Internet Telephony: Beyond the Black Telephone

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

Nokia, Boston

August 3, 1999

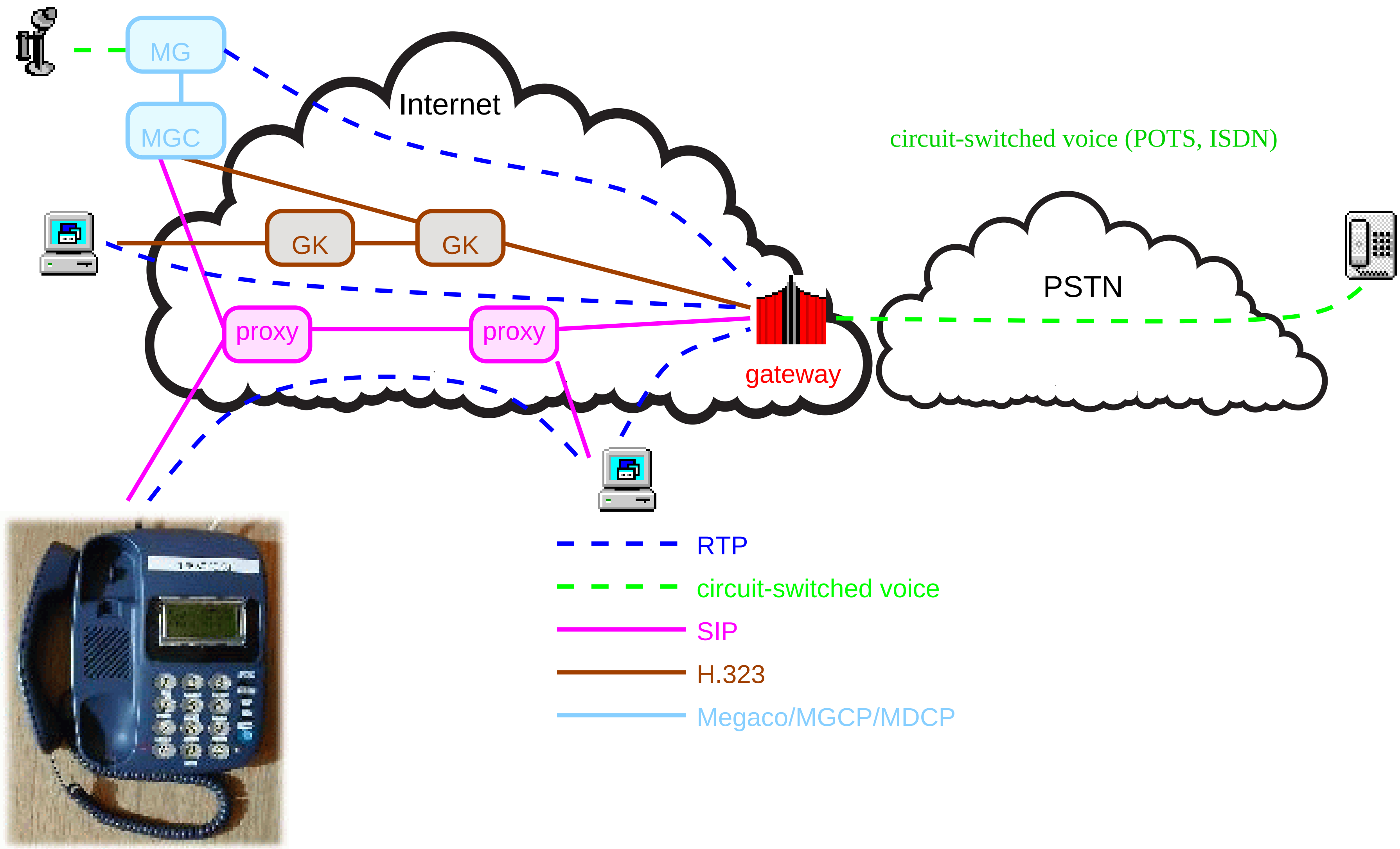
Overview

- Internet telephony issues
- extensions:
  - interaction with QOS
  - caller preferences
  - call control
- programming services
- mobility and wireless Internet telephony
- remote access

Internet telephony issues

“black-phone”	Internet-centric
master-slave	peer
primarily voice	multimedia, games, ...
interworking with PSTN	PSTN = end system
CLASS, CS1 services	programmable services
phone numbers	single email-like addresses
GSM, UMTS	SIP-based mobility
standalone	integration with IM, presence

