Overview

- Motivations for Internet Conferencing
- Vertical Communications
- Horizontal Communications

Conferencing and Computer-Supported Collaborative Work

Characteristics:

synchronous: simultaneous interaction; “video conferencing”

asynchronous: time-shifted: email, news groups, web, shared document editing, workflow computing (edit, sign, process), . . .

➠ may want to use both

Can integrate audio and video into the Internet or use separate network (ISDN, plain old telephony services (POTS)).

Why bother integrating?

- use existing workstations as audio/video terminals
- use existing LAN/WAN infrastructure
- efficiency:
  - true (LAN/WAN) multicast instead of MCUs and bridges – same application scales from two to hundreds of receivers
  - variable-bit rate (VBR) video, but interoperation with H.261 standards
  - audio silence suppression – important for large-scale conferences

Why bother integrating (cont'd)?

- added control functionality:
  - directory services
  - visual speaker indication
  - source selection at receiver
- integration of application-sharing and data applications, WWW

Switching costs

switching method	capacity (Gb/s)	cents/kb/s
10BaseT Ethernet hub	0.12	0.5
10BaseT Ethernet switch	0.24	2.0
100BaseTX Ethernet switch	0.80	1.0
router	2.1	16.0
local ATM switch	2.48	1.6
PBX (256 lines)	0.02	218.
AT&T 4ESS local (107k lines), 3-4 CCS	6.85	383.
AT&T 4ESS local, 30 CCS	6.85	3750.
AT&T 5ESS toll (100k lines)	6.40	7.8

Major cost is cabling + interface (both telephone and optical), not switching ➠ single infrastructure
CCS = hundred call seconds/hour ➠ 100% $\equiv$ 36

Disadvantages of packet audio/video

- no resource reservation + insufficient bandwidth ➠ poor quality
- may push overloaded networks over the edge
- low frame rates for workstation video codecs
- hands-free speaking in infancy (echo)
- packetization overhead, delay
- operating systems ill-suited for real-time applications ➠ single-user, no background load
- office environment: acoustics, lighting, . . .

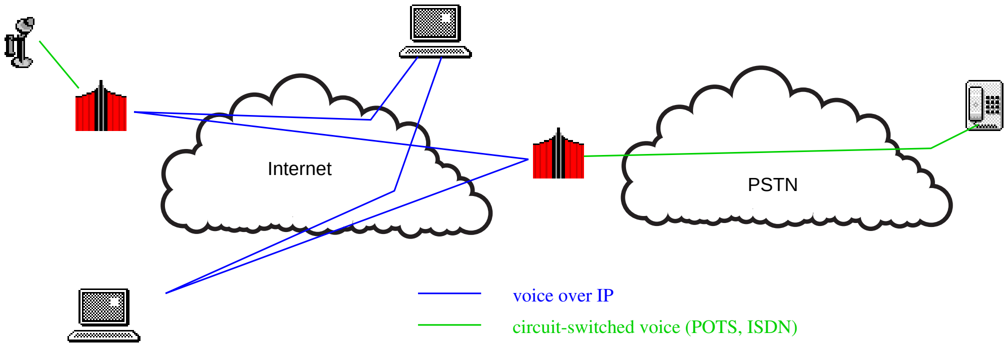
Integration of real-time and data traffic

- TCP not suited to carry real-time traffic:
  - flow control ➠ window backoff and slow start
  - retransmission delay
- TCP backoff nice for making room for real-time traffic
- options:
  - low load, rely on data traffic backoff (but NFS doesn't)
  - priorities at MAC/link level (starvation!)
  - bandwidth-allocating scheduling at link level
- transport protocols can't help (except with set-up)!

Internet telephony modes

Connectivity:

- packets end-to-end: Mbone, Vocaltec, . . . ➠ requires continuous Internet connectivity
- tail-end hop off: use PSTN from POP



Bundling:

“Retail”: individual packets per connection

“Wholesale” (trunking): interconnect PBXs or gateways

MBone Conferencing

- traditional (POTS, ISDN, ATM) conferences:
  - central server tracks participants
  - multipoint control unit replicates data
- minor disadvantages:
  - doesn't scale
  - complicated when dealing with failures
  - single point of failure; network partitions!
- usually don't need to know *immediately* when somebody enters or leaves the room (there are exceptions…)
- ➠ ideal for soft-state and IP multicast
- multicast convenient for large conferences: no need to inform all others or central server

