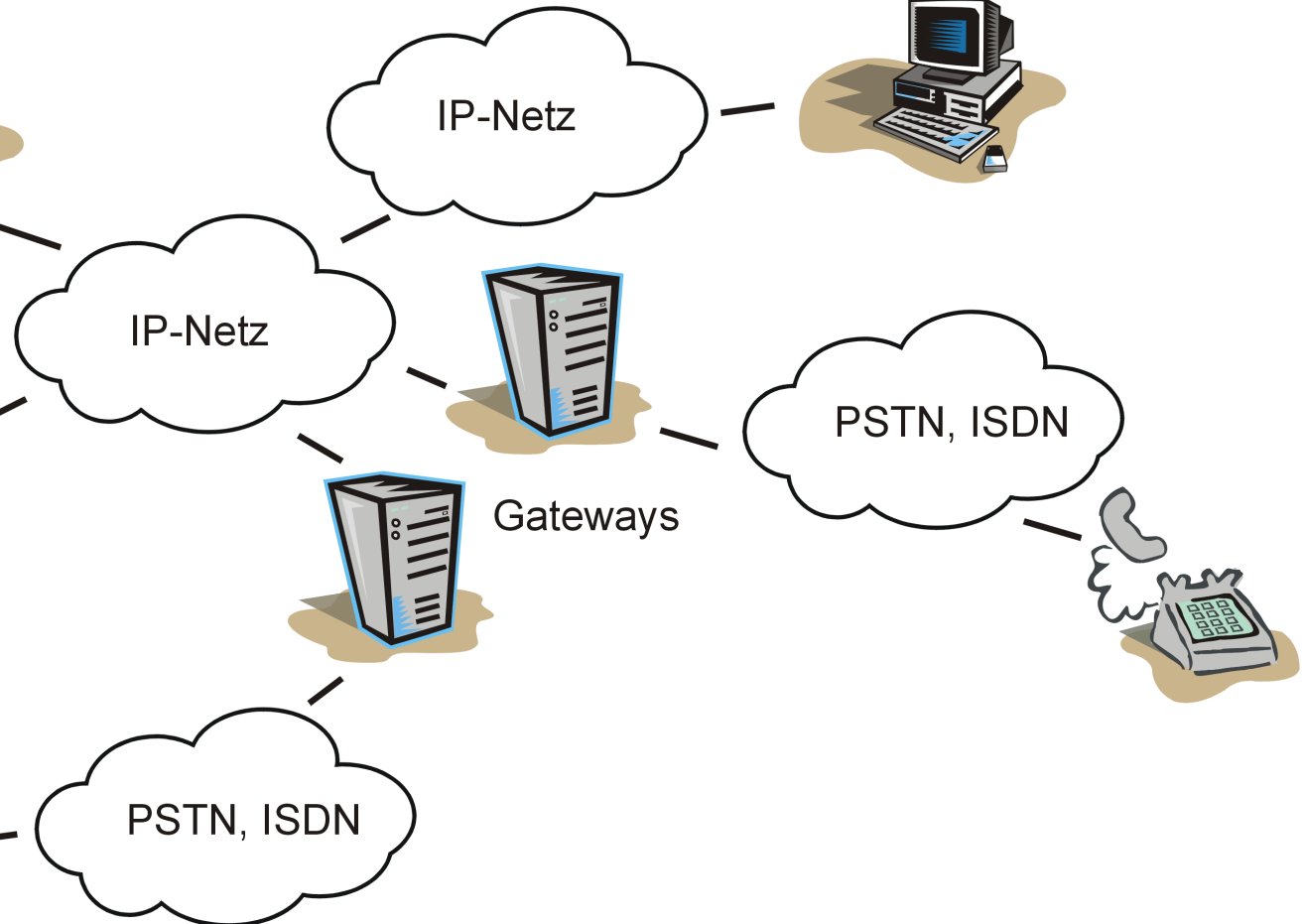
PC zu Telefon  
(z.B. Net2Phone)



PC zu PC  
(z.B. NetMeeting)



Telefon zu Telefon





# H.323 System Components

- o Terminal

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

- o Gatekeeper

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

- o Gateway

Gateway services between IP and PSTNs

- o Multipoint Conference Unit

Reflector server for group communication



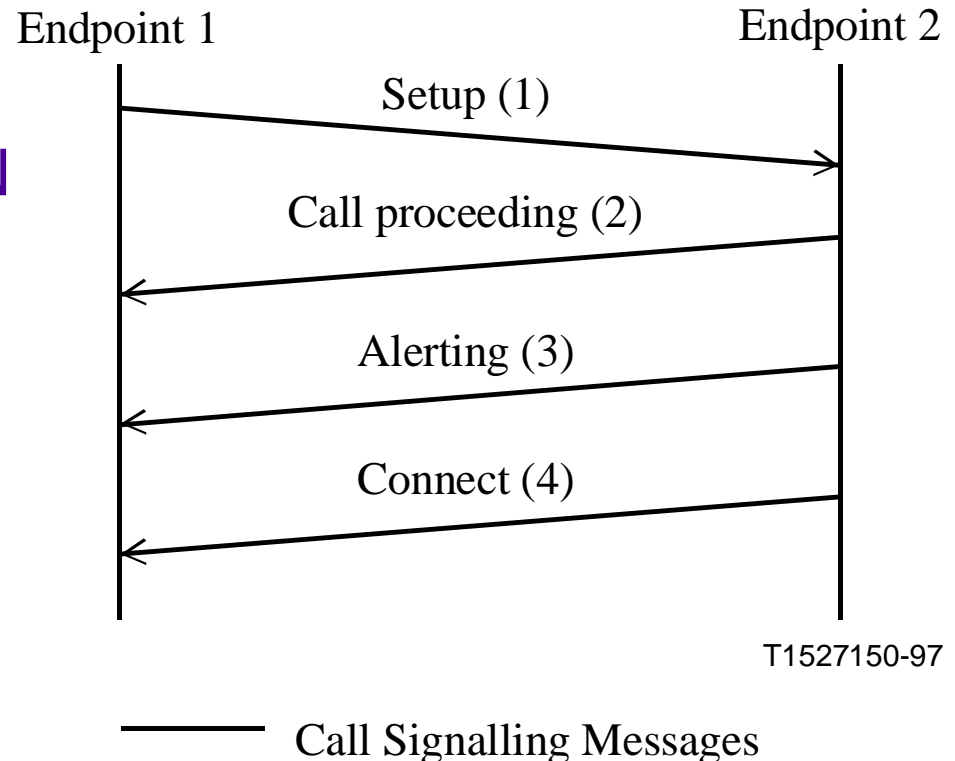
H.323-Standard						ISO-OSI-Reference	
Video Codecs	Audio Codecs	Management/ Control				7 - 6 - 5 -	A p p l i c a t i o n
H.26x	G.7xx GSM 6.10	R T C P	R A S	Signalling H.225.0	H.245		
RTP							
UDP			TCP			4	T r a n s p o r t
IP						3	
LLC / MAC – IEEE-802.x						2	
Fiber, Twisted Pair, ...						1	

