SIP for Internet Telephony and Conferencing

Henning Schulzrinne
Dept. of Computer Science
Columbia University
New York, New York
schulzrinne@cs.columbia.edu

VON April 1, 1998 – San Jose, CA

Overview

- Panelists:
  - François Menard, Mediatrix
  - Henry Sinnreich, MCI
- “tutorial”
- implementations and issues
- future

Tutorial Overview

- basic protocol
- examples of advanced services
- security
- interoperation with SS7
- mobility

Light-weight signaling: Session Initiation Protocol (SIP)

IETF MMUSIC working group

- unified view: Internet telephony $\longrightarrow$ large-scale conferences
- actually: general notification and awareness protocol
- all 5ESS (non-OAM/billing) services
- conferences: combinations of mesh, MCU, multicast, with transitions
- HTTP-like, but transport-protocol neutral (UDP, TCP, IPX, AAL5, $\ldots$)
- third-party signaling
- gateway-gateway signaling
- mobile telephony
- buddy lists

Architectural comparison

SIP	H.323
textual (HTTP)	Q.931, ASN.1
any media description	H.245
UDP or TCP	TCP
unicast or multicast signaling	unicast only
call $\neq$ connection	call = signaling connection
proxy, redirect (transparent)	gateway routed, RAS (diff. protocols)
self-describing	–
re-invite (change, add)	–