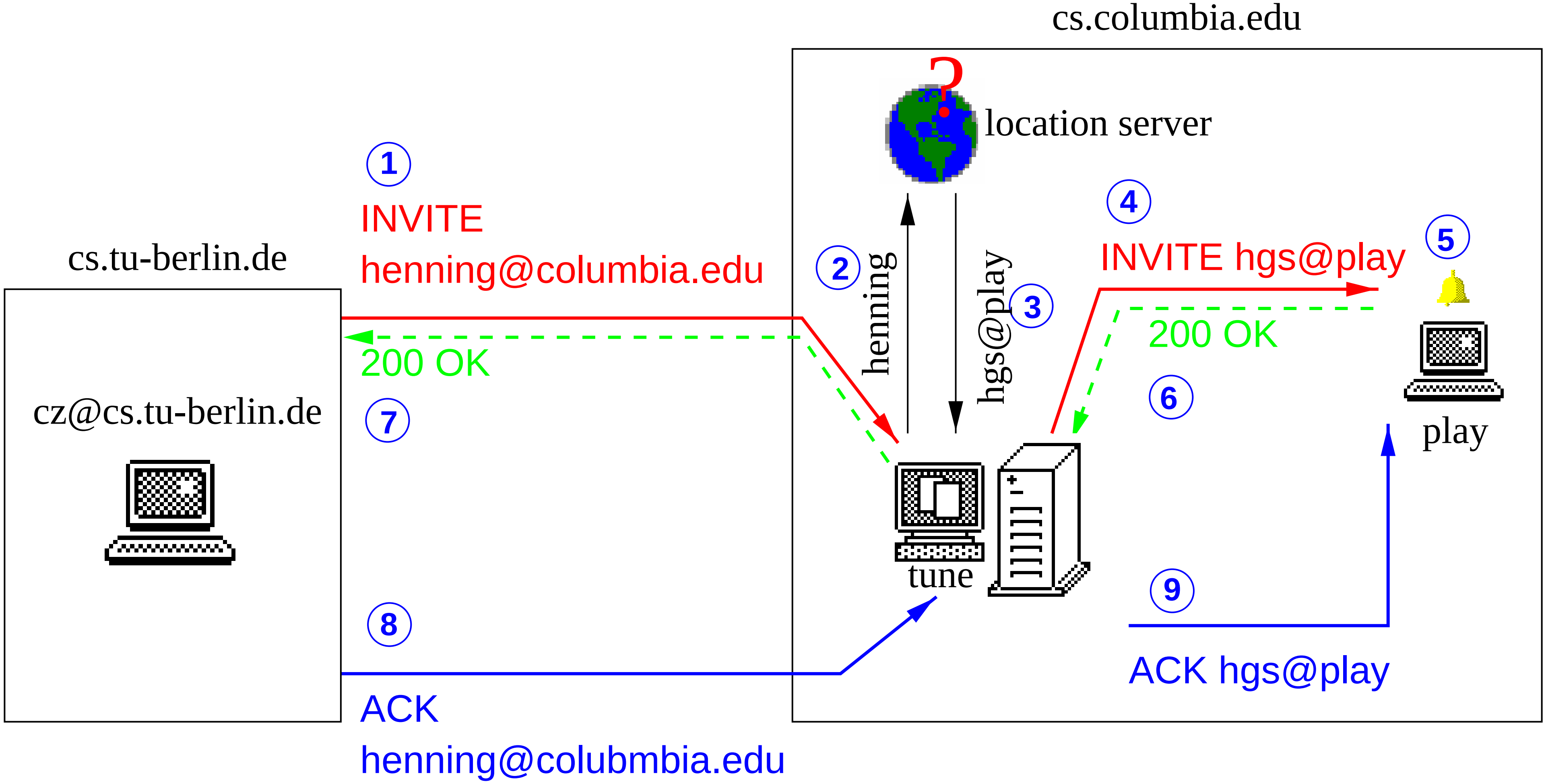
Architecture



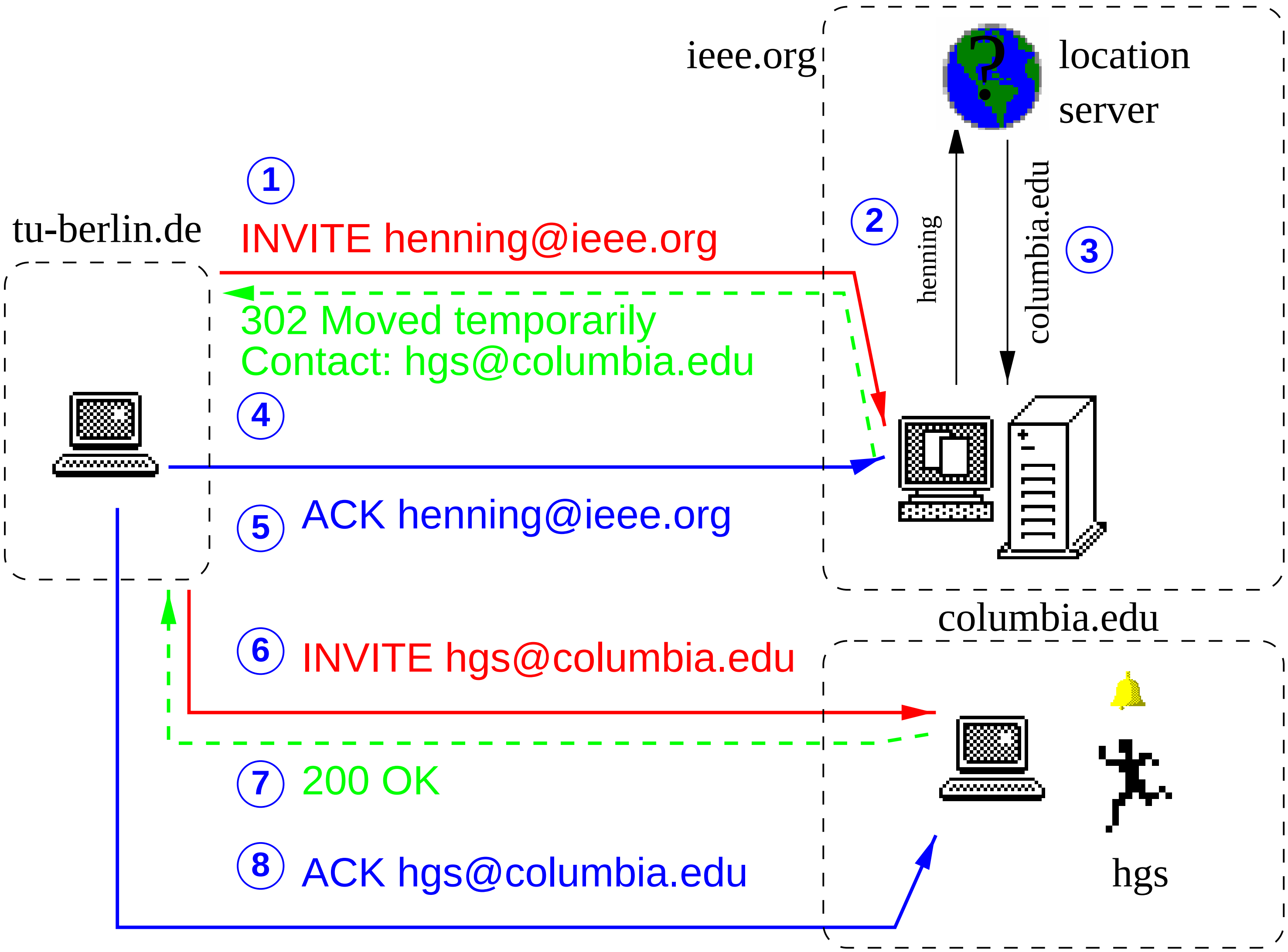
SIP 101

1. SIP = signaling protocol for establishing sessions / calls / conferences / ...
2. session = audio, video, game, chat, ...
3. called server may map name to *user@host*
4. callee accepts, rejects, forward (→ new address)
5. if new address, go to step 2
6. if accept, caller confirms
7. ...conversation ...
8. caller or callee sends BYE

SIP Operation in Proxy Mode



SIP Operation in Redirect Mode



SIP Advanced Features

- operation over any packet protocol (UDP, TCP, X.25, . . .)
- multicast invitations ➠ basic ACD
- “interactive web response” (IWR)
- UA ↔ proxy = proxy / redirect ↔ proxy / redirect
- stateless proxies: self-routing responses
- forking proxies: call several in sequence and / or parallel
- security: basic (password), digest (challenge / response), PGP

SIP Standardization Status

- Feb. 2, 1999: IETF Proposed Standard
- March 17, 1999: IETF RFC 2543
- eligible for Draft Standard: 6 months, 2 implementations $\surd$

Internet phone “appliance”

- phone = \$49.99; PC > \$600 (GPF included)
- *Ethernet phone* ➠ no PBX for switching
- DSP for voice coding ➠ limited memory (128 kB SRAM!)
- implemented minimal IP stack (IP / UDP / RTP, DHCP, SIP)
- no TCP, no DNS
- MP3 radio
- interface to the world