# 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

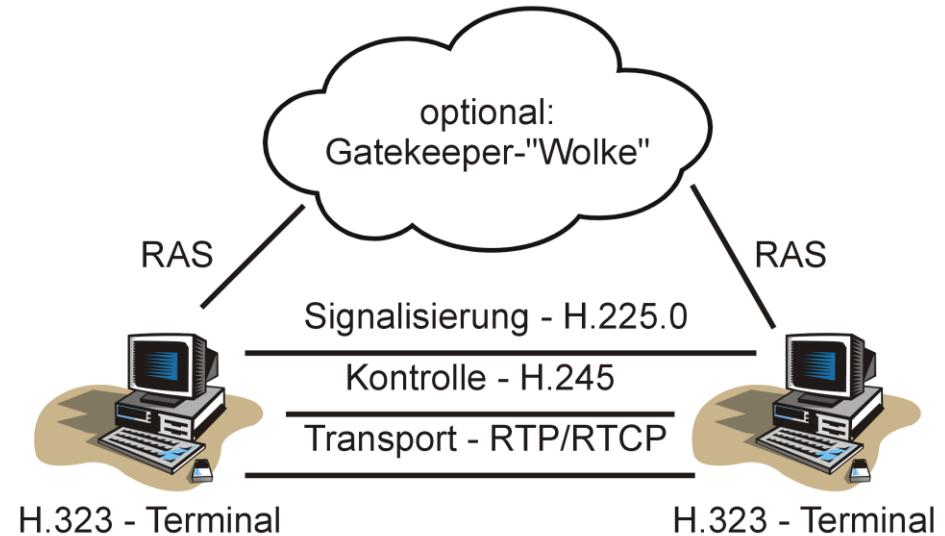


# H.323 Signalling:

## 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

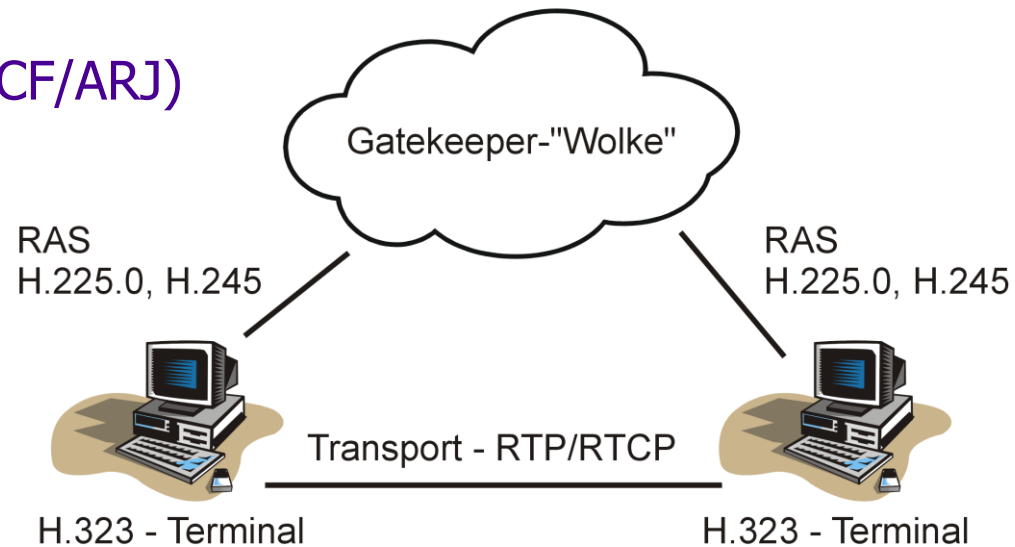
- setup
- ARQ - ACF/ARJ
- connect

### 3. Gatekeeper – Caller

- connect

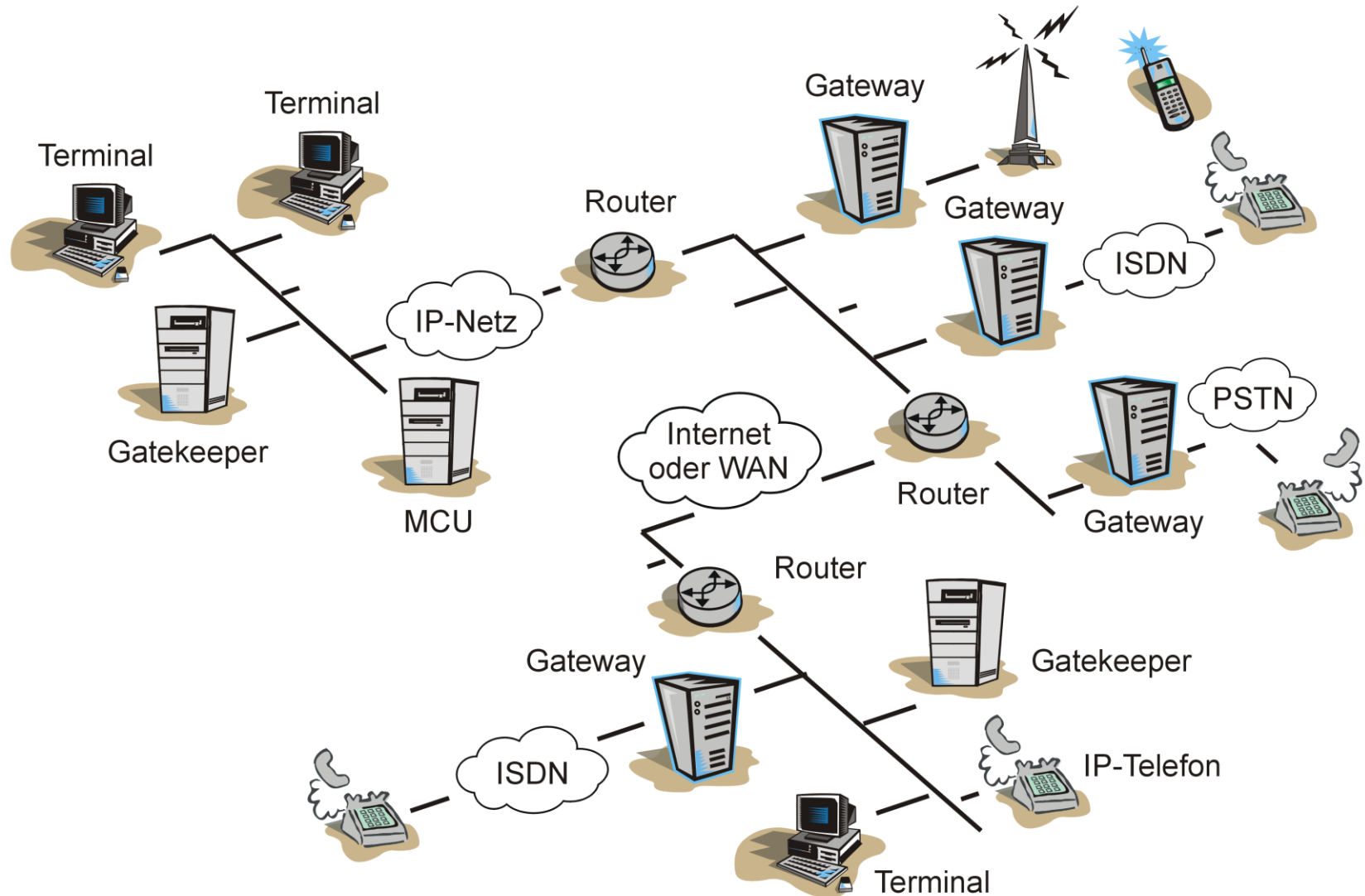
### 4. Caller – Gatekeeper - Callee

- Control Channel Established (H.245)



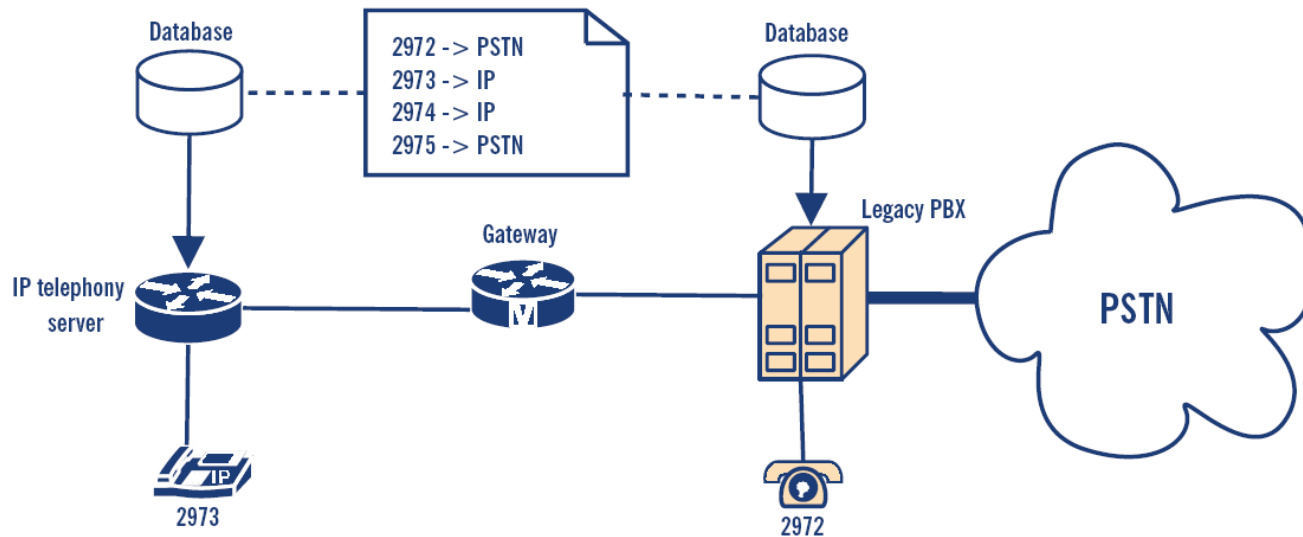
Call control requires operational  
Gatekeeper

# H.323 Scenario



# H.323 – Basic Configuration

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



o Configuring Interfaces + Services at Gateway

o Setting up Security (H.235 – rarely implemented)



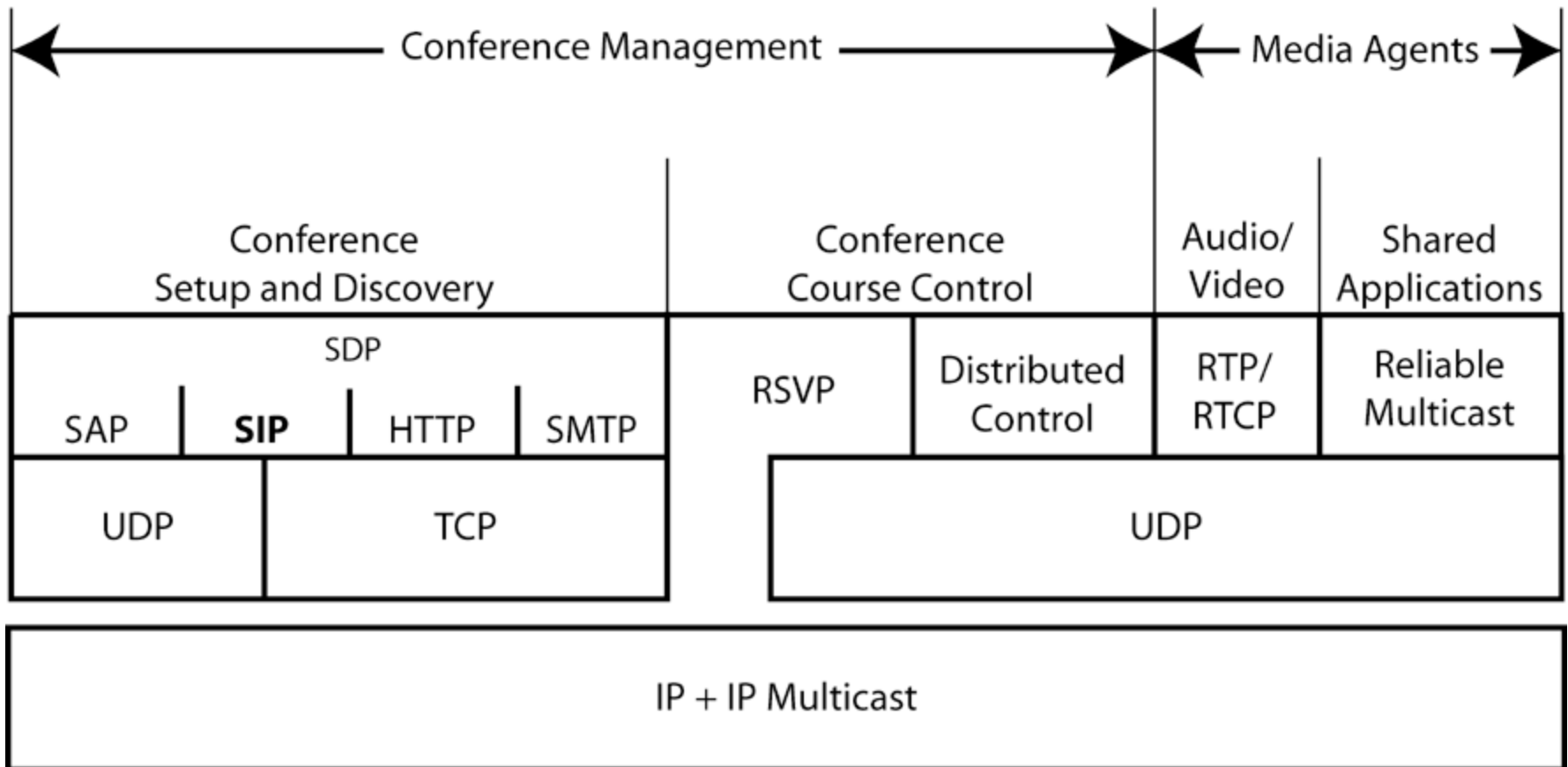


# Agenda

- 🕒 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 identifier

## 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

- o **Transport Address**

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

- o **RTP media type**

Any collection of payload types within a single RTP session

- o **RTP session**

One communication between a pair of transport addresses

- o **RTP multimedia session**

A set of RTP sessions among a common group of participants

- o **Mixer**

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



# RTP Entities (2)

- o Synchronisation source (SSRC)

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

- o Contributing source (CSRC)

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

- o Translator

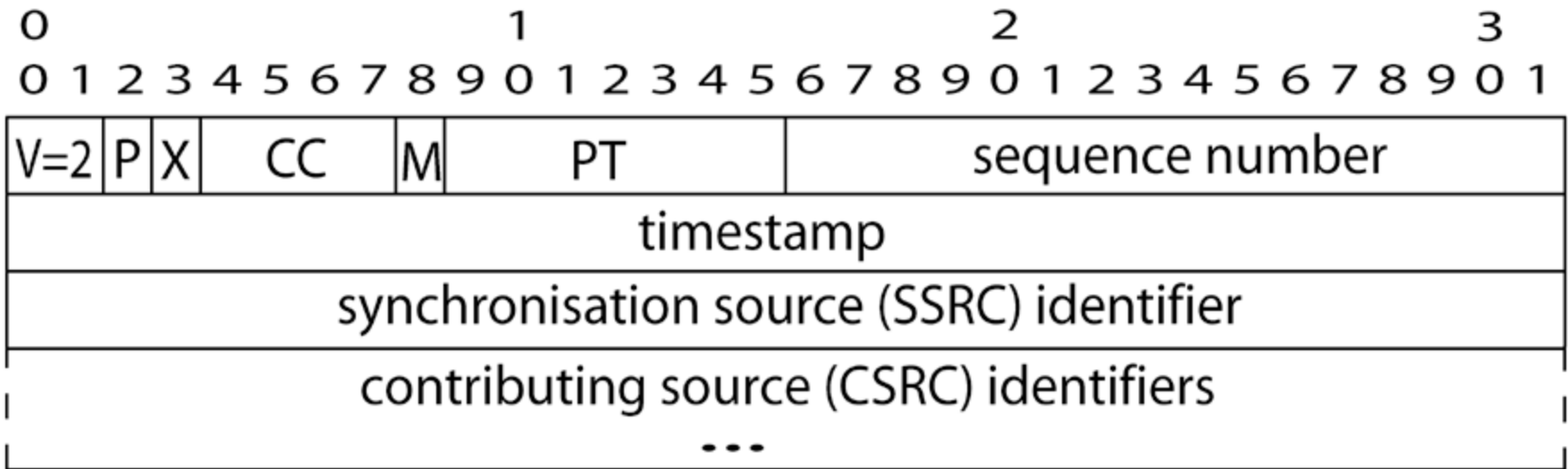
An intermediate system forwarding RTP packets without changing SSRC, but possibly modifying payloads

- o Monitor

An application receiving RTCP packets for diagnostics



# 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**:

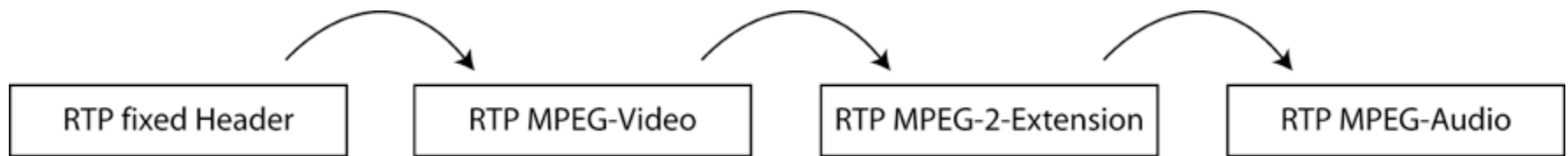
- o **Payload Type** – Identifies format and interpretation of the RTP payload (Audio/Video: RFC 3551)
- o **Marker** – Interpretation of the Marker is defined by a profile, e.g. marking frame boundaries
- o **Extension Headers** – May be defined in Profiles to carry additional, specific information





# RTP Profiles: Header Chain

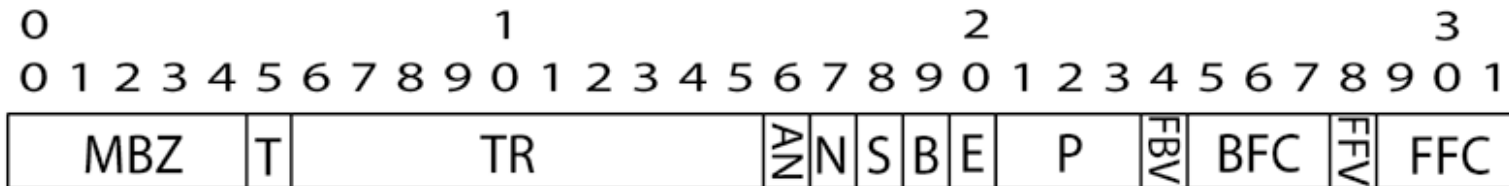
RTP allows for the encoding of media-specific information by possible (a chain of) **Extension Headers**.