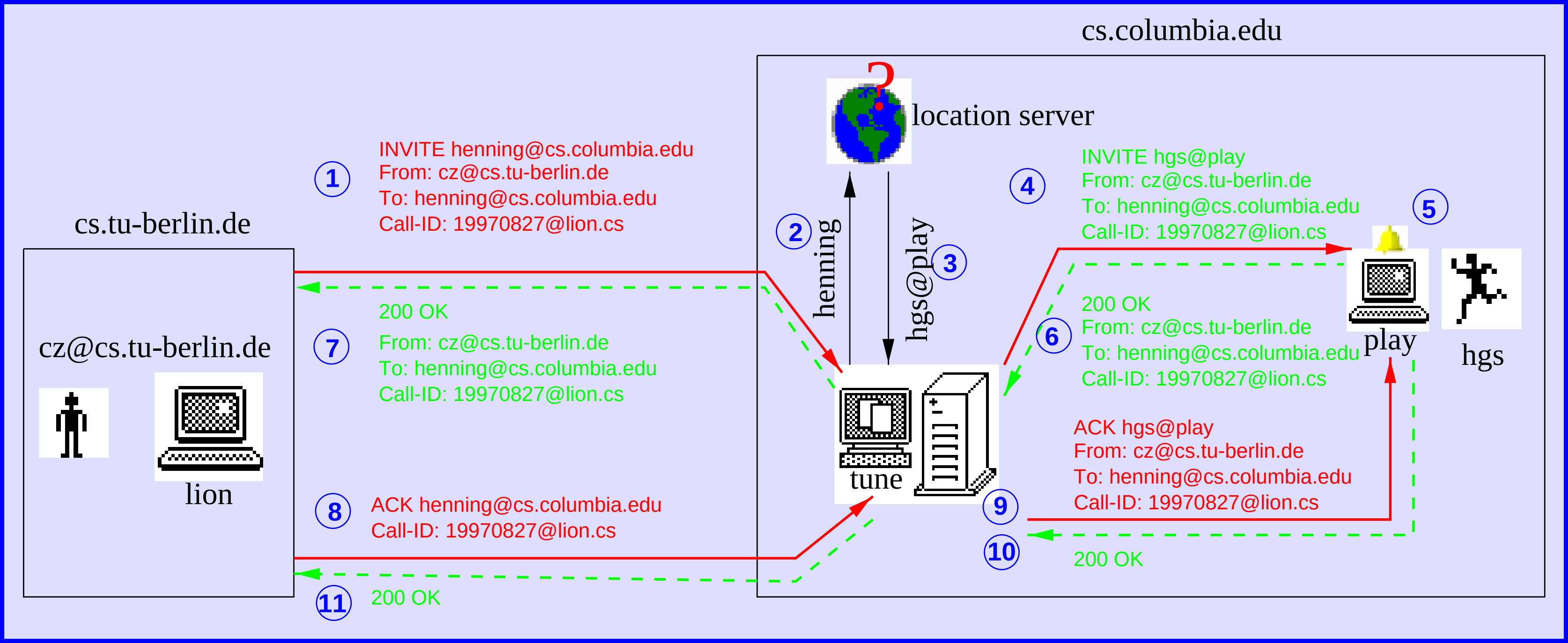
SIP addresses food chain

yellow pages	“president of the United States”	
	↓ *WWW search engines*	
common names	“Bill Clinton, Whitehouse”	
	↓ *directory services*	
host-independent	president@whitehouse.gov	
	↙ *SIP*	↘ *SIP*
host-specific	sip://bubba@oval.eop.gov	sip://+1-202-456-1111@net2ph.com
	↓ *DNS*	
IP address	198.137.241.30	