128.59.19.38
SIP Phone 2.0
Please Dial:


ALTERA
MAX
DSP

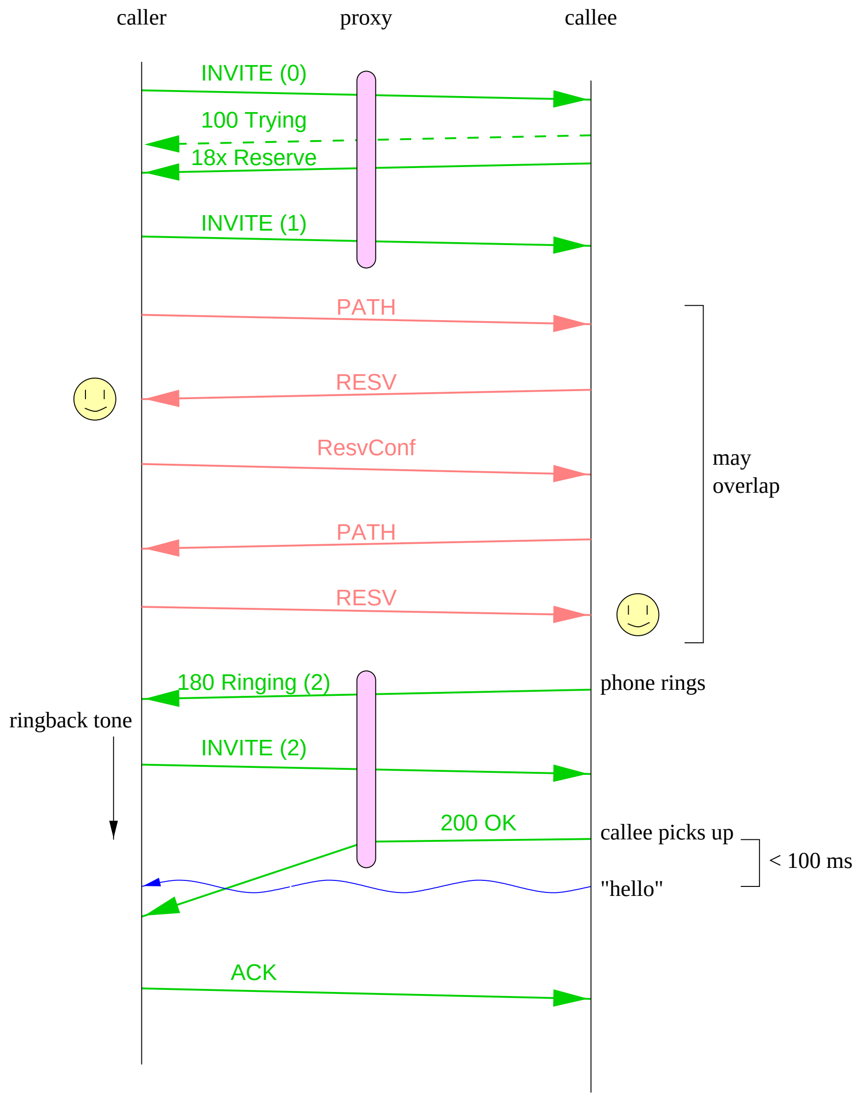
SIP Work Items

- sip-cgi
- call processing language
- reliable provisional (1xx) responses
- caller preferences
- third-party call control
- SIP for subscribe / notify
- SIP–ISUP interworking
- SIP–H.323 interworking
- billing
- reverse channel setup for call progress tones
- pre-ringing resource reservation

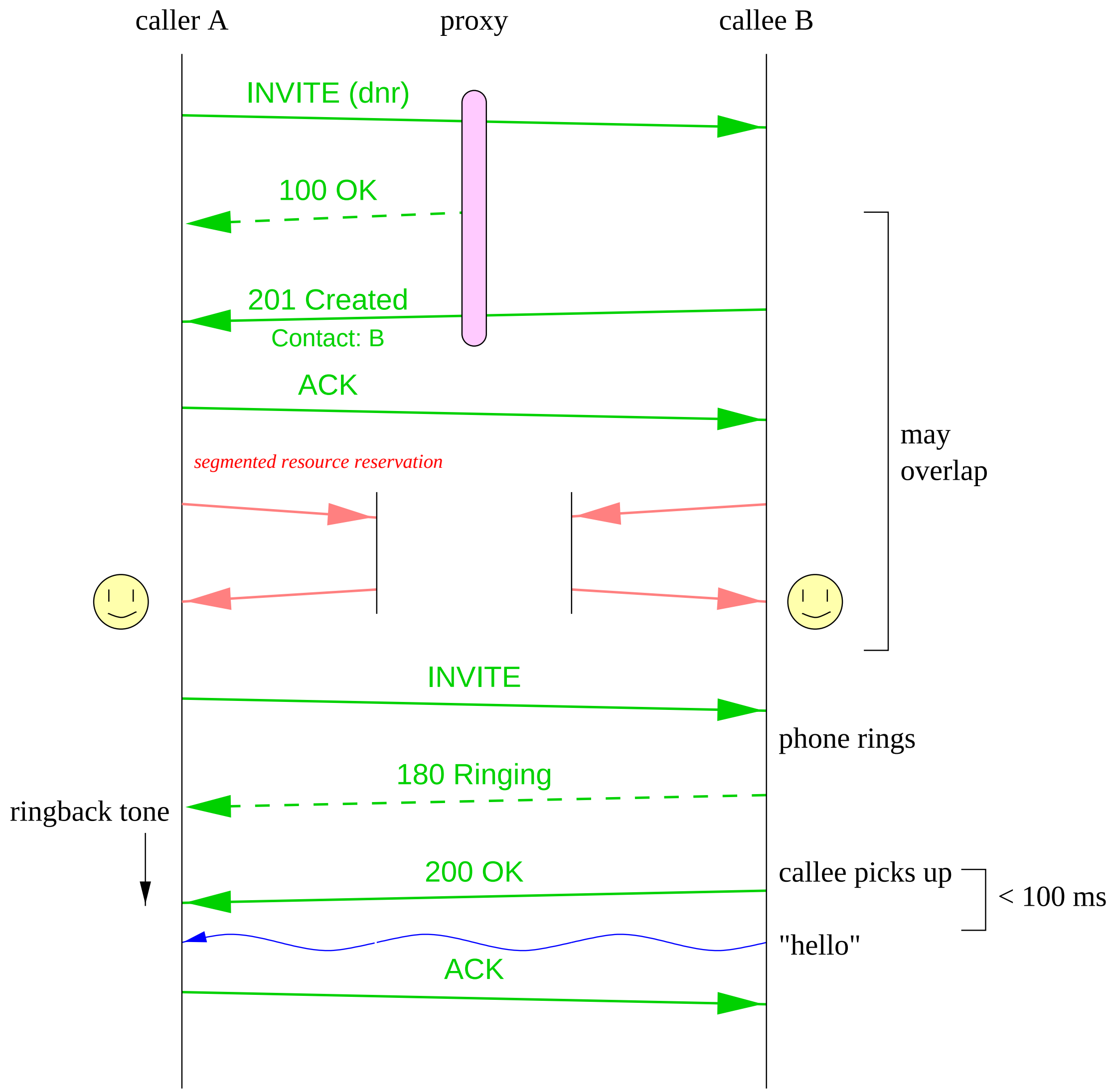
Interaction with QOS

- separate call signaling and resource reservation
- options:
  - diff-serv ➠ no per-call resource reservation
  - end-to-end (RSVP)
  - segmented
- parallel or sequential: should phone ring if not enough bandwidth?
- several options being discussed