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



# RTP MPEG Extension Header



**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

# Real-time Transport Control Protocol

- o 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).
- o 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
- o RTCP facilitates flow control & adaptive coding, but also multicast session surveillance
- o RTCP reports adapt to network capacities and session members



# 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, ...

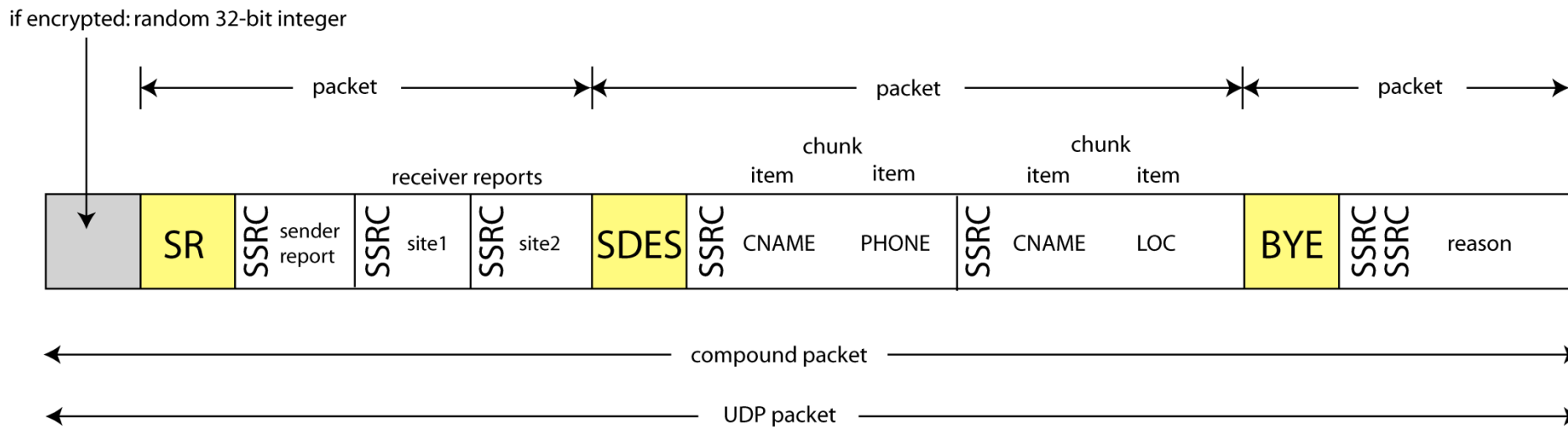
**Bye:** Leave Session

**Application Specific Functions**



# 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

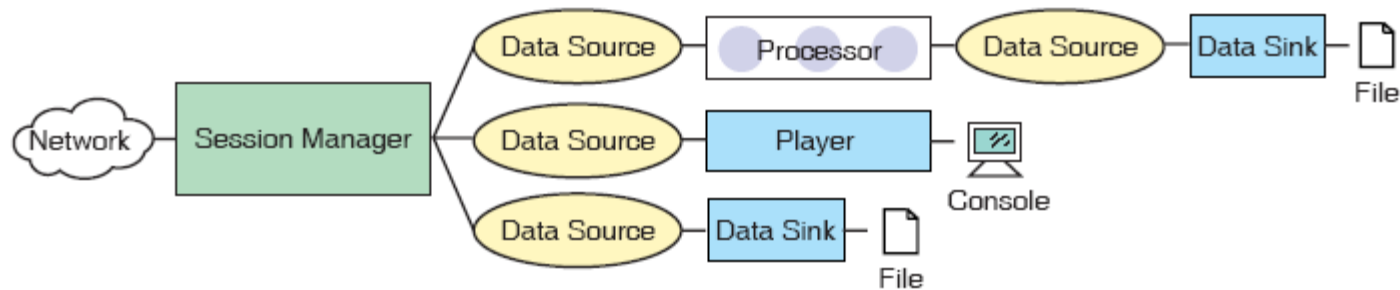
**RTP Data:** Predefined audio & video formats (extensible), transport protocol independent data handlers with input and output data streams

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

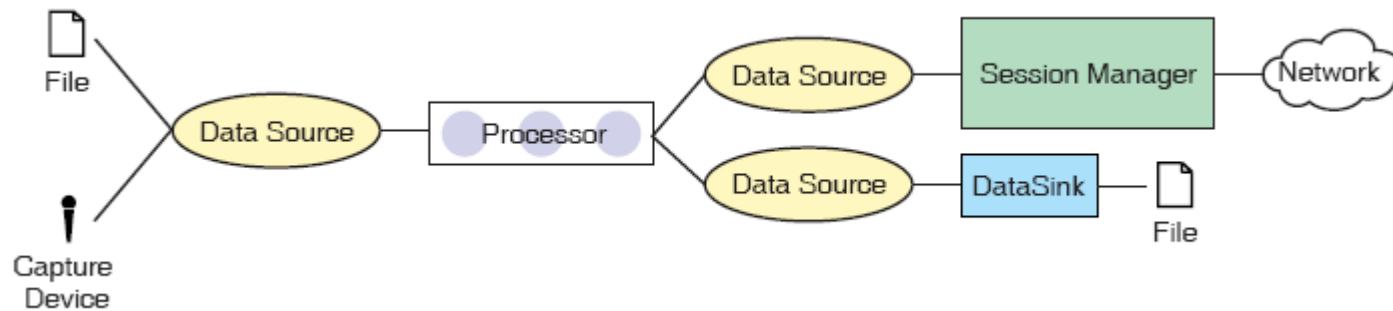


# RTP Programming (Java)

## RTP Reception



## RTP Transmission



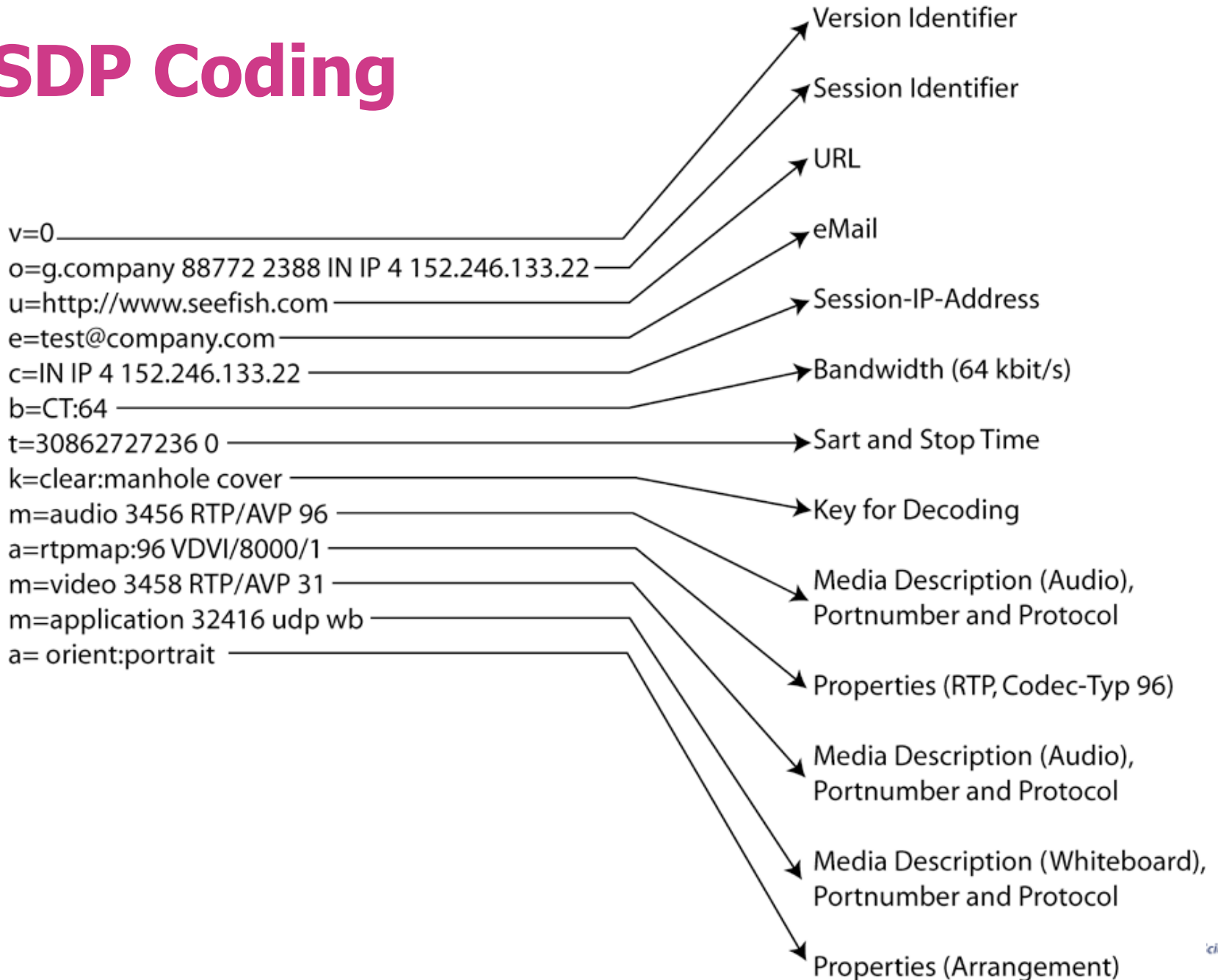


# SDP

## 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 Designed for the purposes
  - Session announcement (e.g. via SAP)
  - Session invitation
  - Real-time streaming
  - Extension within MIME, e.g., in emails or http
- o SDP is only a format, independent of its actual transport