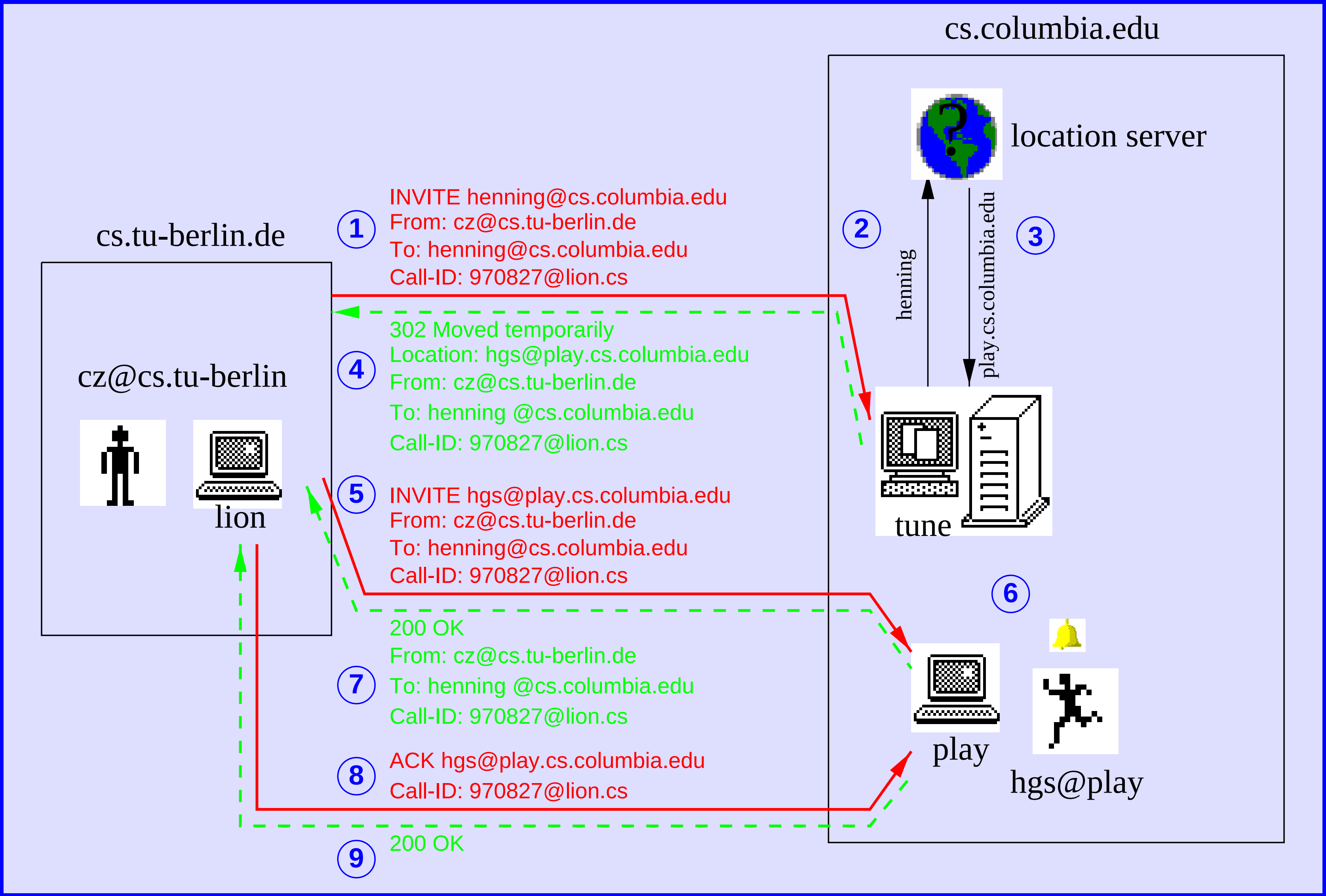
SIP: basic operation

1. use directory service (e.g., LDAP) to map name to *user@domain*
2. locate SIP servers
3. called server may map name to *user@host*
4. server accepts, rejects, forward ($\rightarrow$ new address)
5. if new address, go to step 2
6. if accept, caller confirms with ACK
7. . . . conversation . . .
8. caller or callee sends BYE

SIP operation in proxy mode



SIP operation in redirect mode



Protocol Request

```
INVITE schulzrinne@cs.columbia.edu SIP/2.0
From: Christian Zahl <cz@cs.tu-berlin.de>
To: Henning Schulzrinne <schulzrinne@cs.columbia.edu>
Via: SIP/2.0/UDP 131.215.131.131, SIP/2.0 foo.com
Call-ID: 199612061103.AA1528@cloud9.cs.tu-berlin.de
Content-Type: application/sdp
Content-Length: 187
Priority: urgent
Subject: New error codes

v=0
o=g.bell 877283459 877283519 IN IP4 132.151.1.19
c=IN IP4 132.151.1.19
b=CT:64
m=audio 3456
a=rtpmap:121 red/8000
m=video 5678
```

Response

- accept or reject call, manually or automatically
- redirect temporarily or permanently

Redirection ➠ indicate several possible locations

```
SIP/2.0 100 Trying to find user

SIP/2.0 180 Ringing

SIP/2.0 302 Callee has moved temporarily
Location: sip://jones@salt.lab3.company.com
Location: sip://jones@pepper.lab3.company.com
Location: phone://1.212.939.7042; class=business
```

SIP methods

INVITE	initiate call
ACK	confirm final response
BYE	terminate (and transfer) call
OPTIONS	features support by other side
REGISTER	register with location service

Invitation modes

invitation	conference	
	unicast	multicast
unicast	telephony	MBone session
multicast	reach first	dept. conference

➠ SIP for all modes

SIP reliability

- SIP: UDP and TCP, same messages, same behavior
- UDP: much lower call setup delay, particularly with packet loss
- UDP: multicast signaling
- retransmit after 500 ms, then double, unless RTT estimate
- …until first response

SIP reliability: INVITE

- server responds with *provisional* (100) or final ($\geq$ 200) response
- client confirms final response via ACK ➠
  1. $C \rightarrow S$: INVITE
  2. $S \rightarrow C$: 100, *user location*, *ringing*, ...
  3. $S \rightarrow C$: 200
  4. $C \rightarrow S$: ACK
- server repeats final response if no ACK
- Via header for loop prevention

Security

hop-by-hop: IPsec, SSL

proxy: Proxy-Authentication

end-to-end HTTP: basic (password) and digest (challenge-response)

end-to-end cryptographic: PGP, S/MIME – as filter

SIP extensions and feature negotiation

- receiver ignores headers, parameters it doesn't understand
- if crucial, mark with "Require: *feature*"

```
C->S:   INVITE sip:watson@bell-telephone.com SIP/2.0
        Require: com.example.billing
        Payment: sheep_skins, conch_shells

S->C:   SIP/2.0 420 Bad Extension
        Unsupported: com.example.billing
```

