SIP: more than grandma's phone calls

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

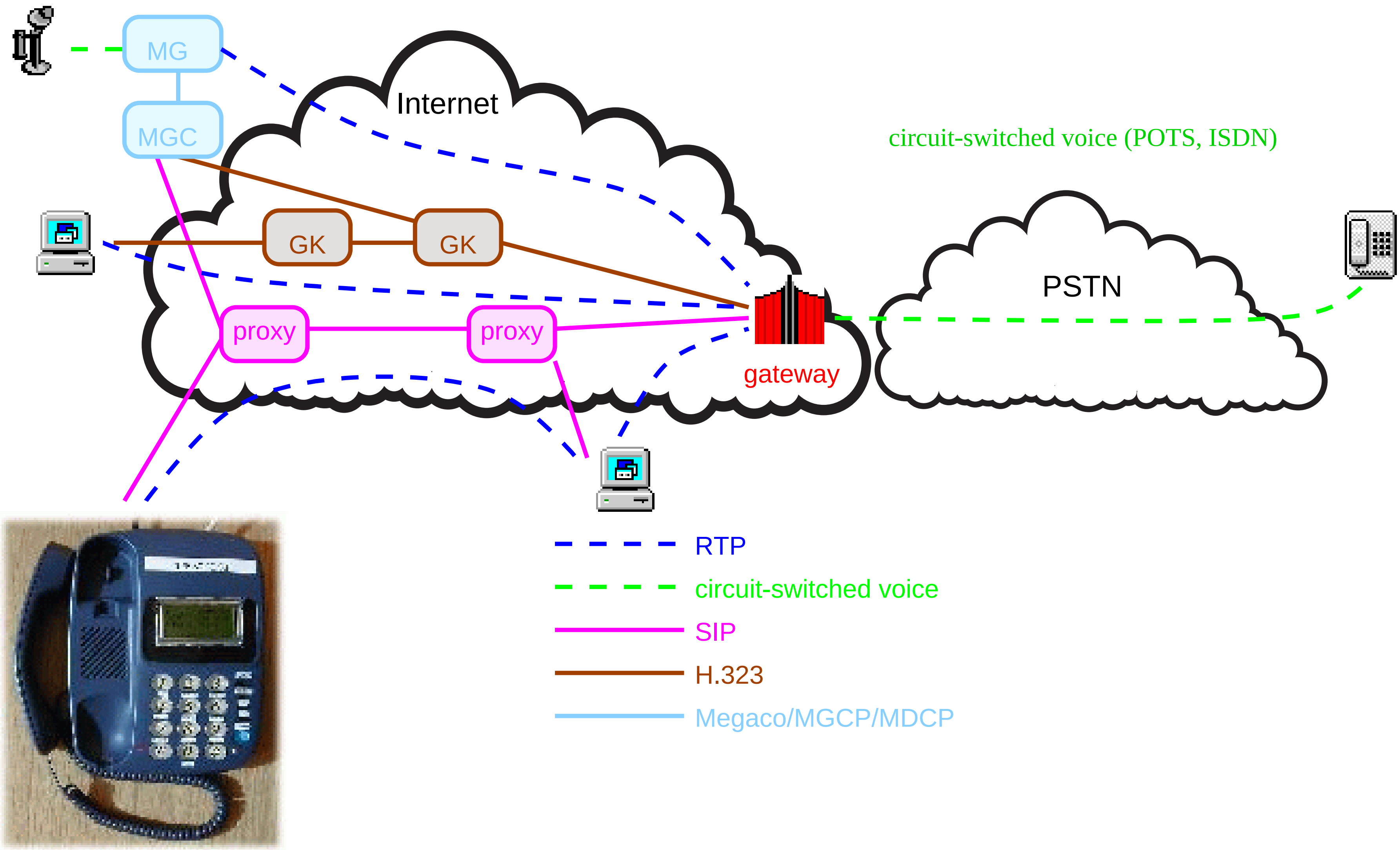
VON Europe Summer 1999 (Helsinki)
Session S13: *SIP Update*

June 22, 1999

Overview

- overview/review
- standardization status
- bake-off
- extensions:
  - interaction with QOS
  - caller preferences
  - call control
- mobility and wireless