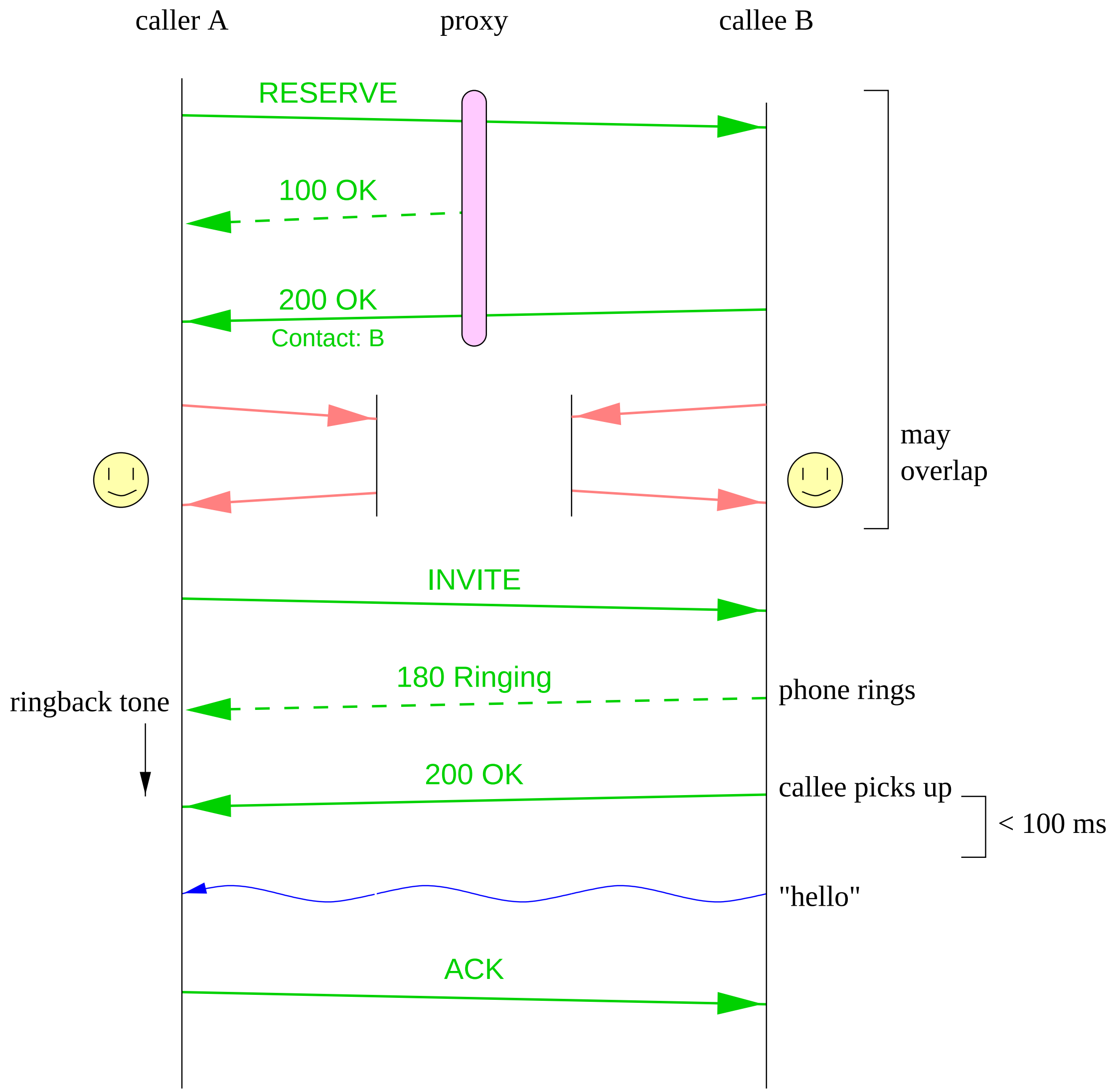
QOS-assured Signaling: One Transaction



QOS-assured Signaling: Segmented



QOS-assured Signaling: New Method



SIP Caller Preferences

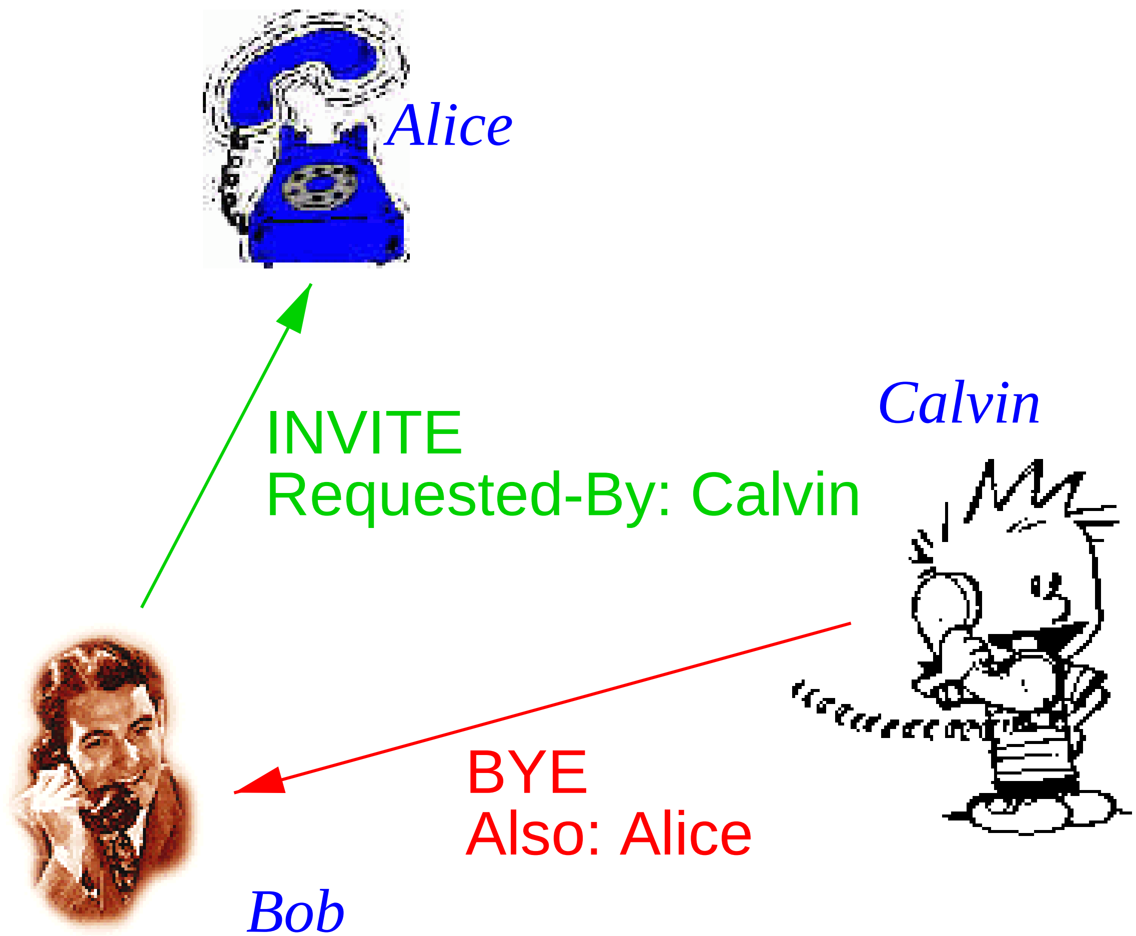
- give *caller* input in forwarding and selection decisions
- “caller proposes, callee disposes”
- examples:
  - forward to home or office
  - type of call: video, fax, chat, ...
  - mobile or landline
  - queue or forwarding to secretary or voicemail
  - languages spoken

Call Control ➠ Mid-Call Features

- basic SIP offers forwarding, hold, call waiting, . . .
- (mid-call) call transfer
- adding parties to full mesh (three-way calls)
- transition between MCU, mesh and multicast
- provide information during transfer
- provide choice: refuse transfer

Example: End-System Blind Transfer

- Calvin transfers Bob to Alice
- Alice knows who asked for transfer
- Bob can refuse transfer



Services

Lots of services . . .