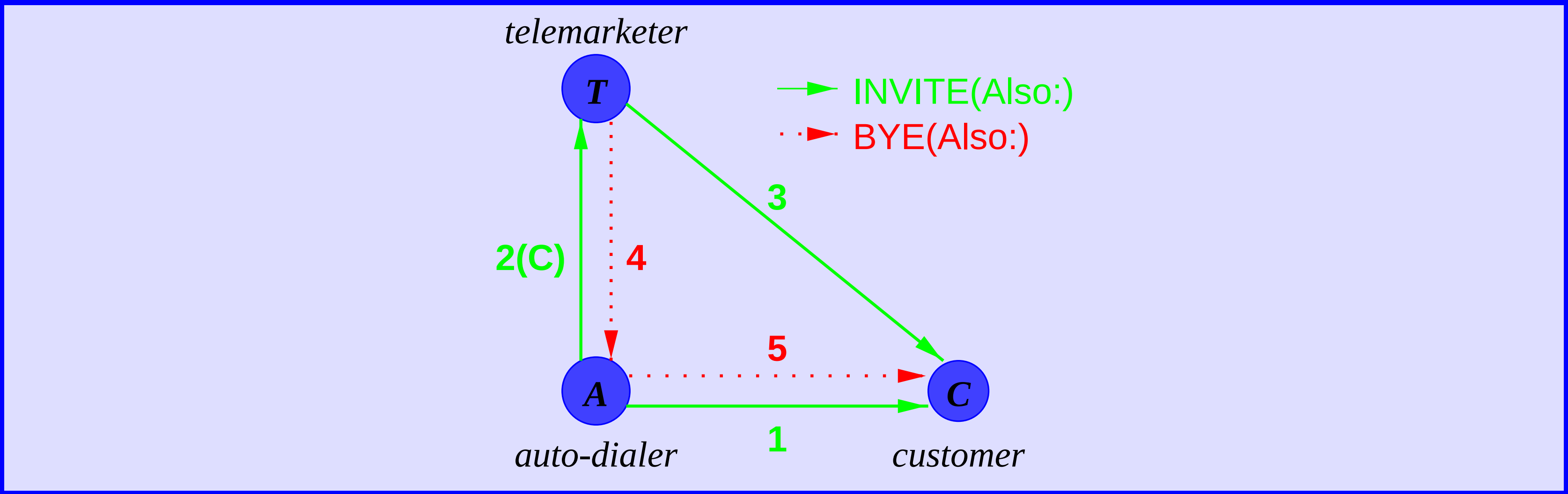
- methods: on failure (506), indicate via Allow
- inquire about capabilities: OPTIONS $\longrightarrow$ Public, possibly supported media types

Building advanced services

Construct from element *behavior*, not feature descriptions:

request URL	next resolution stage
From	logical call source
To	logical call destination
SDP “c=”	address media is to be sent to – Internet or PSTN!
Also	additional participants
Location	alternate participants
Replaces	replace existing call legs