# SDP Coding



# 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

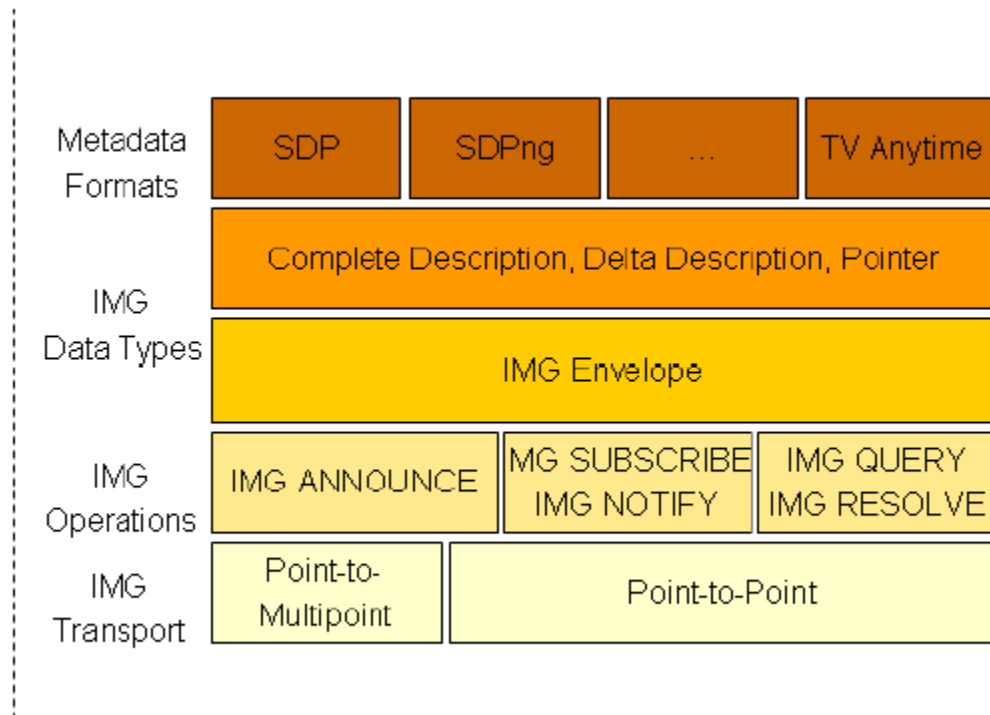
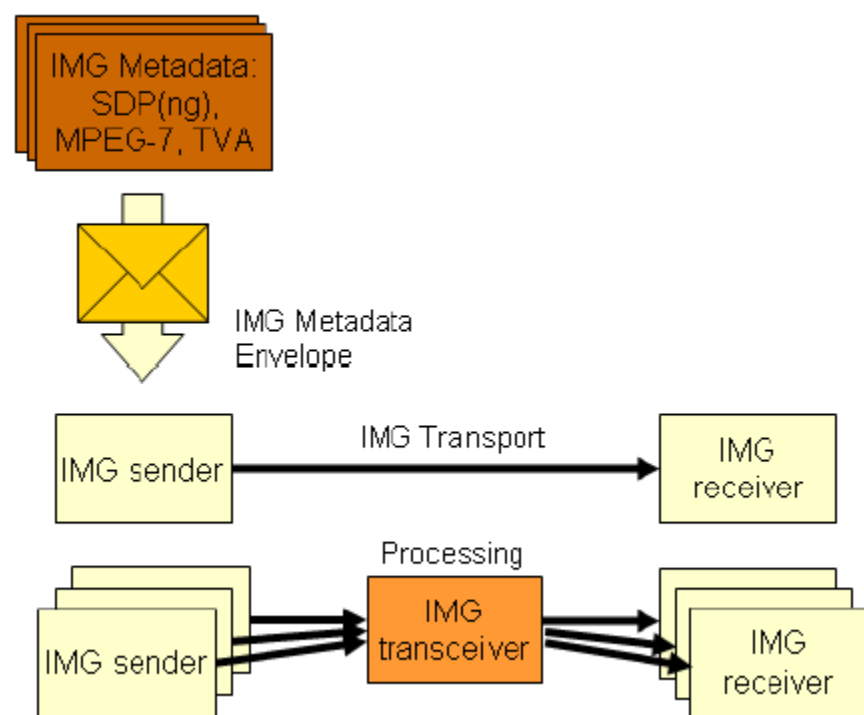
## o Internet Media Guide Framework

- General content description scheme derived from Electronic Program Guides (digital TV broadcasting)
- Current standardisation effort in IETF – s. RFC 4435
- Goal1: arbitrary content meta data support
- Goal2: interoperation of any suitable distribution mechanism (push/pull unicast, multicast, ...)



# Internet Media Guides

- o Abstract meta-data types: Complete, Delta, Pointer (URI to meta data)
- o Packaging in flexible envelopes
- o 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).



# 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
  - ➔ P2PSIP



# SIP - Session Initiation Protocol

- o IETF RFC 3261+ (v2, Schulzrinne et al 2002)
- o Signalling control protocol for multimedia sessions
- o 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
- o Increasing number of implementations for VoIP, conferencing, presence and messaging services





# SIP Protocol

- o End-to-end application protocol, transported via UDP or TCP
- o Designed to establish, modify and terminate stateful multimedia communication (sessions/conferences/instant messaging ...)
- o 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
- o Extendable, but minimal implementation requirements
- o Security mechanisms and transport layer encryption - SIPS

# SIP Components

- o SIP Addresses: URIs

Telephone numbers, sip:user@domain, sip:phone\_number@host, ...

- o SIP Messages

HTTP-like transactions: sip://<request-URI> request → response

- o User agent server / SIP Server

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

- o User agent client / SIP Client

Initiates a session

- o SIP Protocol

Peer-to-peer protocol between UACs and UASs

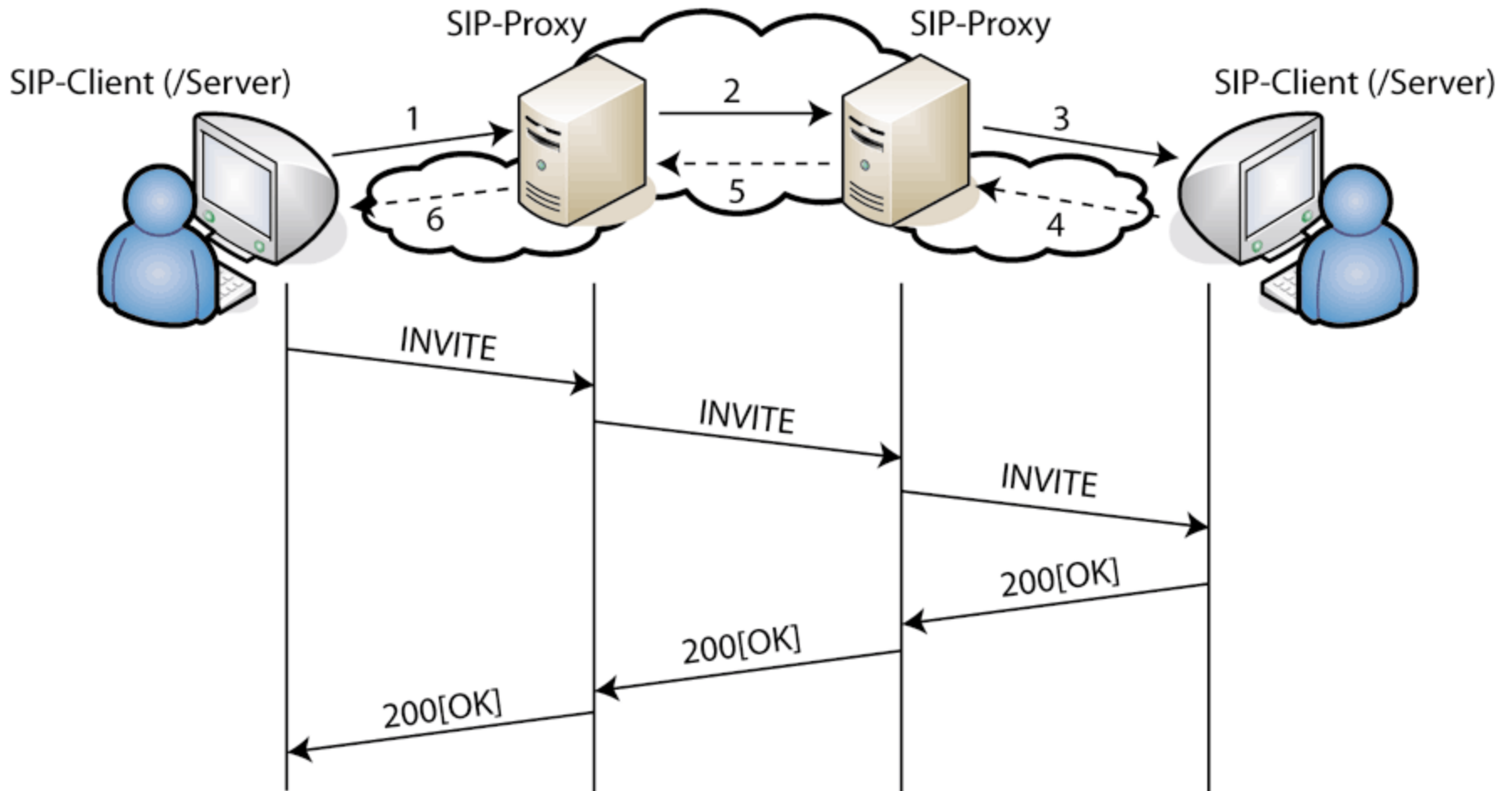


# SIP Protocol (contd.)

- o SIP is a **multi-layered** application protocol
  - Upper layer: **Transaction user**
  - Third layer: **Transaction process layer**
  - Second layer: **Transport layer**
  - Low layer: **Syntax & encoding**
- o 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
- o SIP participants form an overlay
- o Media traffic is in parallel to SIP traffic
  - Media session parameters are included in the SDP



# SIP Session Initiation: User Transaction Layer



# SIP Messages

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

## o Generic Message:

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

```
[message-body]
```

## o Request (Request-Line):

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

## o Response (Status-Line):

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

## o Methods: INVITE, ACK, CANCEL, BYE, REGISTER, ...



# SIP Message Example: Call Initiation

```
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
```

Transaction ID

Member ID

Session ID

(Charlie's SDP not shown)



# 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 pc.brown.com;branch=z9hG4bK776asdhs
    ;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)