- multicast ➠ anybody within ttl radius can get data ➠ need *encryption* for privacy
- but: unless you trust every provider, need it anyway
- −: multicast + encryption ➠ can still easily get some information on participants
- conference is visible to network ➠ re-use network facilities, avoid strange failure modes
- conferencing tools are much simpler

➠ light-weight session model
mostly used for seminar-style conferences, but can also be extended to small, interactive groups

Components of Mbone conferencing

Well-developed tools:

- session directory: sdr
  - "TV guide"
  - allocates addresses from 16K space, with dynamically adjusted ranges for each scope
  - shows conflicts
  - allows invitation to conferences
  - currently flat $\rightarrow$ directory of directories
- audio tool: vat, NeVoT
- video tool: vic, ivs, NeViT
- whiteboard for PostScript slides, with annotations

Not yet fully developed:

- equivalent tools for non-Unix systems
- integration of synchronous and asynchronous collaboration
- voting
- floor control: who gets to talk next
- “talk radio” support (“talent” vs. phone manager)
- application sharing ➠ X sharing

Conferencing architectures

- traditional: single application
- Internet Mbone tools: individual, standalone applications

What if Media Agents Could Talk to Each Other?

- "start-and-forget" ➠ continuous involvement
- audio follows video: stereo placement according to window location
- video follows audio: enlarge image of speaker
- audio enables video: send at higher rate when talking
- auxiliary applications: recorders, talk timers, . . .
- floor controller controls audio, video
- SNMP agent retrieves statistics, controls media agent

➠ must be easy to add, without access to source code
➠ independent of conference control mechanism

Vertical Architecture

control flow in multimedia conferences:

horizontal: conference control (CC), between participants

vertical: local, one participant, between applications

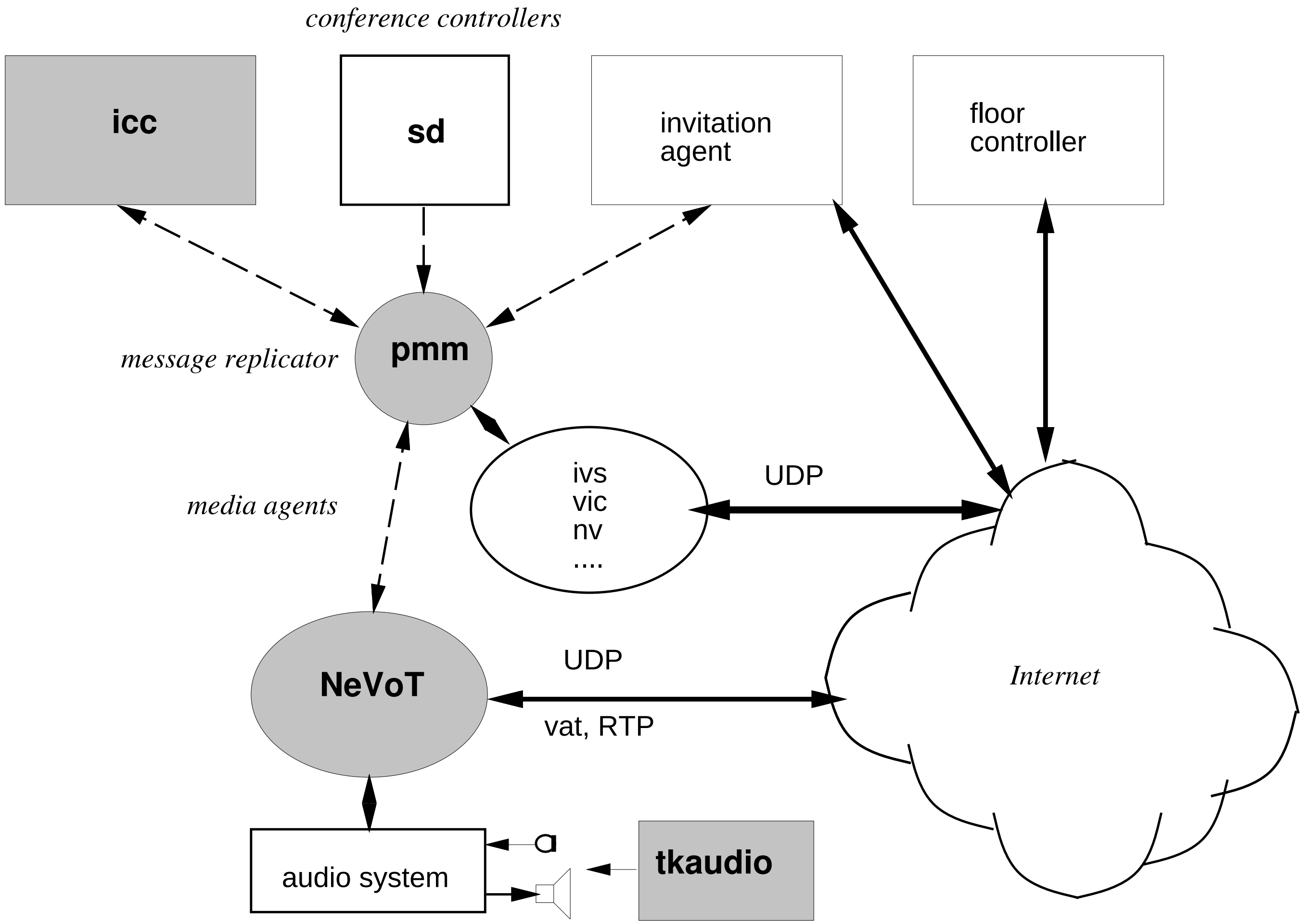
components:

controller: conference controller, floor controller, . . . : $\gg 1$!

media agents: audio, video, whiteboard, . . .

➠ reuse same media agents $\leftrightarrow$ different CC protocols, styles

Architecture



Mechanism

- largely independent of lower layers:

  replicator process via TCP: pattern-matching filter,

  local UDP multicast: per-conference, per-user, per-media?

- media agents/controllers declare interest in receiving particular messages
- sender does not know receiver identity
- can be (carefully) extended beyond single host

related: vic/vat notifications

Protocol

- commands are idempotent
- currently, text-based: *object*, *operation*, *parameters*
- directly Tcl-interpretable (Tk `send` philosophy)
- works equally well for audio, video, whiteboard, ...
- uses RTP/RTCP terminology, but not dependent on it

Example

```
M:   + session */audio/*
C:   + C/audio/3
C:   + C/audio/3/*
C:   create C/audio/3
M:   C/audio/3 created
C:   C/audio/3 ttl 128
C:   C/audio/3 media PCMU 1 8000 0.02
C:   C/audio/3 ...
C:   C/audio/3 open 224.2.0.1
M:   C/audio/3 opened
M:   C/audio/3 hello hgs@fokus.gmd.de
F:   C/audio/3/hgs@fokus.gmd.de send 1
M:   C/audio/3/hgs@fokus.gmd.de sending 1
M:   C/audio/3/hgs@fokus.gmd.de active
M:   C/audio/3/hgs@fokus.gmd.de !alive
M:   C/audio/3/hgs@fokus.gmd.de close
C:   C/audio/3 close
M:   C/audio/3 closed
M:   — C/audio/3
```

Bootstrapping Applications

Must handle

① single application per media session (video)

② several media sessions per application (nevot)

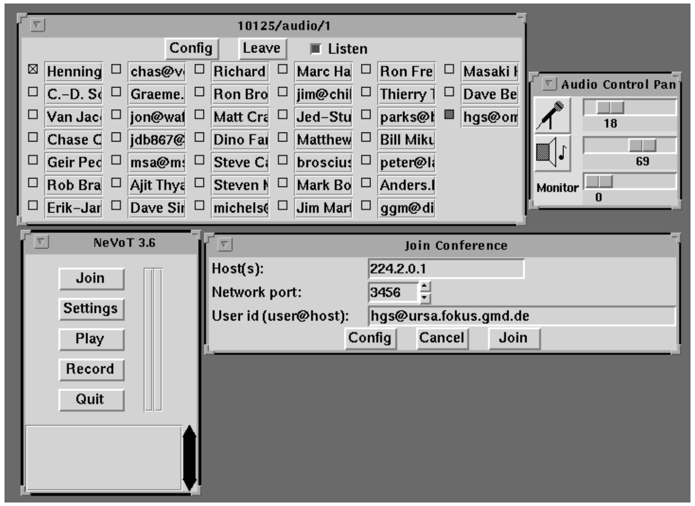
③ several media per application (audio/video)

replicator knows if `*/audio/*` registered; if not, create

multicast: send `create`; wait for reply; if none, start

Implementation

- NEVOT MBONE audio terminal
- controllers: interface to `sd`, telephone-style CC
- example: active badge: who's in the room?
- extended conference controllers: audio stereo location