Outgoing call handling: telemarketing

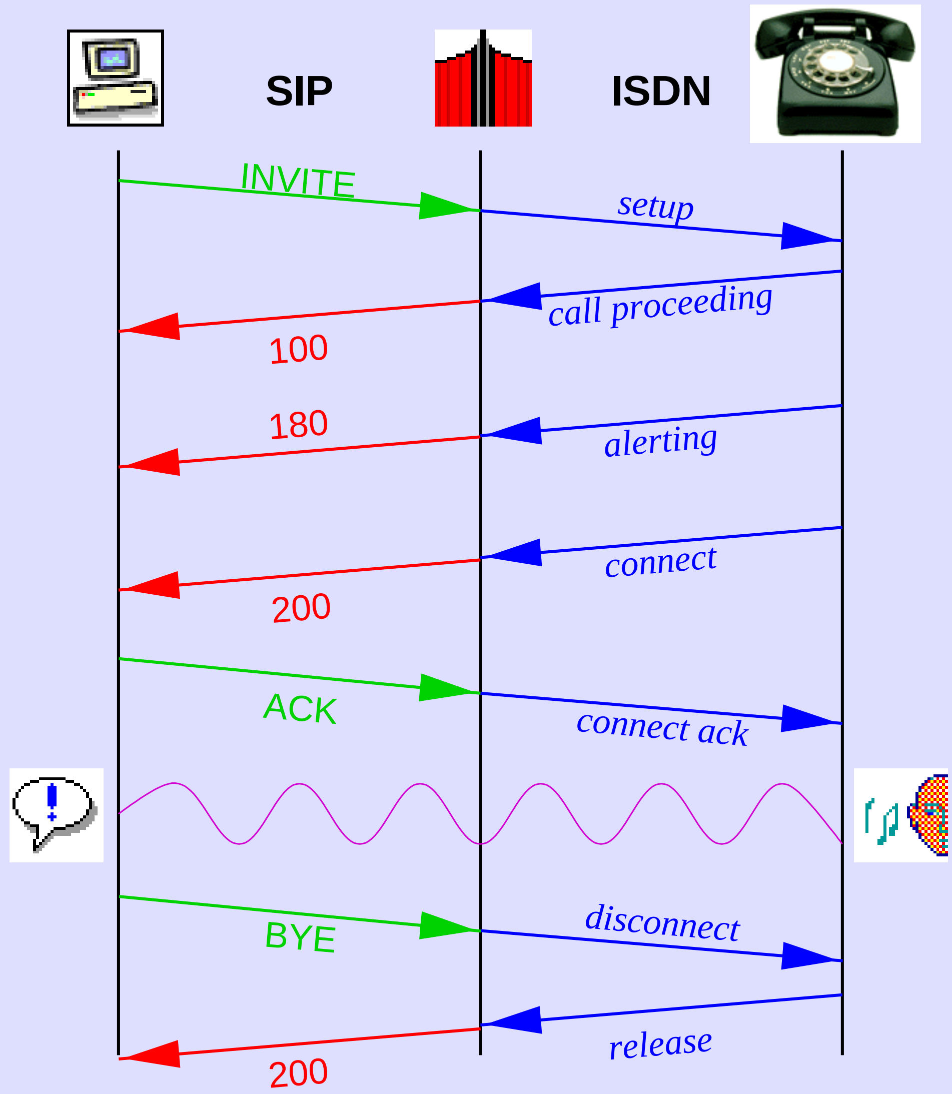


1. auto-dialer A INVITEs customer C
2. A sends INVITE with Also: C to next available telemarketer T
3. T INVITEs C for same call; C accepts automatically
4. T sends BYE to A
5. A sends BYE to C