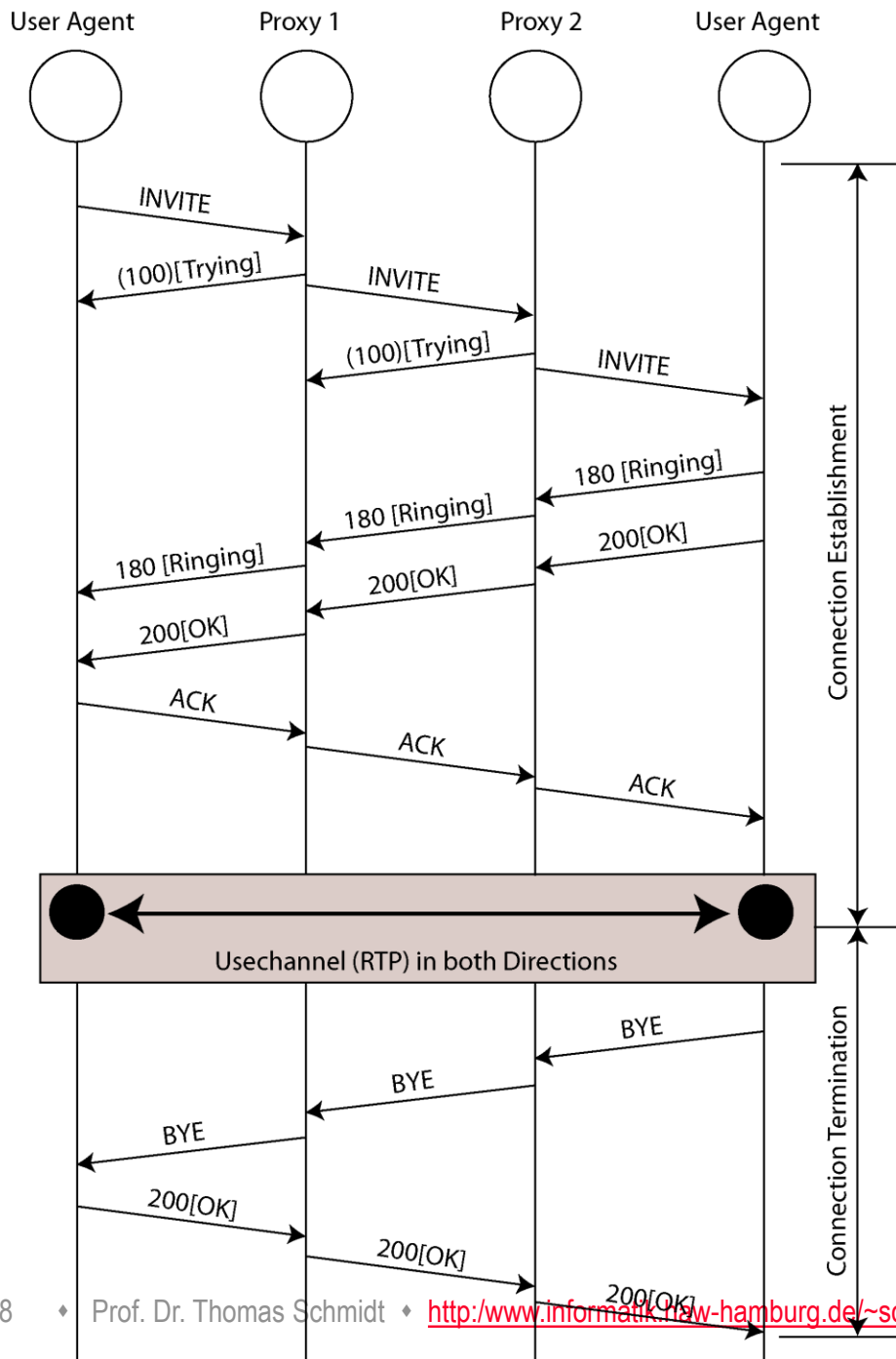
Proxy Transaction

Init. Transact

New Member ID

Same Session





# Basic SIP Session Handling





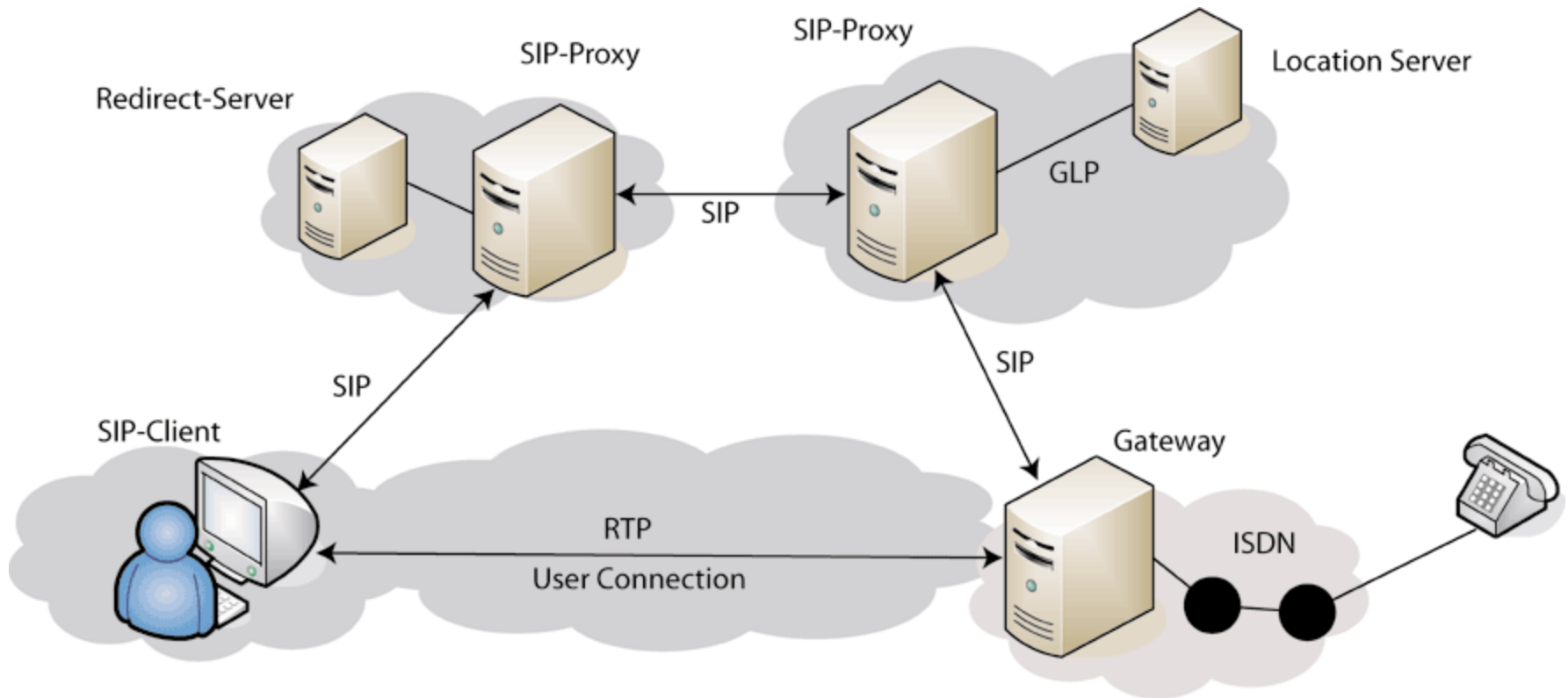
# Registering with a Proxy

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

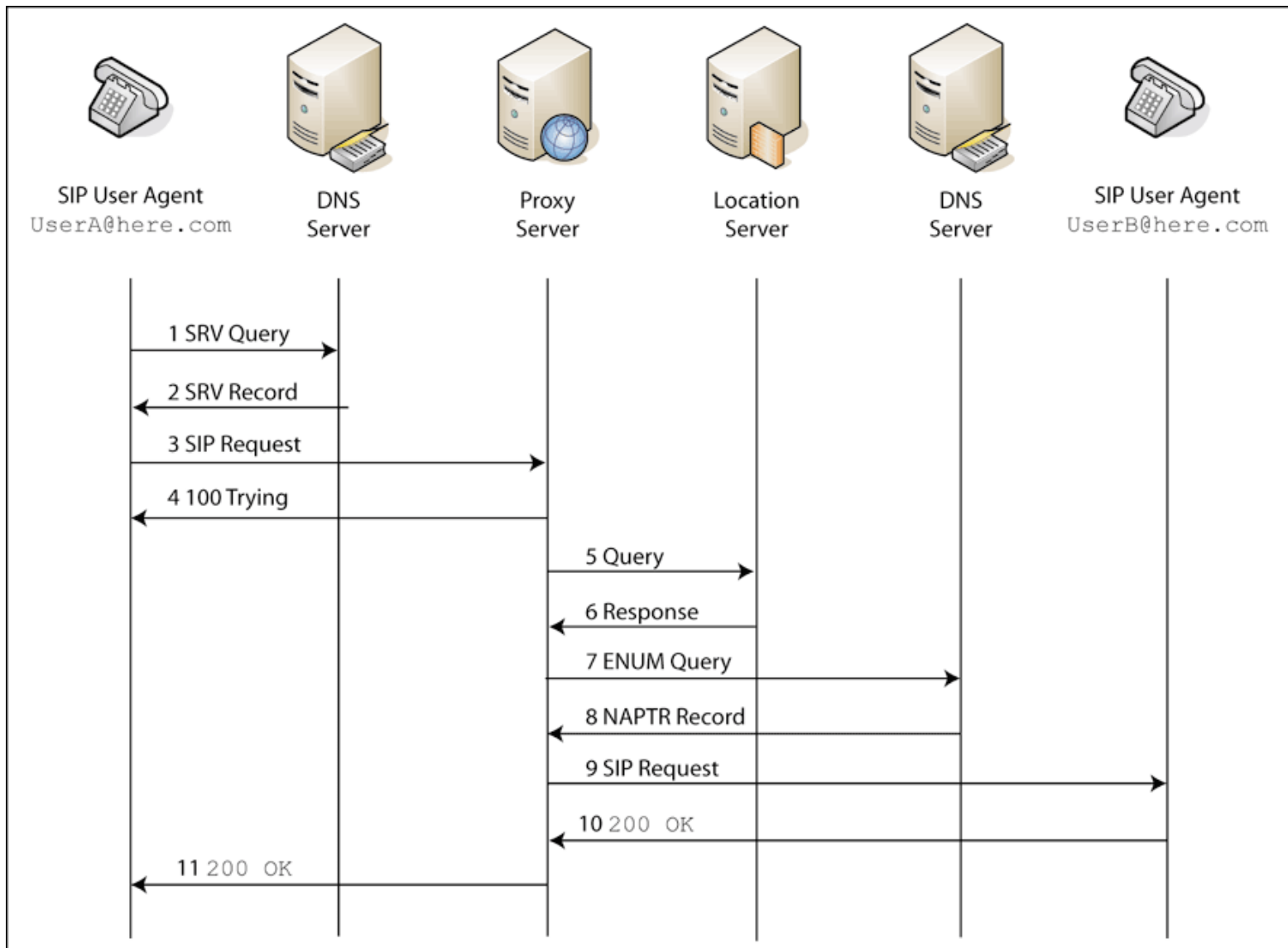
```
REGISTER sip:registrar.dog.net SIP/2.0
Via: SIP/2.0/UDP 141.22.8.8:5060;branch=z9hG687b
Max-Forwards: 70
To: Snoopy <sip:snoopy@dog.net>
From: Snoopy <sip:snoopy@dog.net>;tag=7654
Call-ID: 147@141.22.8.8
CSeq: 44 REGISTER
Contact: <sip:RB.Snoopy@airterm.dog.net>;expires=3600
Content-Length: 0
```