- call redirect to web page
- web IVR
- time-of-day routing
- email: “Joe <sip:joe@foo.com> called”
- follow-me
- distributed home line emulation

. . . but somebody has to create them!

Who Creates Services?

- service providers
- local administrators, vertical application vendors, . . .
- end users

➠ security and reliability concerns:

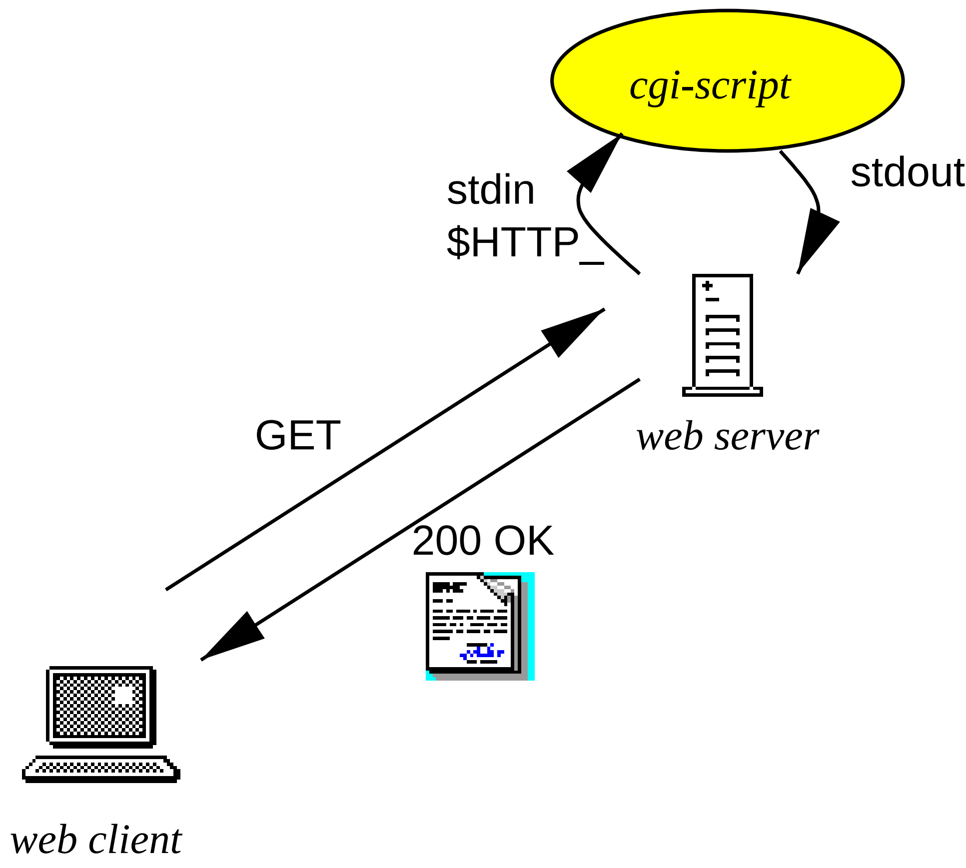
- crash server
- snoop
- calls directed to nowhere

Service Creation Requirements

- rapid development
- rapid deployment: can’t reboot or recompile server
- cross platform: users want to take code with them
- remote installation: code runs far away
- “programmers” may have little software expertise

Web "Service" Creation: Cgi-bin

- cgi = common gateway interface
- e.g., Perl, executable
- request (form): client → server
- server forks process
- form via URL, stdin
- script: web page to stdout

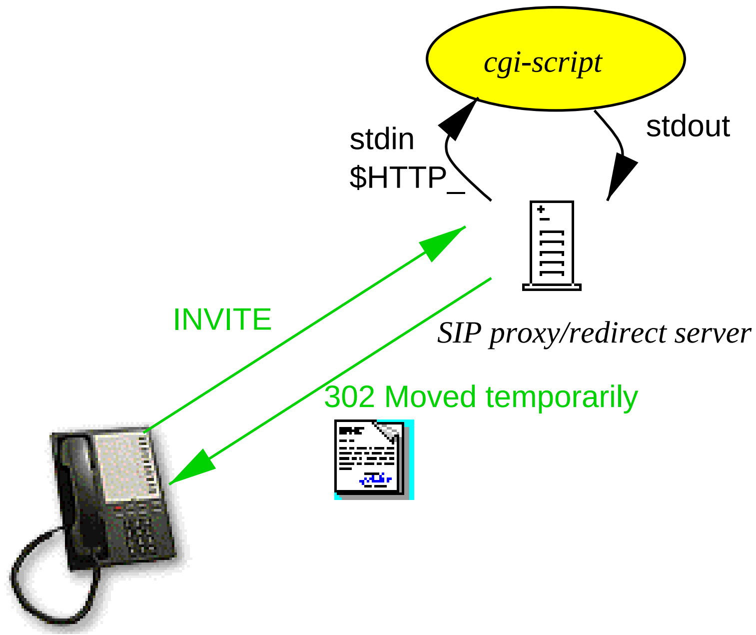


SIP cgi

SIP (cgi) and HTTP (cgi-bin) are similar, but:

- persistent scripts
- initiate proxy
- multiple responses (100, 3xx)

➠ use commands on stdout



SIP cgi Benefits

- any programming language
- can add/change scripts dynamically
- full access to databases, networked services (if script allows)
- can use restricted interpreters for decent security
- minimal SIP knowledge needed

- “call forward unconditional”
- database for forwarding list
- returns error if not in database

```
use DB_File;
sub fail {
    my($status, $reason) = @_;
    print "SIP/2.0 $status $reason\n\n";
    exit 0;
}
tie %addresses, ’DB_File’, ’addresses.db’
    or fail("500", "Address database failure
$to = $ENV{’HTTP_TO’};
if (! defined( $to )) {
    fail("400", "Missing Recipient");
}
$destination = $addresses{$to};
if (! defined( $destination )) {
    fail("404", "No such user");
}
print "CGI-PROXY-REQUEST-TO $destination SIP
print "CGI-Reexecute-On: never\n\n";
untie %addresses;
```

But cgi is Not for Everyone

CGI has access to full SIP power

- ideal for service providers
- users don't want to write Perl scripts
- lots of error conditions
- "We're sorry, the Perl script you have dialed has crashed. Please try again later."