Session invitation protocol (SIP): goals

- don't need to know IP address of user's host
- many dial-up users do not have a permanent IP address
- invite to new and on-going conferences
- allow for manual and automatic forwarding
- personal mobility (complements data link/IP mobility)
- reach first (load distribution) or reach all (department conference)
- possibly change permissible encodings during conference
- vehicle for conference control

Addressing users: UPT

UPT = universal personal telecommunications

- (mostly) dumb end systems
- permanent telephone *number*
- mapped to current landline number or mobile number
- separation: service provision $\leftrightarrow$ mapping ("number portability")?
- year 2010?

Addressing users: Internet approach

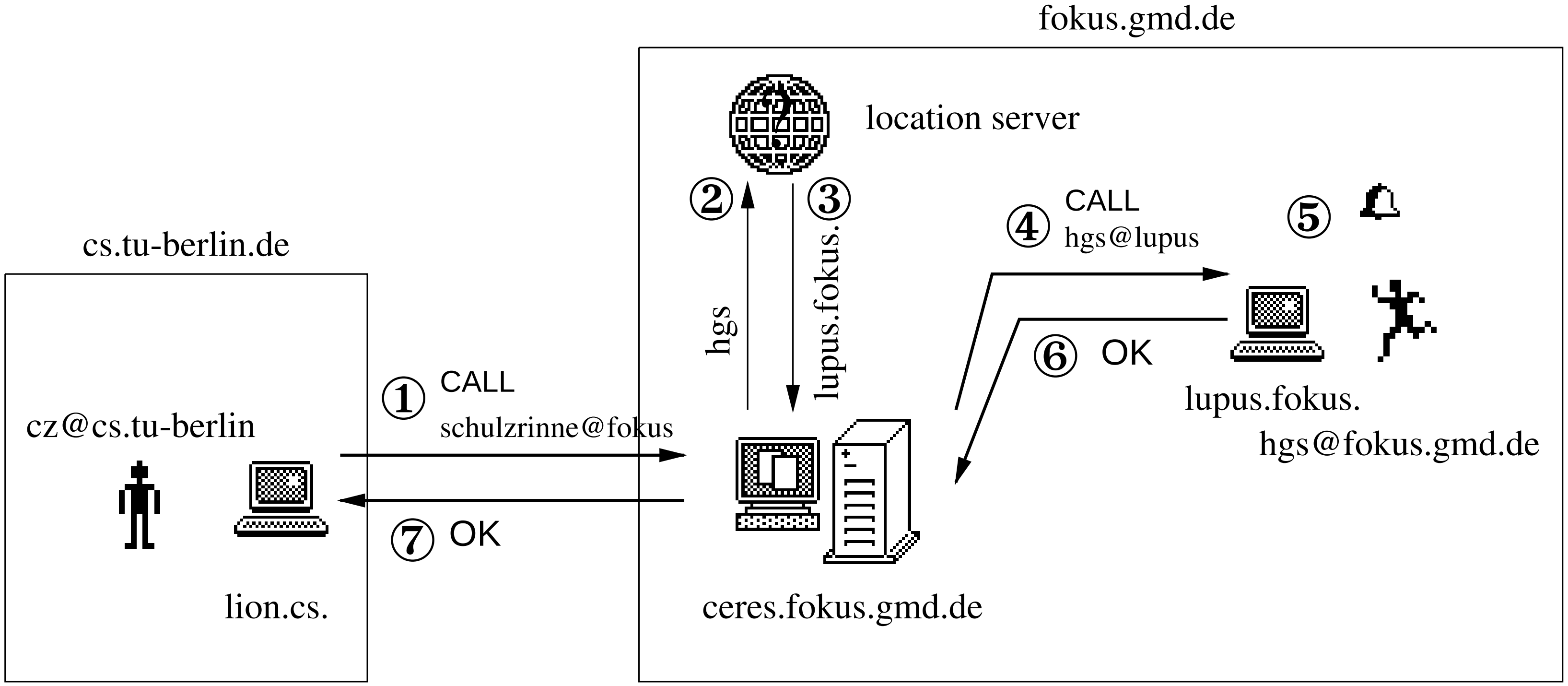
- use electronic mail address ➠ easier to remember
- name provider possibly $\neq$ pipe provider
- friendly naming: `john.doe`, `j.doe`, ...
- DNS MX records for organization addresses, reliability
- email aliases ("life-time phone number": `foo@ieee.org`)
- mailing lists
- email directories (X.500, whois++, ...)
- cannot use email $\leftrightarrow$ real-time signaling