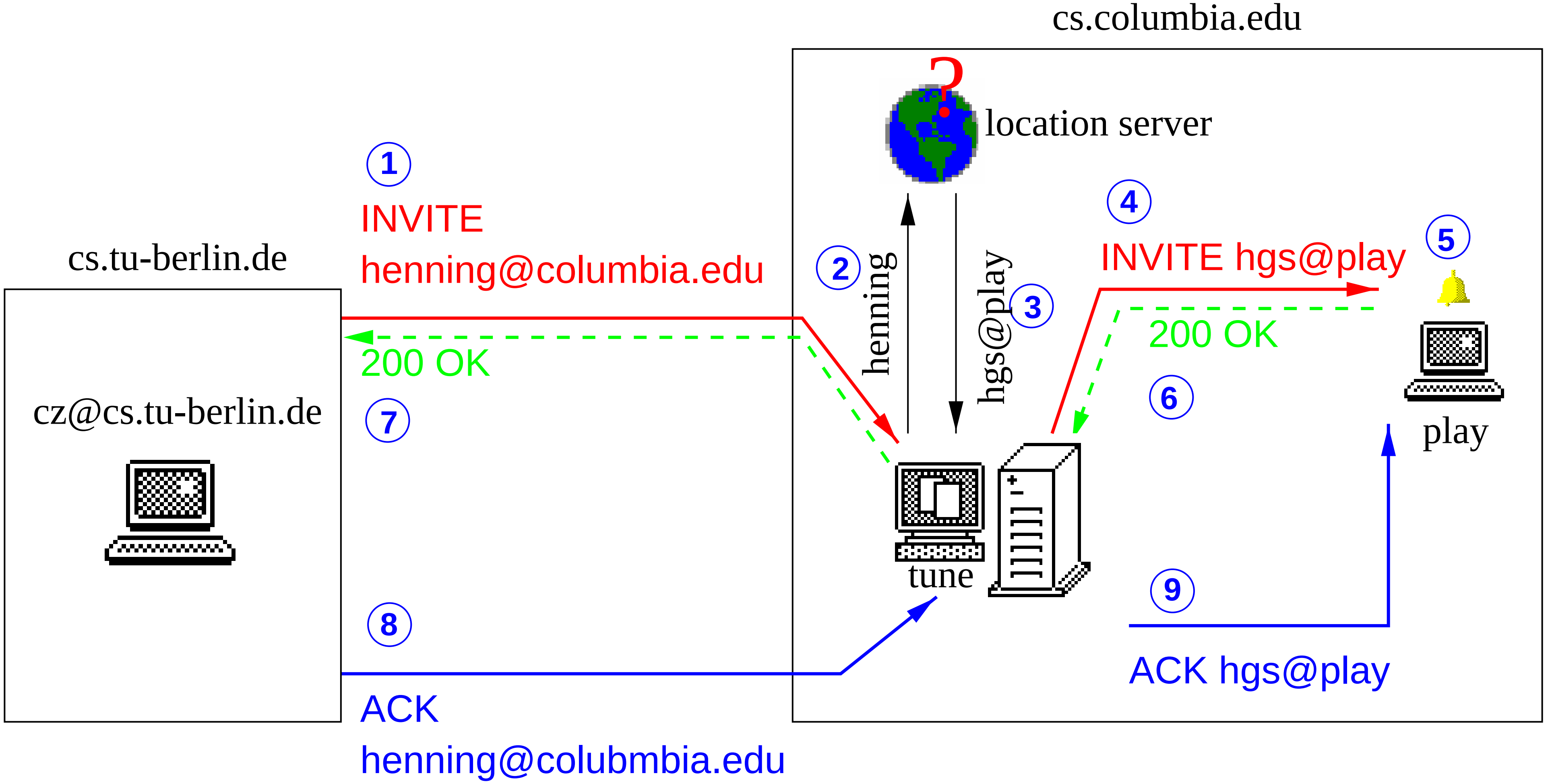
Architecture



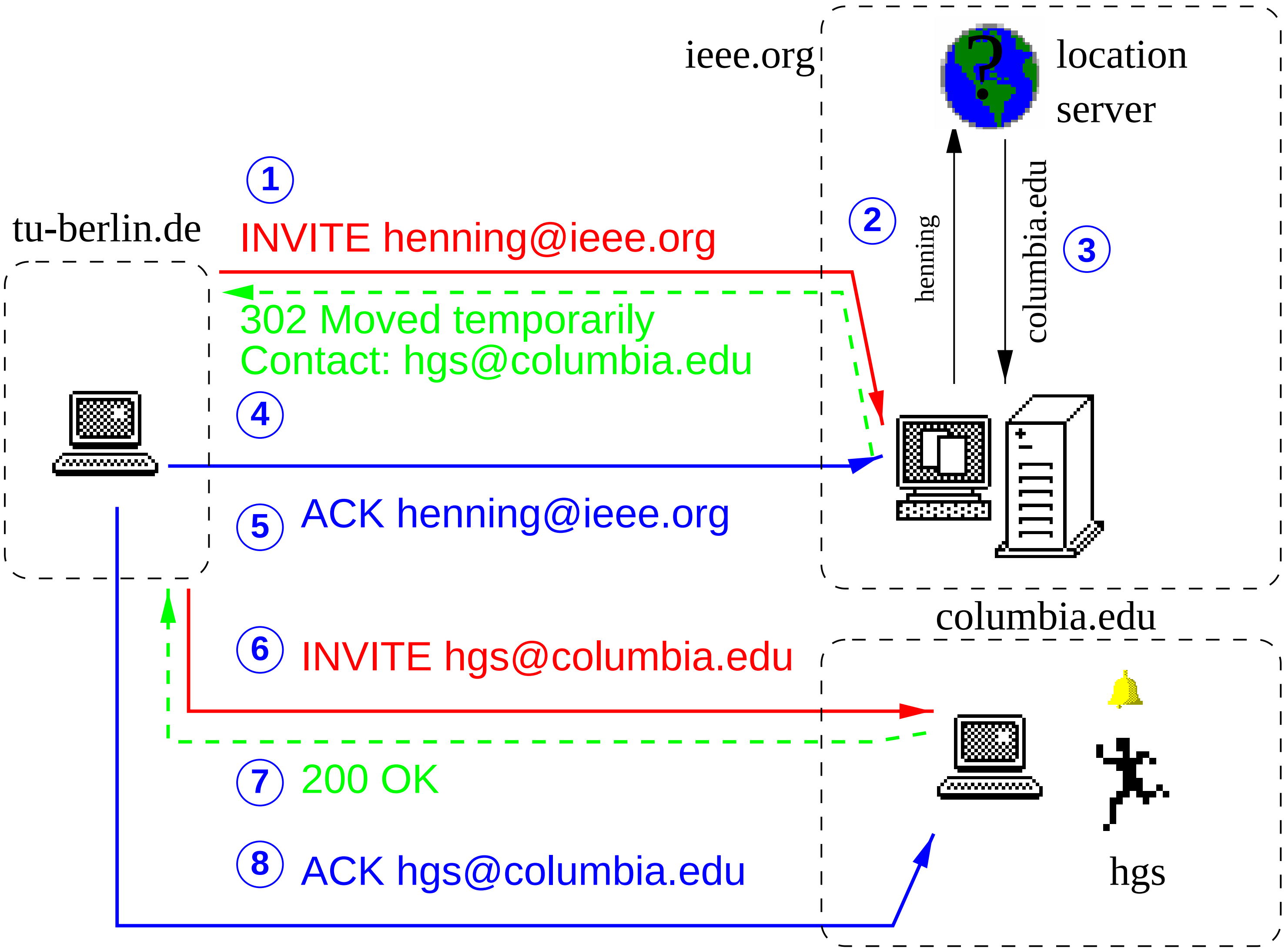
SIP 101

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 $\leftrightarrow$ proxy = proxy/redirect $\leftrightarrow$ 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 $\surd$

SIP Bake-Off

- 35 implementors met at Columbia University, April 8th/9th, 1999
- tested
  - hardware
  - PSTN gateways
  - proxy/redirect servers
  - clients
  - test instrument, . . .