Want restricted functionality:

- protect server resources
- allow limited services
- provable correctness
- bounded execution time

Call Processing Language

- special-purpose scripting language
- guaranteed safe
- XML-based ➠ hand or tool-generated

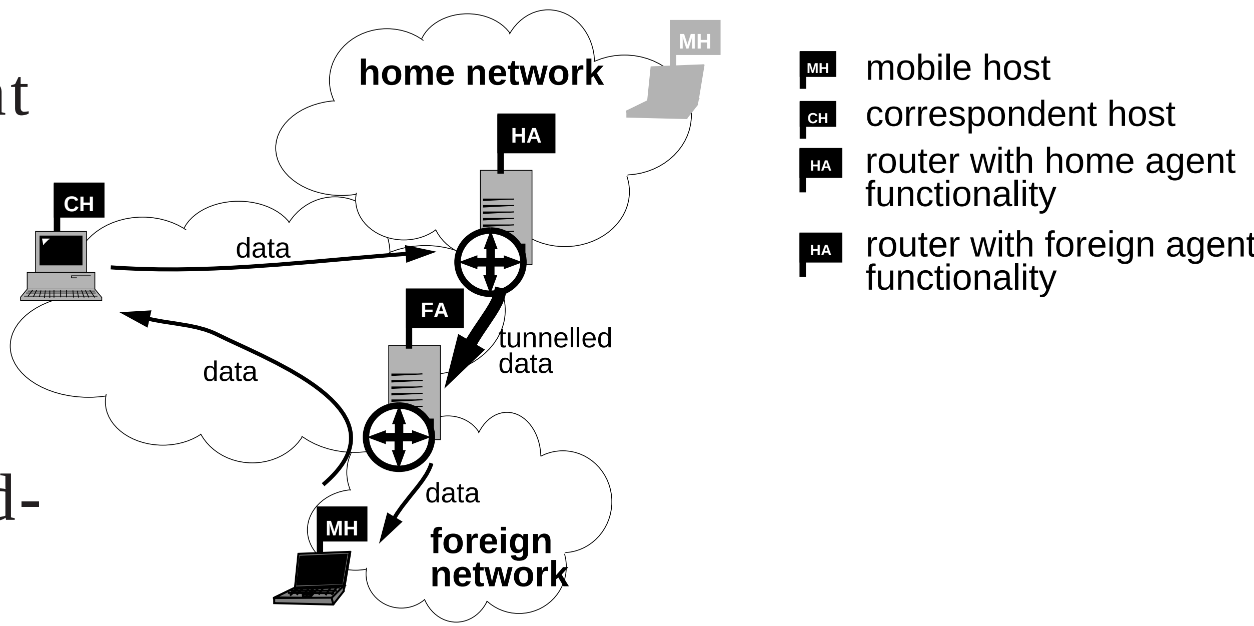
```
<call>
  <location url="sip:jones@jonespc.example.c
     <proxy timeout="8s">

       <busy>
         <location url="sip:jones@voicemail
                   id="voicemail" >
            <proxy />
         </location>
       </busy>

       <noanswer>
         <link ref="voicemail" />
       </noanswer>
     </proxy>
  </location>
</call>
```

Getting Scripts into the Server

- script based on:
  - inbound proxy: From
  - outbound proxy: To
  - classes of users: administrative
- upload
  - pre-install on server
  - web form ⟶ cgi script ⟶ CPL, sip-cgi
  - web upload
  - upload via REGISTER

Mobility

- new network: IP addr. changes (DHCP)
- mobile IP hides address changes
- but: little deployment
- encapsulation
- dog-legged routing
- may fail with IP address filters



Aside: Where is Mobile IP Needed?

Not needed if short-lived, restartable connections ➠

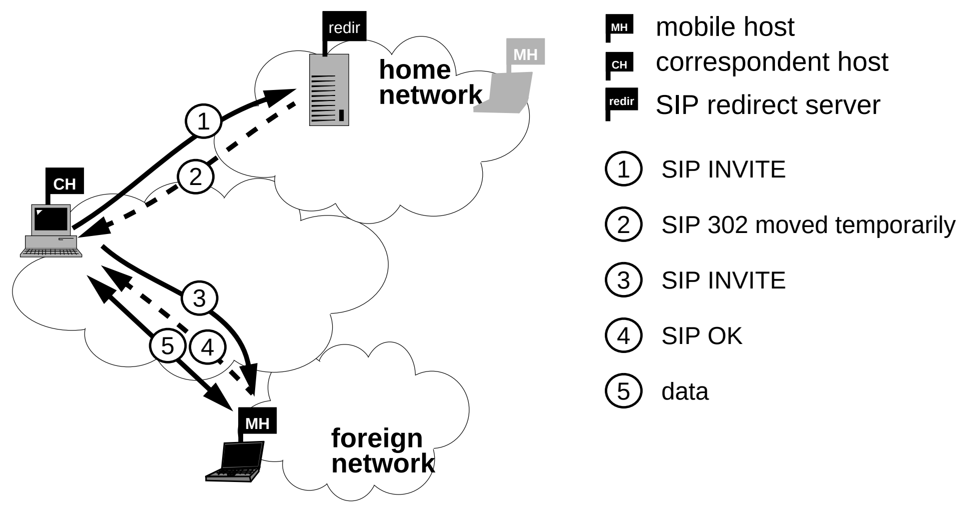
http	short, stateless
smtp	short, restartable
pop, imap	short, restartable
telnet	yes, but rarely used by mobiles (?)
ftp	restartable, rare
chat, irc	yes, but fixable (proxy, protocol)

SIP Mobility Overview

- pre-call mobility ➠ SIP proxy, redirect
- mid-call mobility ➠ SIP re-INVITE, RTP
- recovery from disconnection