Basic Operation

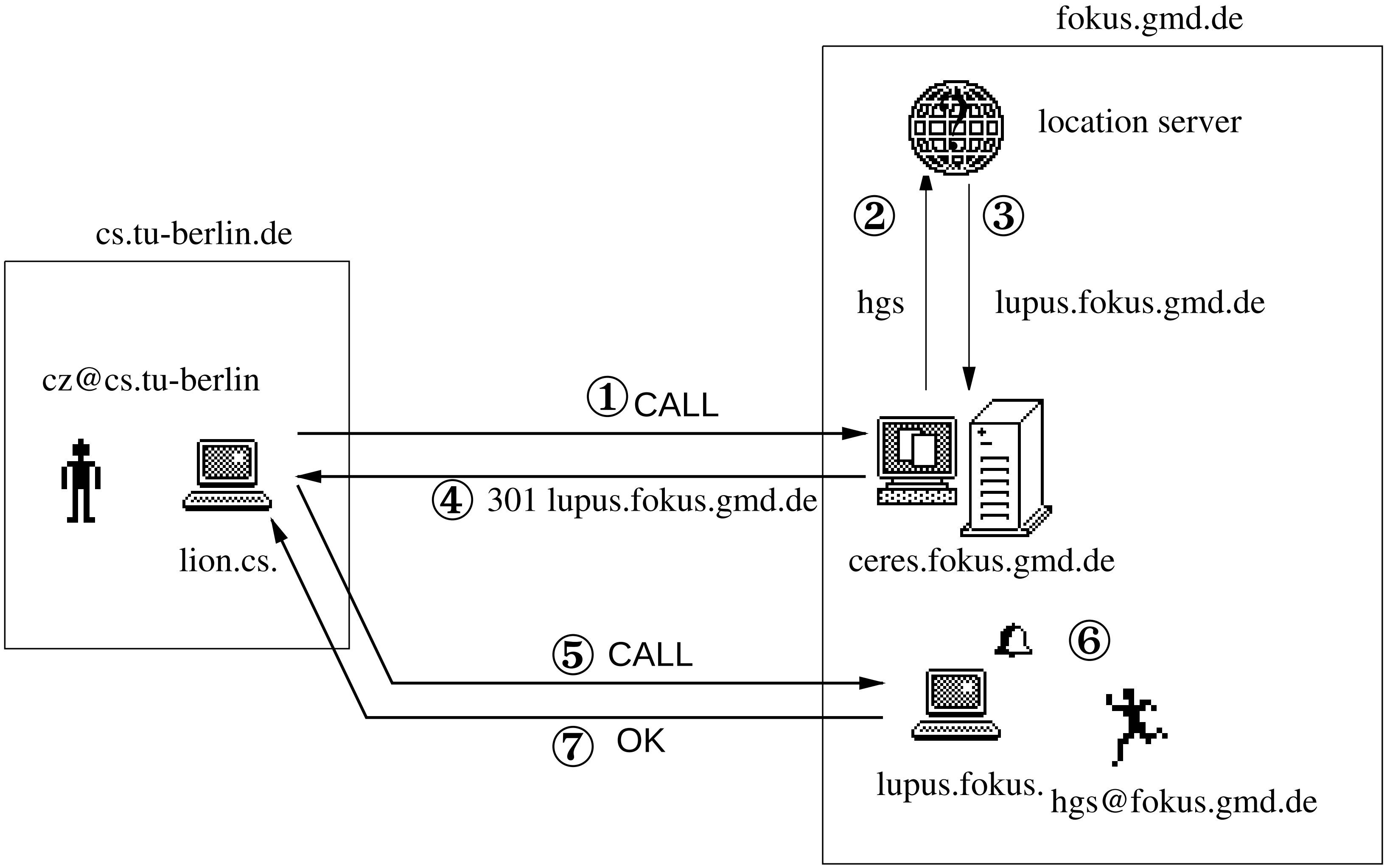
Example: call `schulzrinne@fokus.gmd.de`