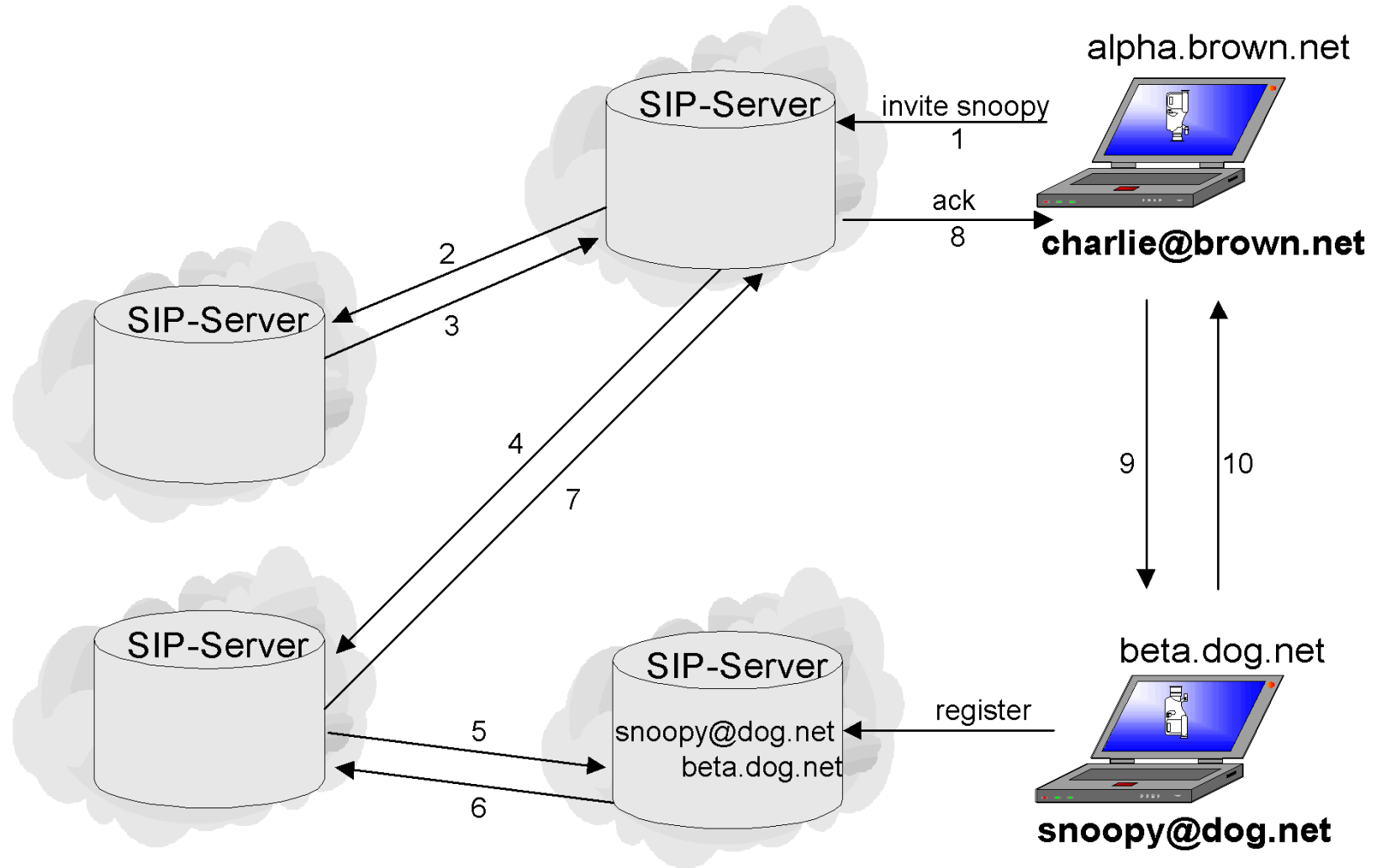
# SIP Redirect and Gateway Services



# 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, ...

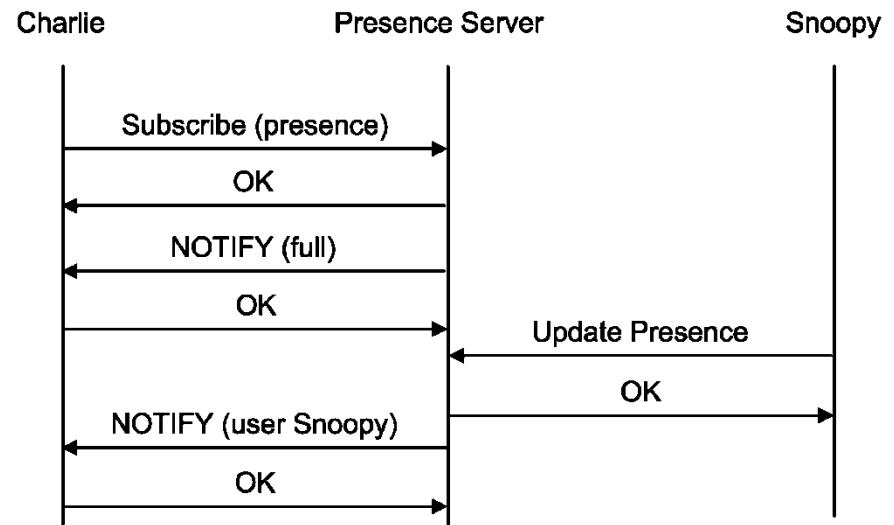
3. Define call sequence behaviour

Numerous RFCs and I-Drafts around



# SIP Event Packages

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



# SIP Presence Event Package

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



# Conferencing with SIP

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

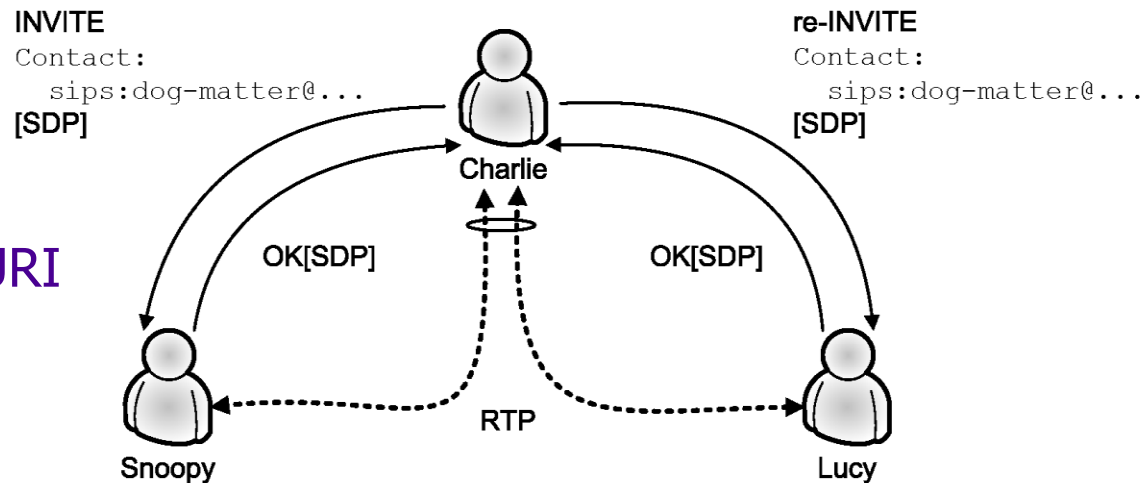




# 3-Way Conference

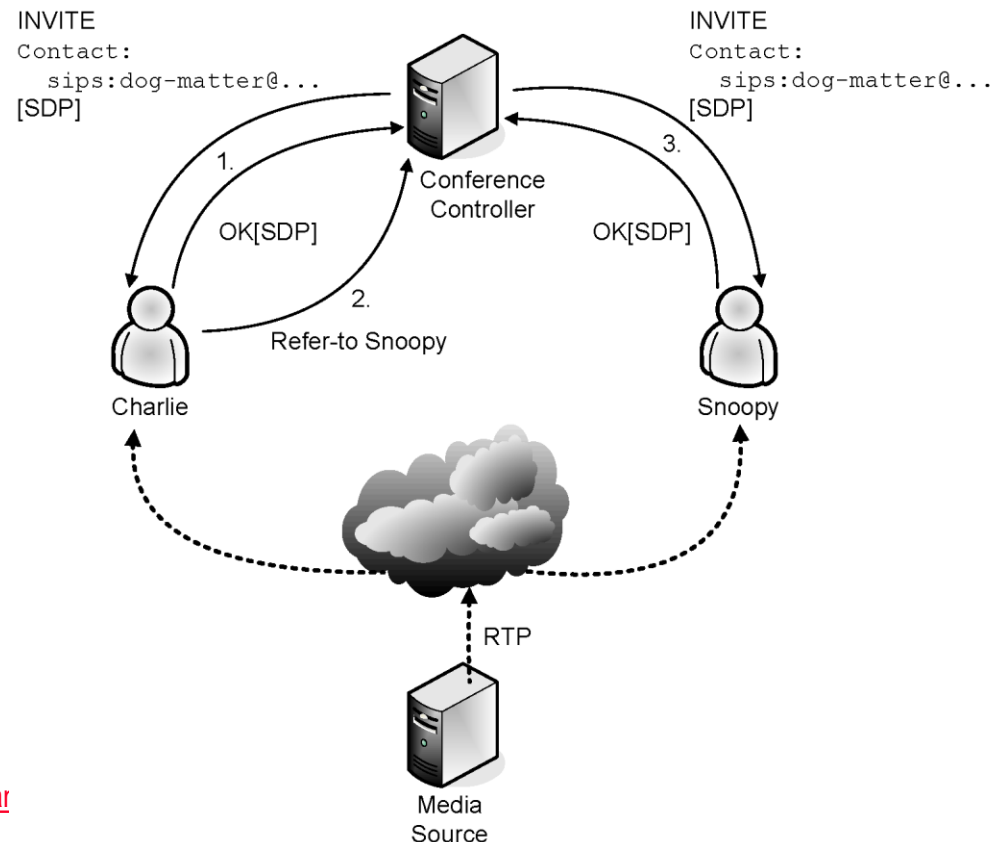
- o Typical Scenario: Two parties in a call extend conversation to a 3rd member (ad hoc)
- o Could be handled implicitly by application, but
  - No explicit group context (wiretapping!)
  - No way to switch relaying party

- o SIP introduces conference Focus in Contact header
  - explicit conference URI
  - `isfocus` tag