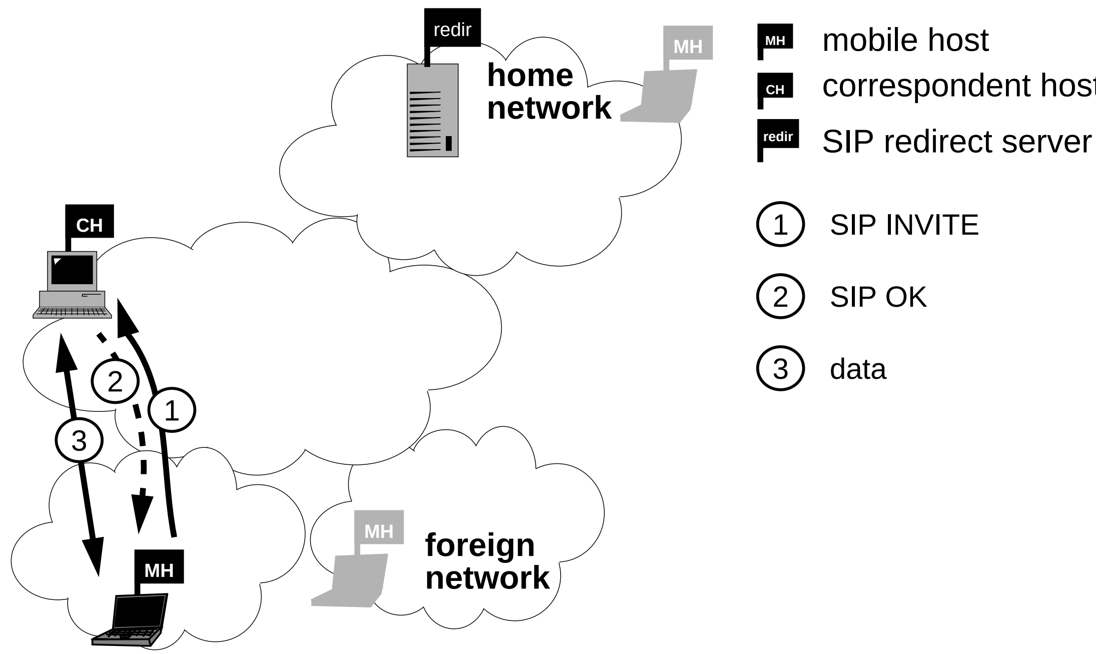
SIP mobility: pre-call

- MH acquires IP address via DHCP
- optional: MH finds SIP server via multicast REGISTER
- MH updates home SIP server
- optimization: hierarchical LR (later)



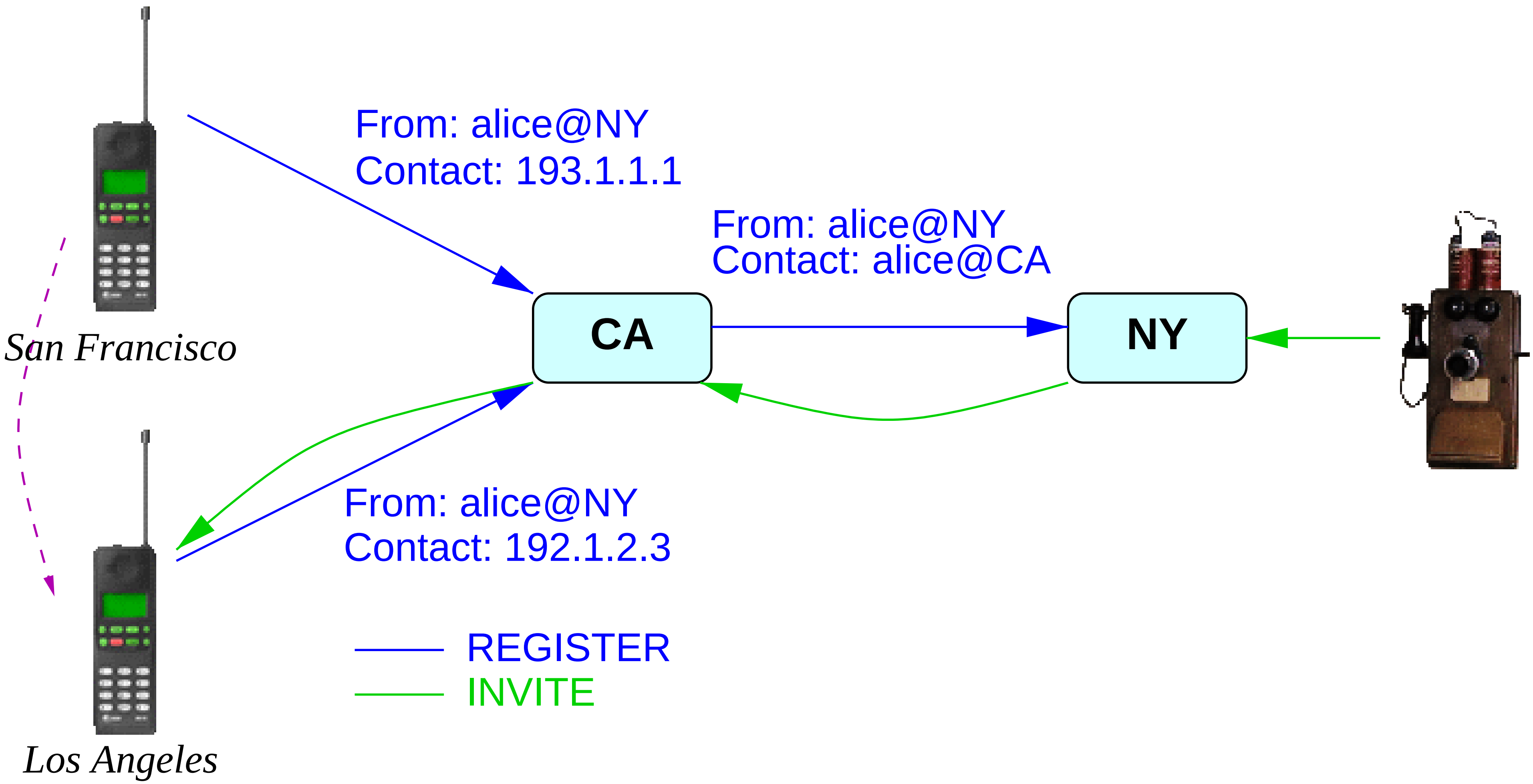
SIP Mobility: Mid-call

- MH→CH: new INVITE, with Contact and updated SDP
- re-registers with home registrar