- caller resolves `fokus.gmd.de` to actual host
- send SIP request to server
- server either . . .
  1. if workstation: accepts/rejects/forwards call, maybe after asking user (invisible to caller)
  2. possibly bridges to telephone
  3. locates user ➠ forwards call if not local ➠ redirect mode ➠ client tries again
  4. acts as client ➠ proxy mode
- user location is a local matter (but: could use SIP to register)

Operation in proxy mode



Operation in redirect mode



Protocol

- similar to HTTP or SMTP: TCP-based, ASCII, client/server
- re-use HTTP authentication, encryption
- connection may stay open for change requests
- HTTP error codes, redirect

Protocol request

```
CALL hgs@lupus.fokus.gmd.de SIP/1.0
From: Christian Zahl <cz@cs.tu-berlin.de>
To: Henning Schulzrinne <schulzrinne@fokus.gmd.de>
Call-Id: 9510021900.AA07734@lion.cs.tu-berlin.de
Referer: ceres.fokus.gmd.de
Expires: Mon, 02 Oct 1995 18:44:11 GMT
Required: fc99cb08 audio/pcmu; port=3456; transport=RTP;
  rate=16000; channels=1; pt=97; net=224.2.0.1; ttl=128,
  audio/gsm; port=3456; transport=RTP; rate=8000; channels=1,
  audio/lpc; port=3456; transport=RTP; rate=8000; channels=1
Required: 83ae5290 video/h261;port=4134;transport=RTP;rate=128,
  video/nv;port=4136;transport=RTP;rate=250,
  application/x-wb;port=1236
Accept: 56af7e9c application/editor;port=3500
Phone: +1 413 555 1212
Email: Christian Zahl <cz@cs.tu-berlin.de>
Location: TU Berlin; tz=MET; loc=52 32 00 N 13 25 00 E
Reimburse: ecash
Priority: urgent
Reach: first
Subject: New error codes
```

Response

- accept or reject call
- redirect temporarily or permanently

Redirection ➠ indicate several possible locations

```
SIP/1.0 302 Callee has moved temporarily
Location: jones@salt.lab3.company.com
Location: jones@pepper.lab3.company.com
Location: +49.30.254.99.182
```

RTSP

- Media Servers
  - VOD servers (pre-recorded)
  - Live Feeds (concerts, TV, etc)
- Desirable to control media servers
  - Content descriptions
  - Start, stop, record, pause
- RTSP ➠ HTTP like control protocol
- media stream and sessions defined by a RTSP URL

```
rtsp://media.example.com:554/twister/autrack
```

- Can reuse many components in http ➠ less coding, coreside RTSP servers with http servers

RTSP

- Many methods defined:
  - GET: get a session description from a server `C->S: GET rtsp://server.example.com/fizzle/foo RTSP/1.0 312 Accept: application/sdp, application/sdf, application/mheg Bandwidth: 4000`
  - Can also use http to get session descriptions
  - SETUP: Inform server of delivery mechanisms `C->S: SETUP foo/bar/baz.rm RTSP/1.0 302 Transport: rtp/udp;port=458`
  - PLAY: start sending data `C->S: PLAY rtsp://audio.example.com/twister.en RTSP/1.0 833 Range: smpte=0:10:20-`
- Can invite a server to existing conferences, new conferences
- Server can also be asked to record

H.323

- ITU standard for conferencing on Internet
- Video Codecs: H.261 or H.263
- Audio Codecs: G.711, G.722, G.723.1, G.728, G.729
- H.245 protocol used for content negotiation, opening media channels
- Q.931 for initial call setup
- H.225.0 for call control
- Gatekeepers
  - Call admission
  - Telephone number to IP address translation
- Multipoint Conferencing
  - centralized, network multicast, or hybrid

  - MCU used for mixing media streams (MCU = MP + MC)
  - MC always needed for content negotiation
- Gateways provide translation of media, signaling to H.324, H.320
- Signalling is complex, many round trips until actual data sent
- Parallel functionality to SIP,sdp/sdf
- Growing popularity among vendors