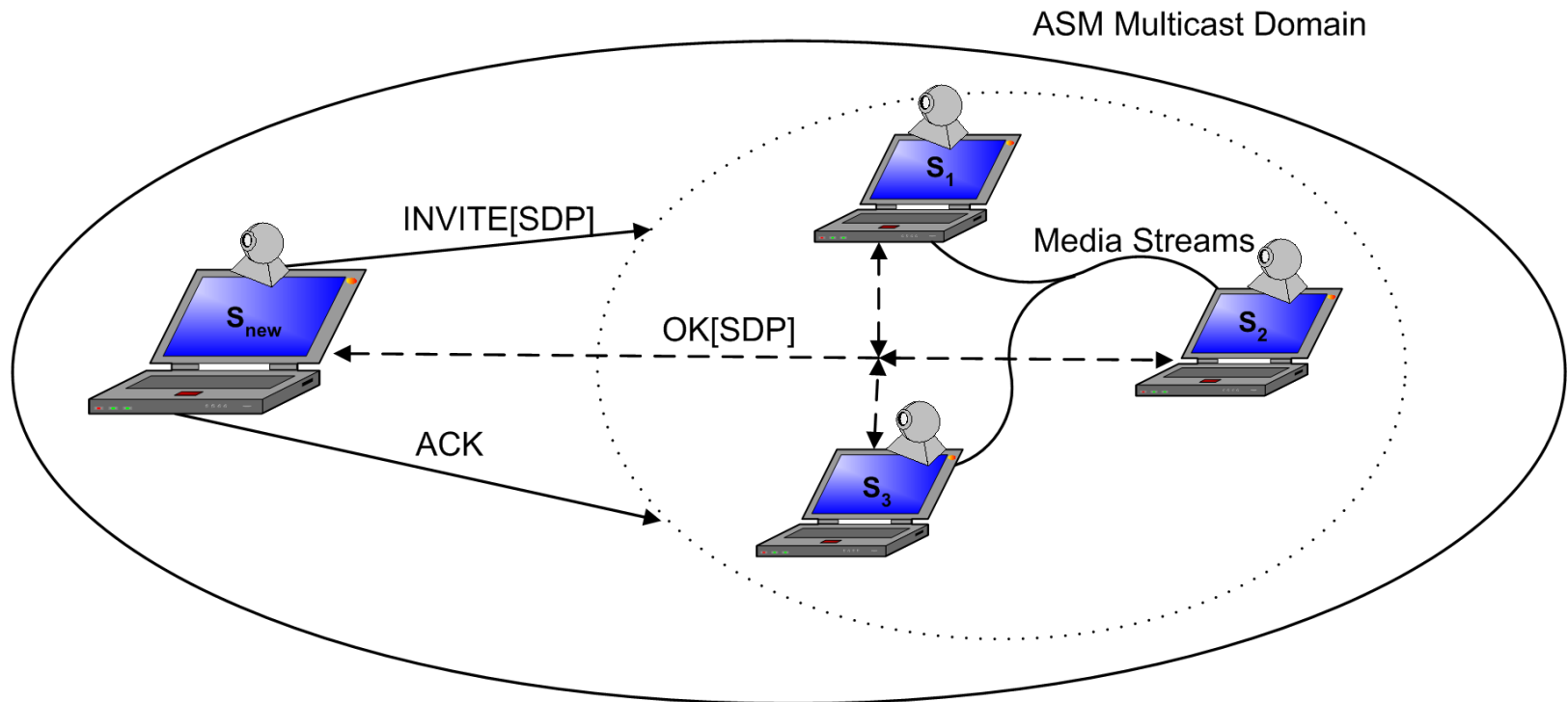


# Large Scale Conferences

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



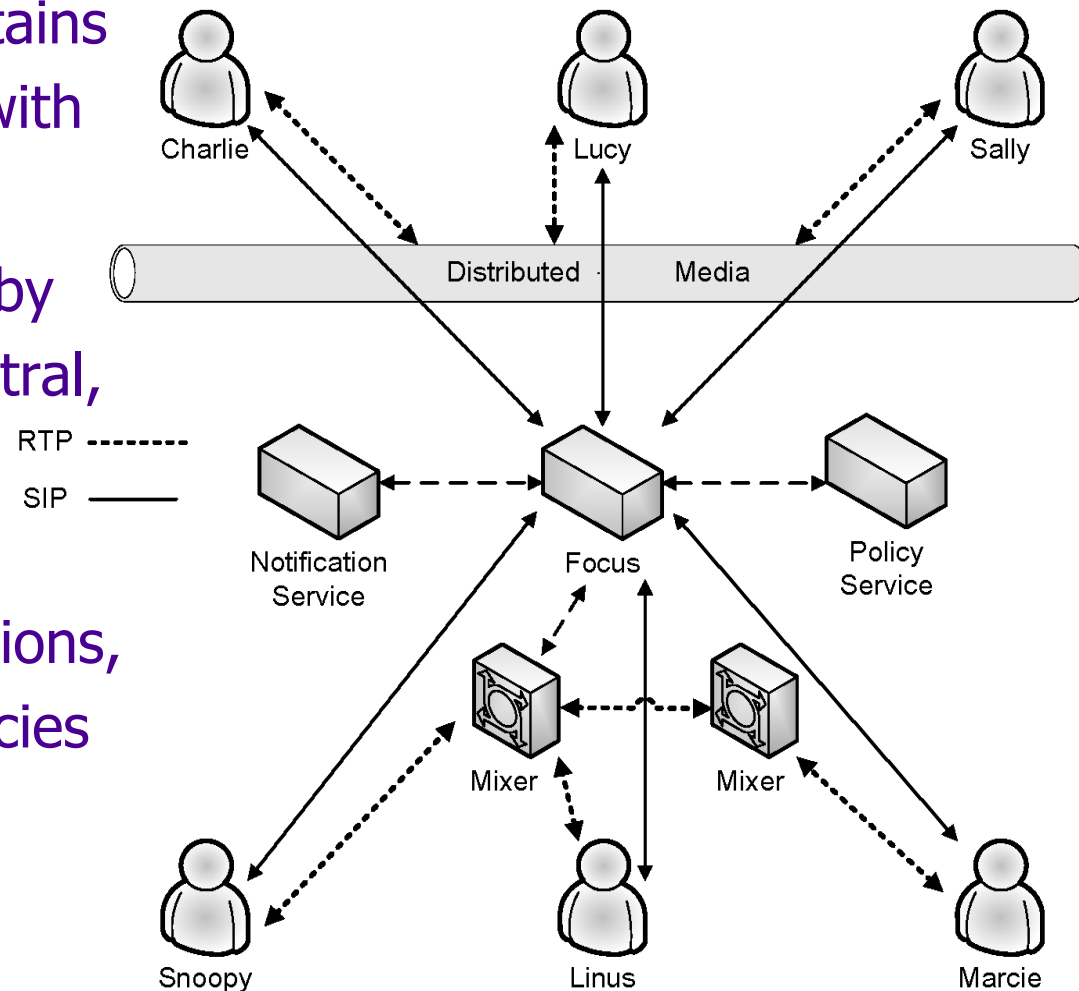
# SIP via Group Communication



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

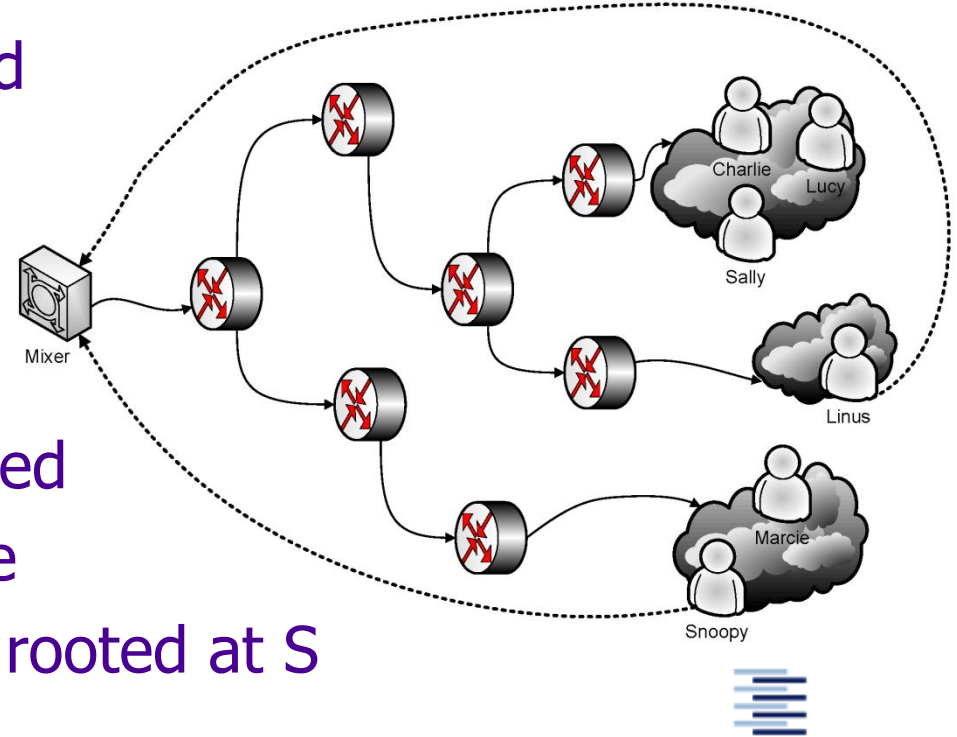
# Architecture of Tightly Coupled Conferences

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



# Media Distribution via SSM

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

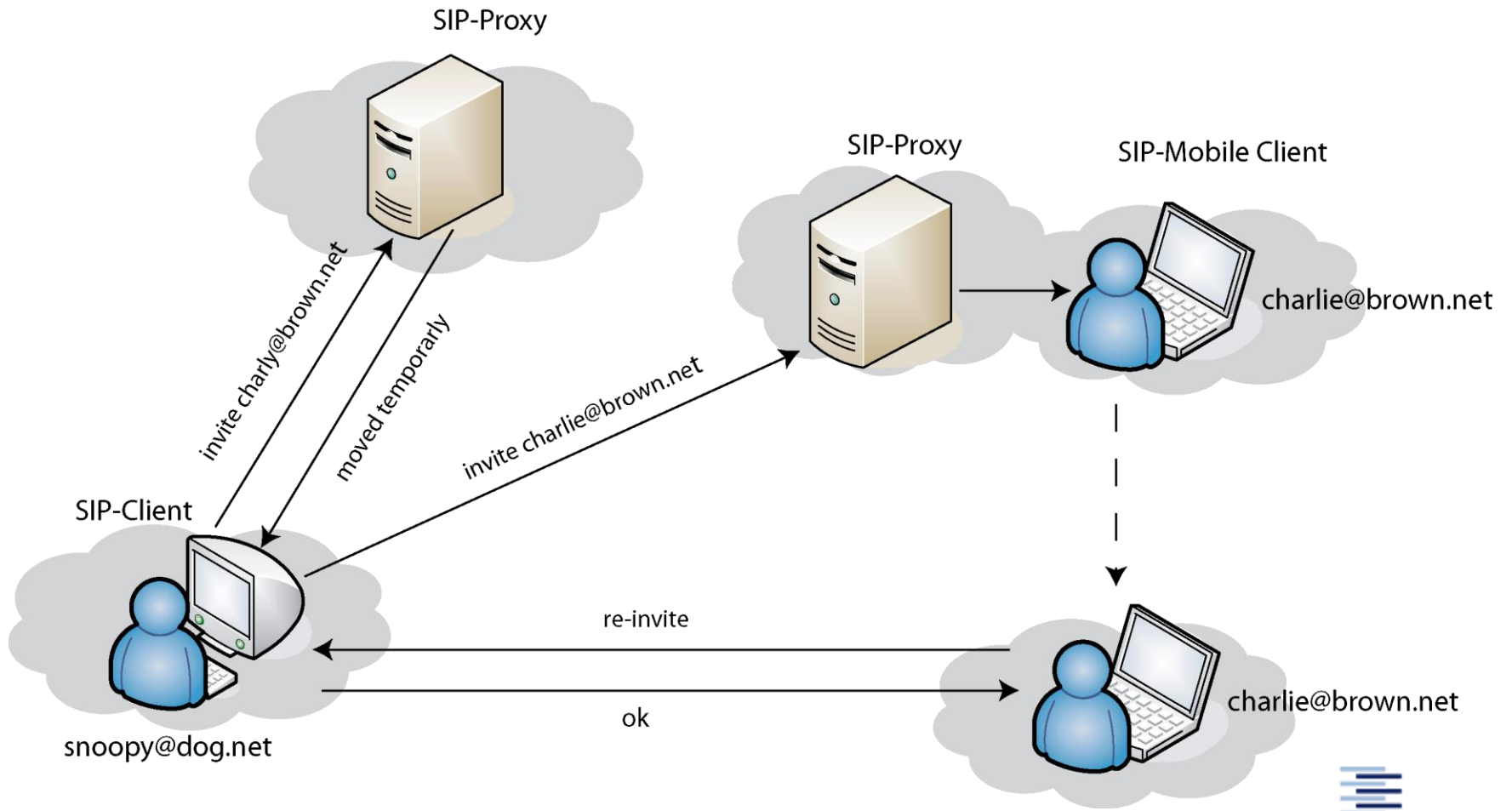


# Application Layer Mobility with SIP

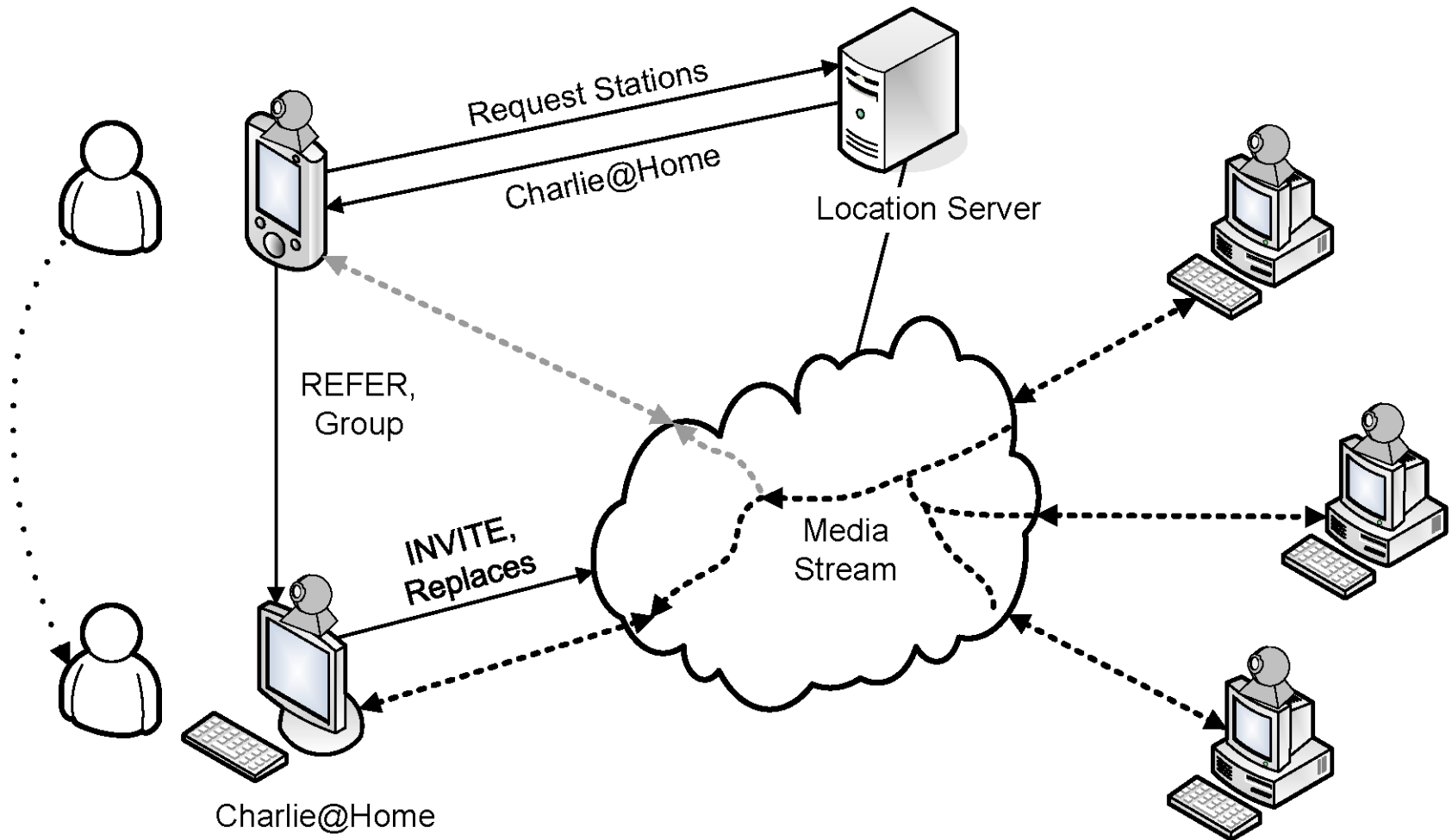
- o Two types: personal, session & midcall mobility
- o Personal mobility:
  - Multiple registration: with home and visited registrar
  - On call registrar returns "temporarily moved to"
- o Session mobility:
  - Releasing station issues a REFER to new conference instance
  - Accepting station uses re-INVITE with replaces to transfer call
- o Midcall mobility:
  - Mobile host issues a re-INVITE with its new session & contact data