SIP Mobility: Multi-stage Registration

Don't want to bother home registrar with each move



Movement Detection: Ad-Hoc Mode

no “access point” ➠ regular station as BS

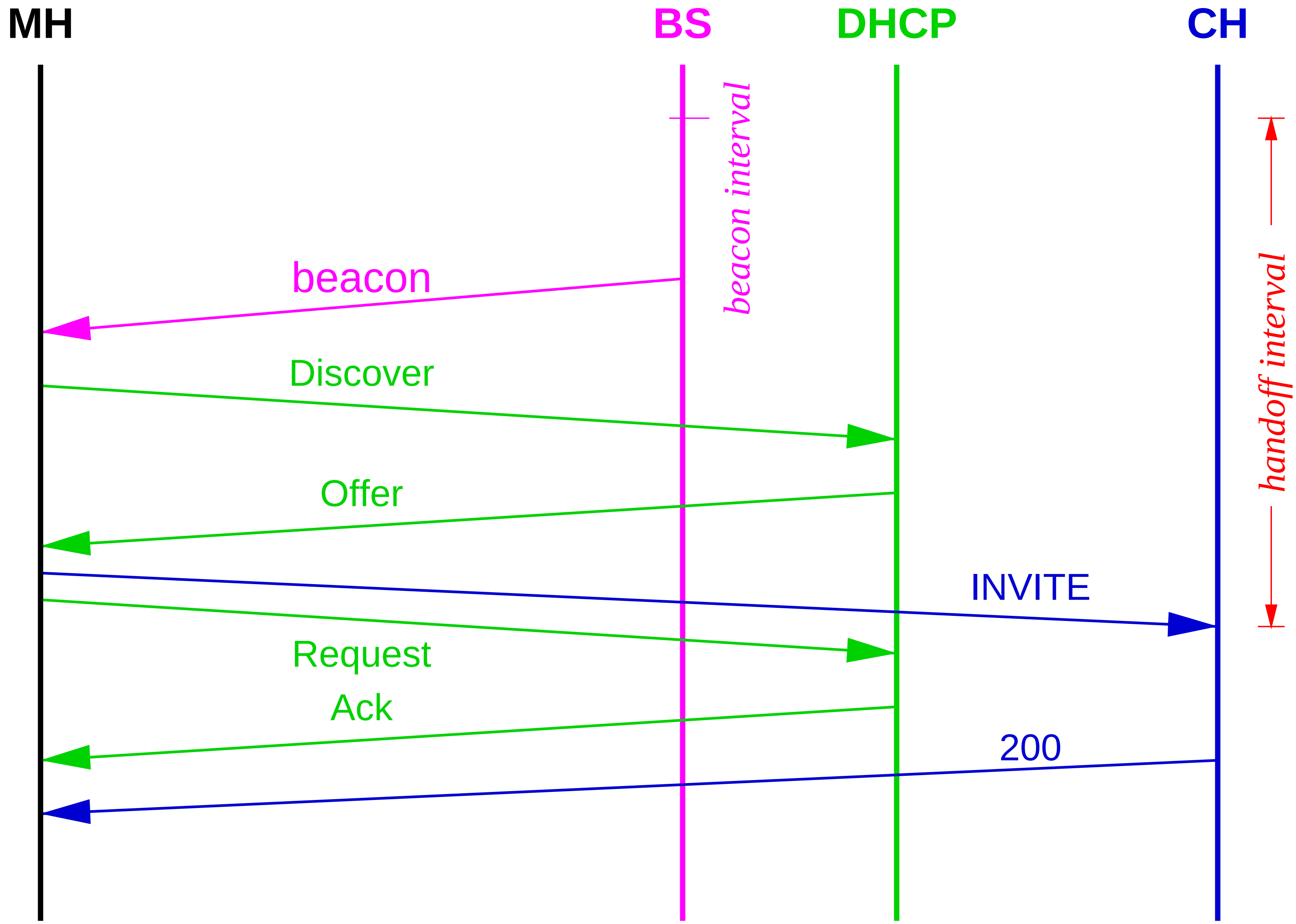
- BS serves as default router
- periodic multicast beacon
- pick best: driver provides SNR, strength
- could use regular multicast packets for quick BS discovery

Movement Detection: Infrastructure Mode

access point (AP) for BSS

- attachment handled by MAC layer, invisible to application
- BSSID is contained in 802.11 packet, but
  - BSSID not visible to application
  - driver doesn't get notified if MH attaches to new AP
- modified driver that polls hardware?

Handoff Performance



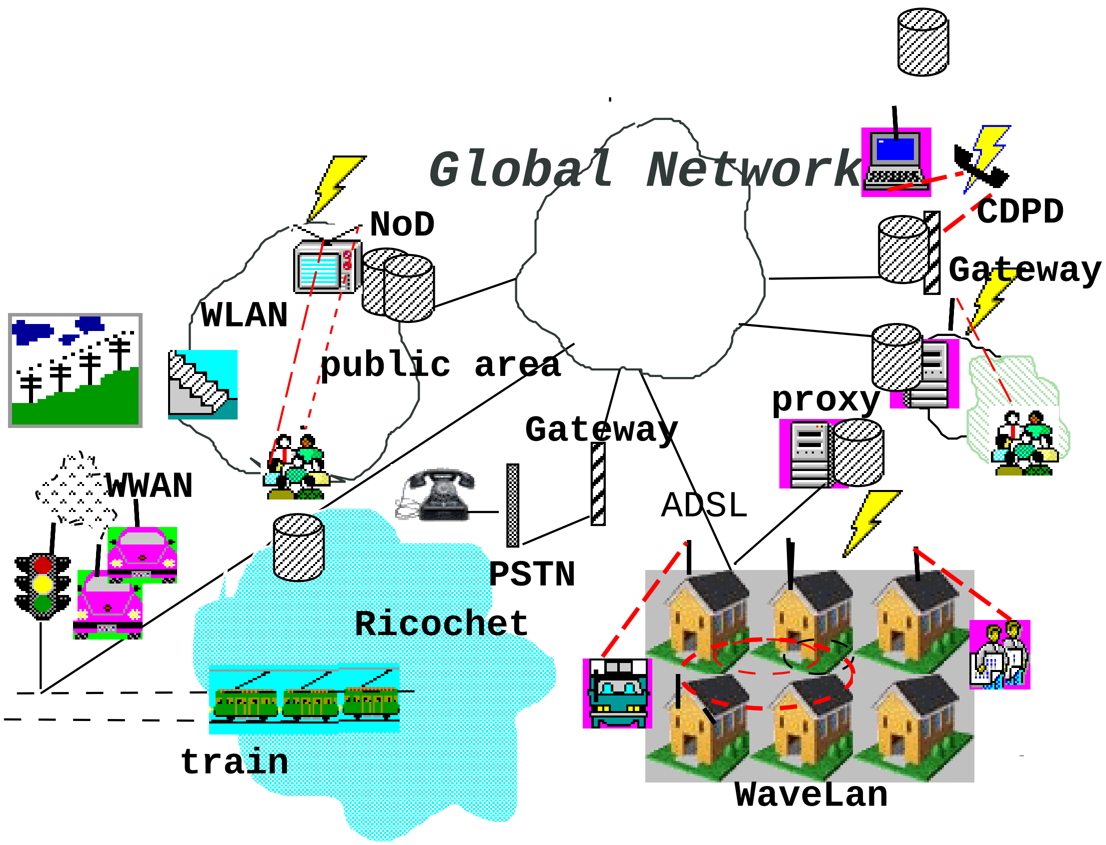
Open Issues

- handoff performance in a loaded network
- soft hand-off: IP-level vs. application proxies
- soft hand-off for 802.11 infrastructure mode possible?
- RTP issues: collision detection

Other Research Efforts in IRT

- forward error correction
- mobile Internet access
- Internet radio
- pricing for congestion-feedback: RNAP
- simplified resource reservation protocols: YESSIR
- scalable resource reservation: BGRP

Internet mobile services



Internet mobile services: “social” ad-hoc networks

- connection sharing
  - multiple wireless networks: 2–10 Mb/s (WL Ethernet, IR) to 28 kb/s (Ricochet, CDPD)
  - share wide-area connections with neighbors
  - load sharing with mobile or in-home gateways
- social caching
  - subway model: in-car high-speed receiver updated in stations
  - socially optimal retrieval
  - anticipatory caching of streaming media
  - “leave the newspaper behind”

Conclusion

- more than just PSTN on packets
- signaling via SIP
- terminal mobility as special case
- programming of services

For more information...

Internet and telecom statistics:
http://www.cs.columbia.edu/~hgs/internet

Papers: http://www.cs.columbia.edu/IRT

RTP: http://www.cs.columbia.edu/~hgs/rtp

SIP: http://www.cs.columbia.edu/sip