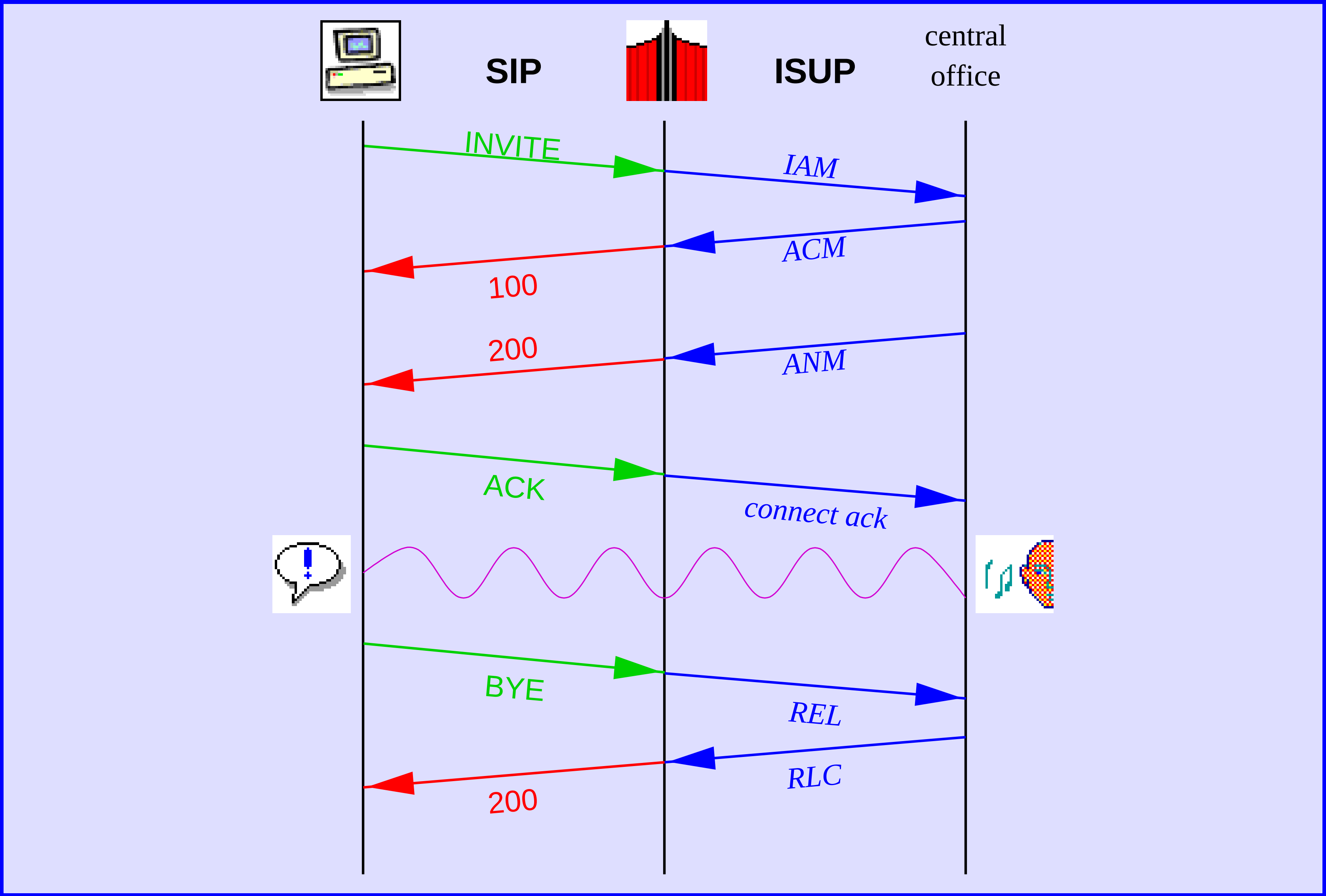
Multicast signaling

- send request to local group: “help@foo.com”, “everybody in family”
- first to pickup gets the call
- ➠ home phone model
- ➠ cheap ACD

SIP interaction with Q.931 (ISDN)



SIP interaction with SS7 (ISUP)



Buddy Lists

- let others know I'm around (and ready to communicate)
- predecessor: ILS – flat space, doesn't scale
- Alice wants to know when Bob wants to talk:
  1. Alice $\rightarrow$ Bob's home server: "HERE Bob"
  2. Bob REGISTERs ➠ Alice gets "200 OK"
  3. Alice $\rightarrow$ Bob's home server: "GONE Bob"
  4. Bob leaves ➠ Alice gets "200 OK"
- normal call processing/forwarding applies
- scaling mechanism via DNS

Mobility

- call forwarding ≈ mobility
- REGISTER with HLR and local VLR
- UDP ➠ IP address doesn't matter
- mobile IP not needed for voice applications ➠ use DHCP to get local addresses

Call Processing Language

- currently in the works ➠ IPtel working group
- not just an API (TAPI, TSAPI, . . .) ➠ don't reflect parameters, third-party signaling
- simple: handle incoming calls ➠ similar to email filtering
- advanced:
  - state-based ➠ reflects IVR, protocols
  - independently extensible (IVR!)
  - high-level (timeouts, errors, parallel searches, ask for password, . . .)

Summary

- SIP = single, simple, comprehensive, extensible protocol
- with H.323: user location, registration, buddy lists, gateway signaling, . . .
- libraries and applications emerging