# SIP User & Midcall Mobility

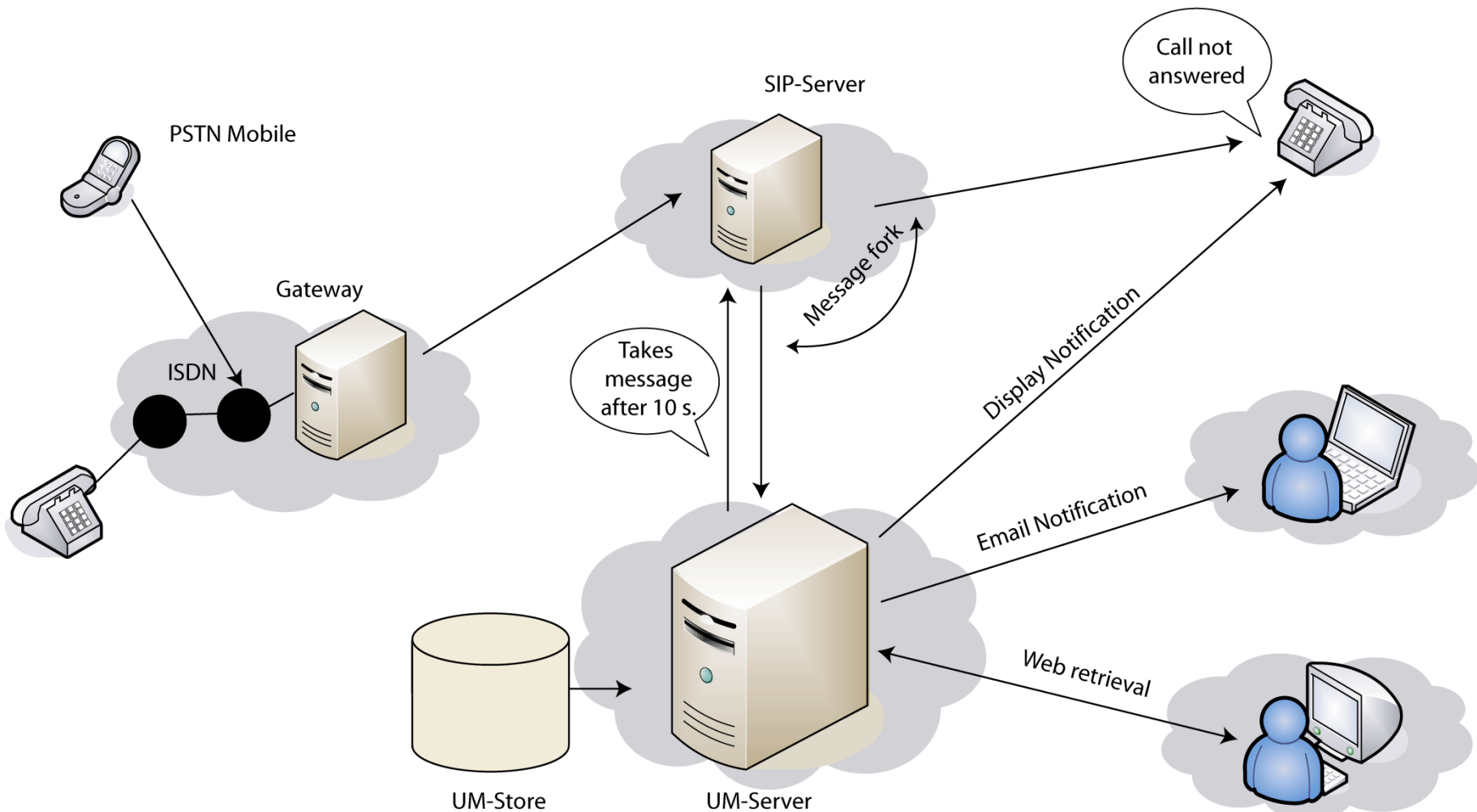


# SIP Session Mobility





# SIP Unified Messaging



# Directions in SIP Development

## o P2P SIP

- Initiated as a standard answer to Skype (2005)
- Establish a DHT infrastructure instead of proxies
- Use DHT for user and presence location and NAT-traversal assistance

## o Distributed Conferencing

- Split the central conferencing focus
- Sustain tight coupling (SDP negotiations) at a logical focus point



# Peer-to-Peer SIP

- o Combine SIP signaling capabilities with scalability of P2P Overlays:
  - SIP messaging over P2P networks
    - SIP inherently is a P2P protocol
  - Register SIP records as overlay resources
    - replace proxy servers
  - Service/user/node discovery and rendezvous via P2PSIP
    - find members without provider peering
  - Avoid central servers
    - enable fully distributed call management at peers

# Base Protocol: RELOAD

draft-ietf-p2psip-base

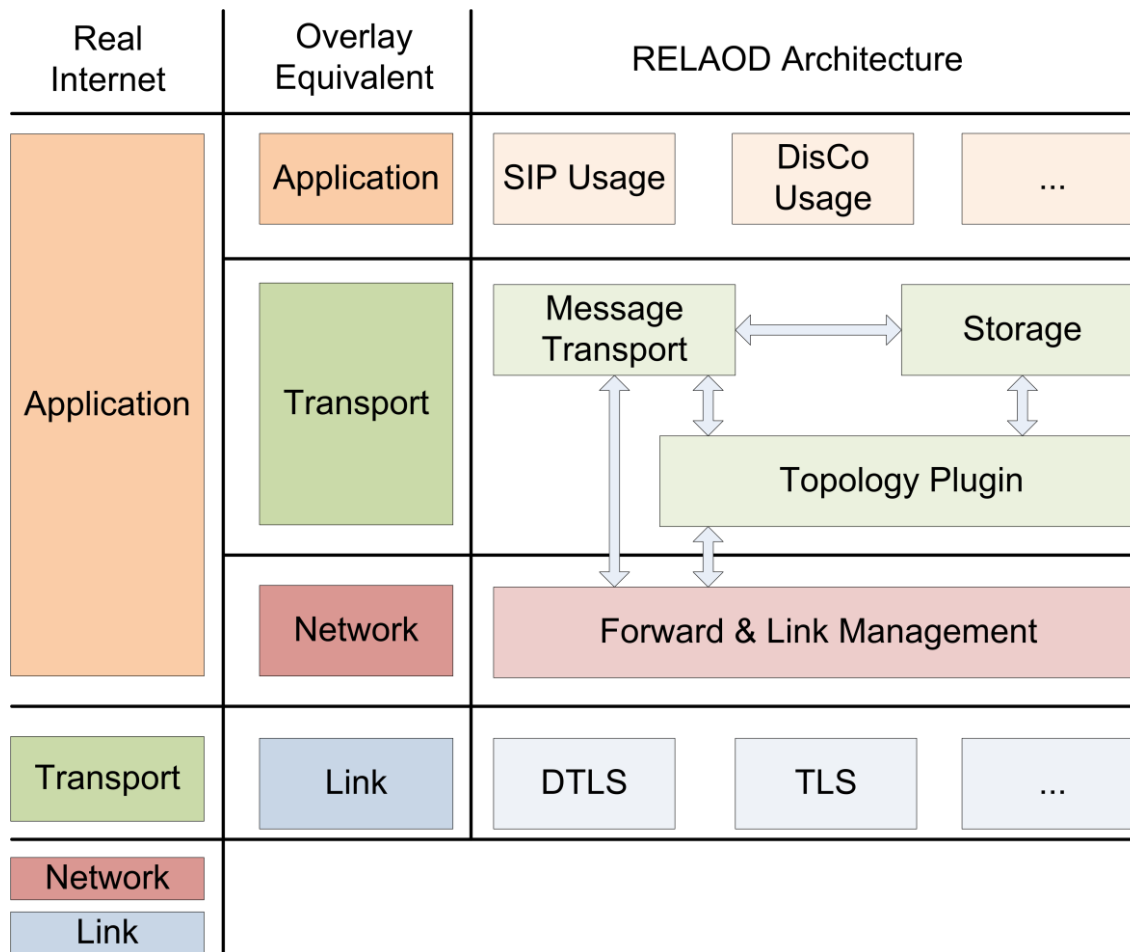
- o RELOAD = REsource LOcation And Discovery
- o Signaling protocol for P2P networks
- o Features:
  - Data storage and retrieval in a P2P network
  - Authentication, messaging and routing services
  - Meta data model to support variety of application
  - Pluggable overlay algorithm
  - Rigorous security definitions
  - NAT and Firewall traversal





# RELOAD Stack Architecture

## Pluggable and Extendable



### o **Application:**

- The application specific behaviors, called **Usages**

### o **Transport:**

- Defines messages (store, fetch, join, etc)
- Defines storage types and permissions
- Pluggable Overlay

### o **Network:**

- Creates, maintains and deletes connections

### o **Link:**

- Underlying secure transport protocols



# RELOAD Kinds

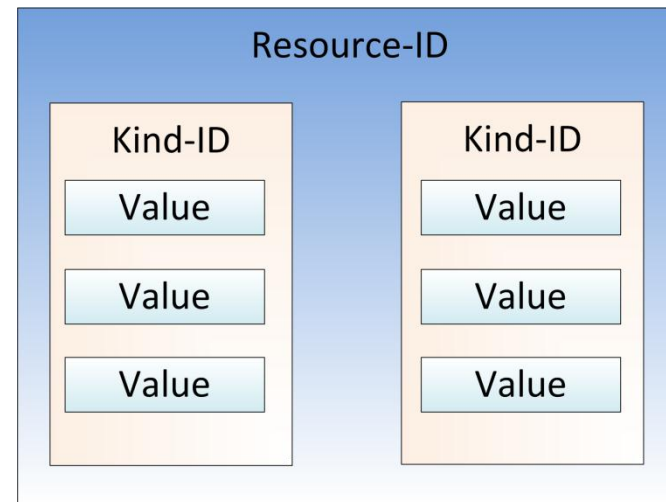
Meta-concept to support multiple applications

## o Overlay Resource IDs containing several Kinds

- Kinds are application specific data structures
- Resource-ID = Hash(Resource name)
- Each application data is assigned a Kind-ID

## o Kinds can store:

- Single values
- Arrays
- Dictionaries

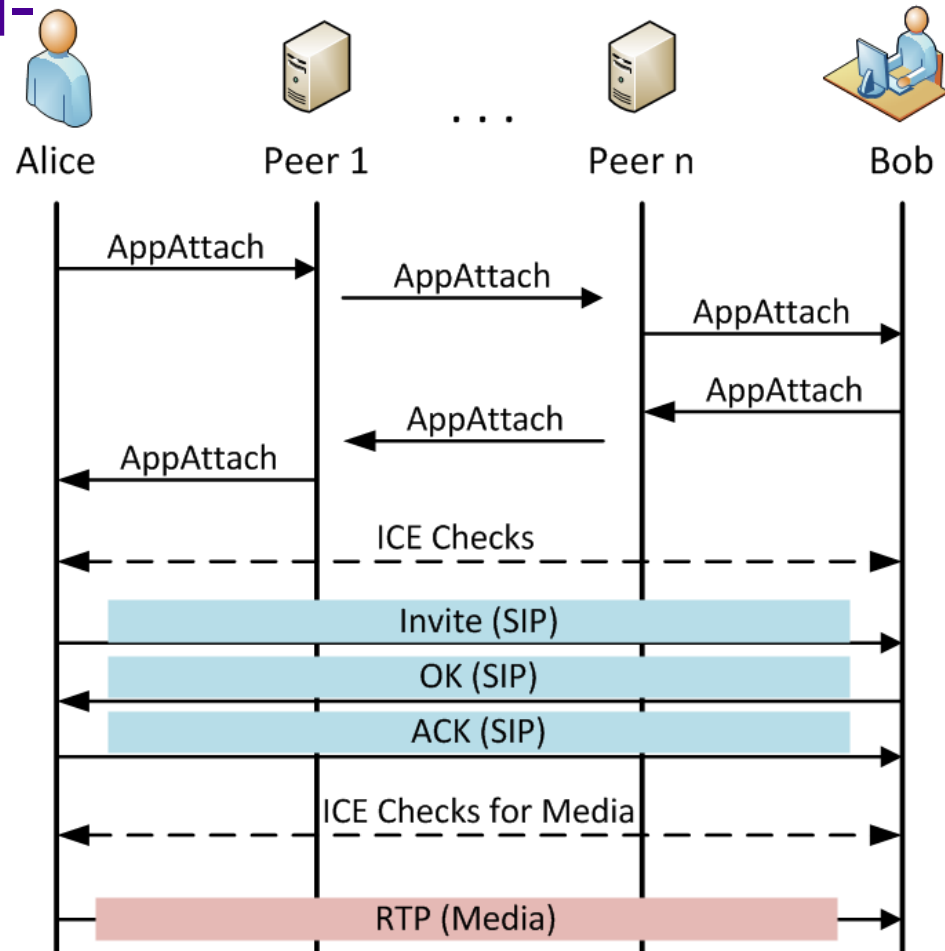


# SIP Usage for RELOAD

draft-ietf-p2psip-sip

Defines a Usage to distribute the SIP Proxy/Registrar function

- Storage of SIP URI + Route objects
- NAT traversal assistance (ICE)
- SIP signaling remains unaltered (end-to-end)





# 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>)

reSIPProcate (<http://www.resiprocate.org/>) – Minimal UAC example [here](#)



# Reading

- D.B. Johnston: **SIP, 3<sup>rd</sup> Ed.**, Artech House, 2009.
- Sinnreich, Johnston: **Internet Communications Using SIP, 2<sup>nd</sup> Ed.** Wiley & Sons, New York, 2006.
- Syed Ahson, Mohammad Ilyas (Ed.): **SIP Handbook: Services, Technologies, and Security**, CRC Press, Boca Raton, December 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](http://www.rfc-editor.org) .
- ITU Documents: [www.itu.int](http://www.itu.int) .