SIP Bake-Off Participants

- 3Com
- Alcatel
- Cisco
- British Telecom
- Columbia University
- Dialogic
- dynamicsoft
- Ellemtel
- Ericsson (2)
- Helsinki Univ. of Technology
- Hewlett-Packard (2)
- Lucent
- MCI Worldcom
- Mediatrix
- Nortel
- Pingtel

SIP Bake-Off Goals

- basic call set-up
- registration, user location
- proxies and redirect server operation
- advanced features: security
- identify implementation bugs and robustness issues
- identify spec ambiguities

SIP Bake-Off Results

- almost all implementations could establish basic calls – either on arrival or after minor on-site fixes
- tested redirection, proxying, security, registration, . . .
- generated interoperability test cases and tools
- will fold clarifications into Draft revision of RFC and web page at http://www.cs.columbia.edu/~hgs/sip
- second bake-off early August (Melville, NY), with advanced features (DNS SRV, forking, call routing, . . .)
- public test servers:
  - sip:sip.pcs.ellemtel.net
  - sip:siphappens.com (3Com)
  - sip:sip.pulver.com (Columbia sipd)

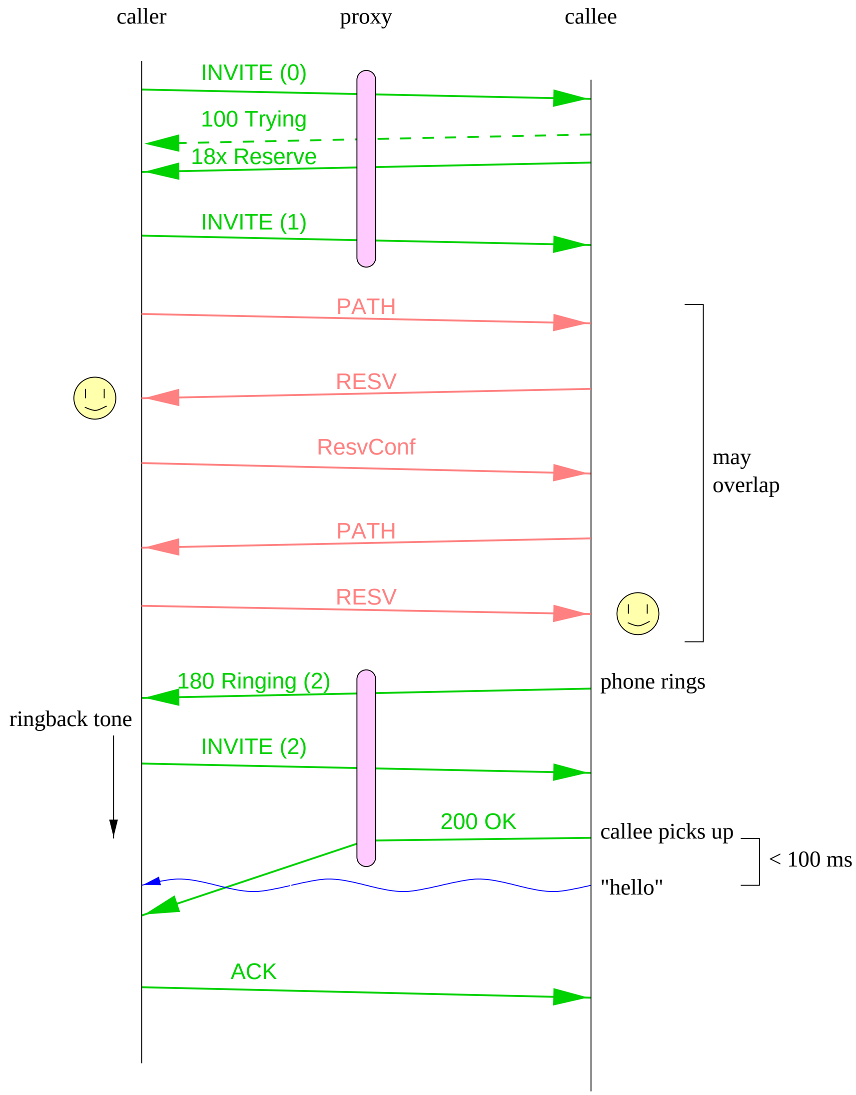
SIP Work Items

- 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

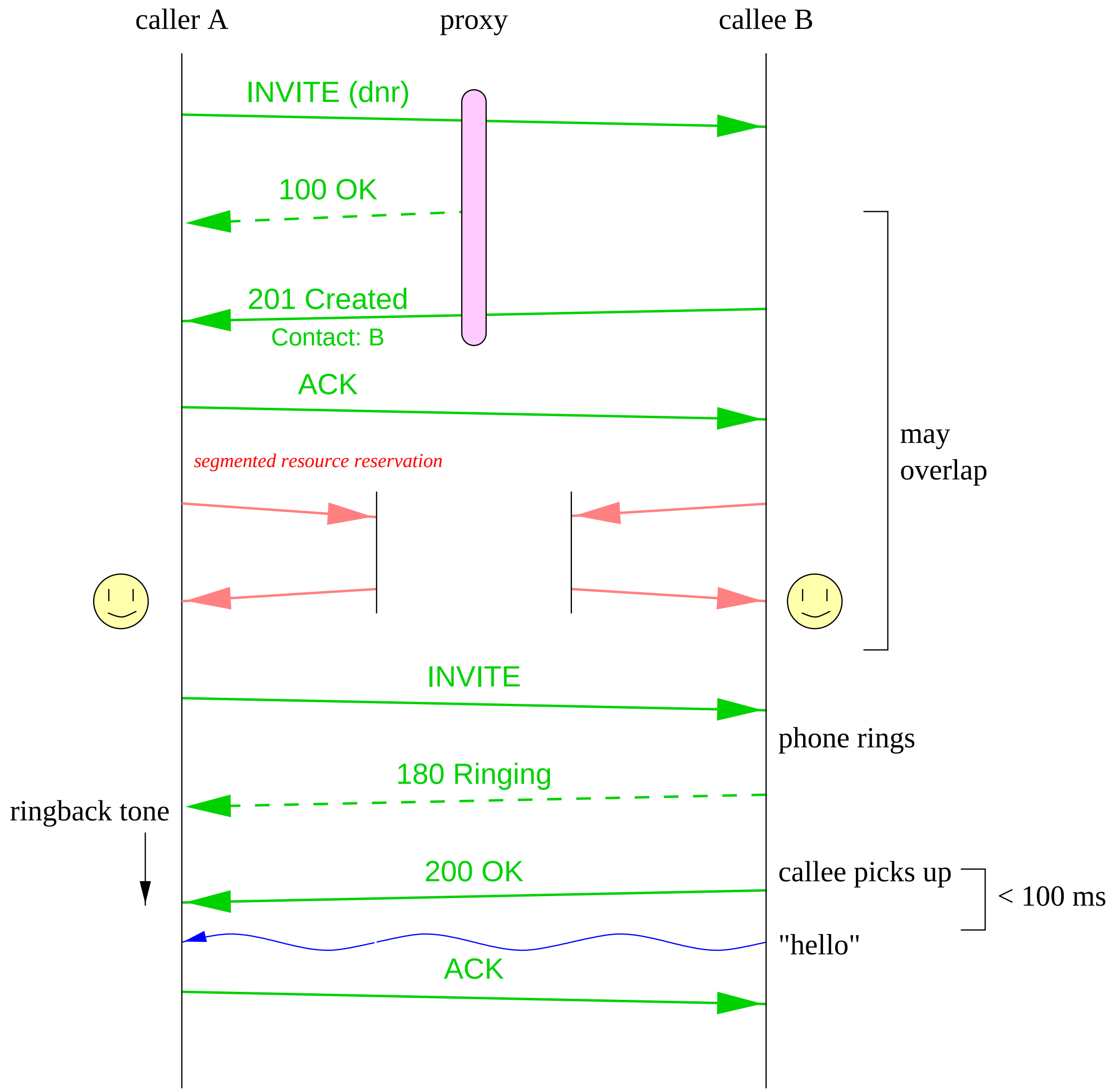
Interaction with QOS

- 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

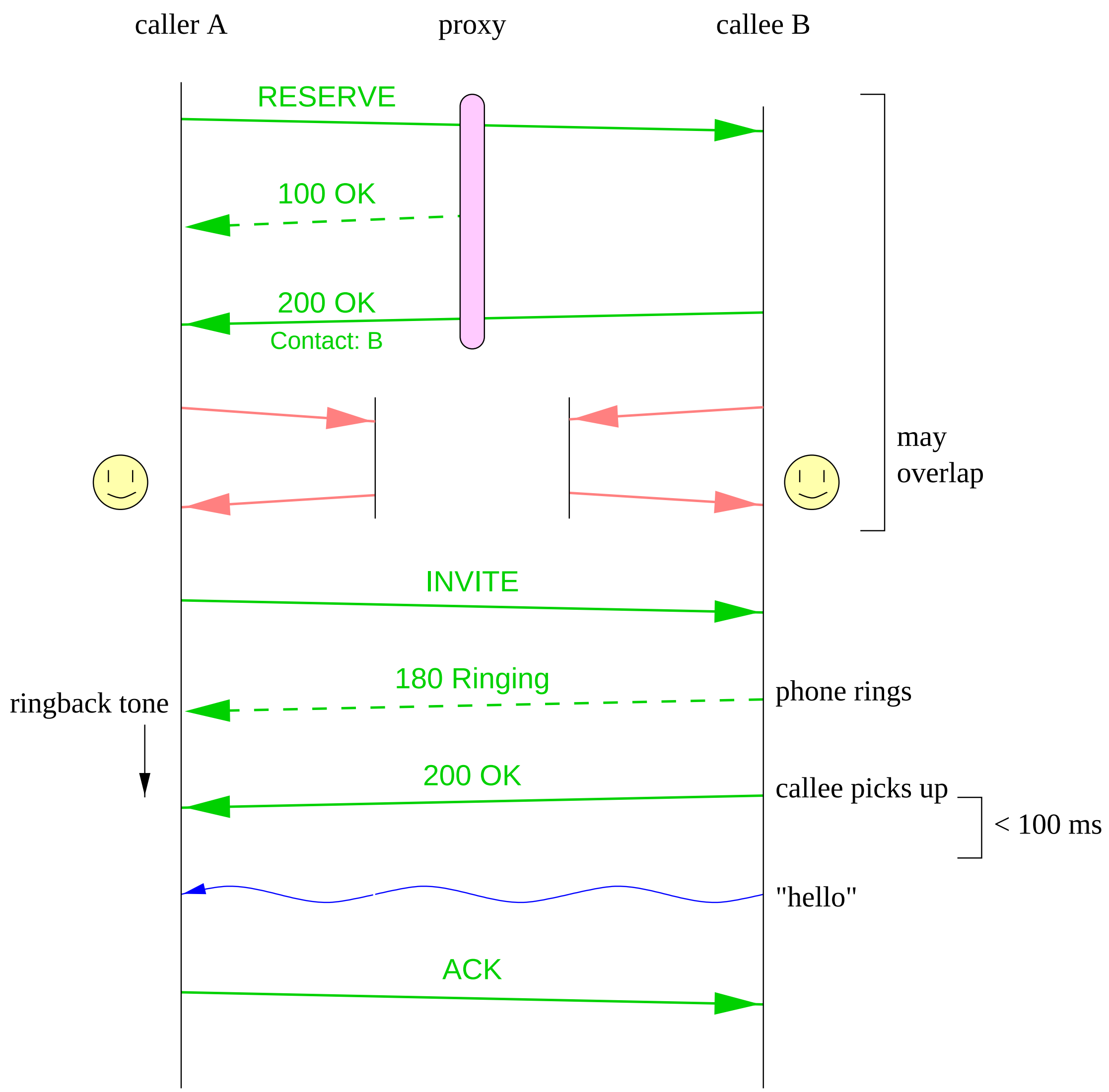
QOS-assured signaling: one transaction



QOS-assured signaling: segmented



QOS-assured signaling: new method



SIP caller preferences

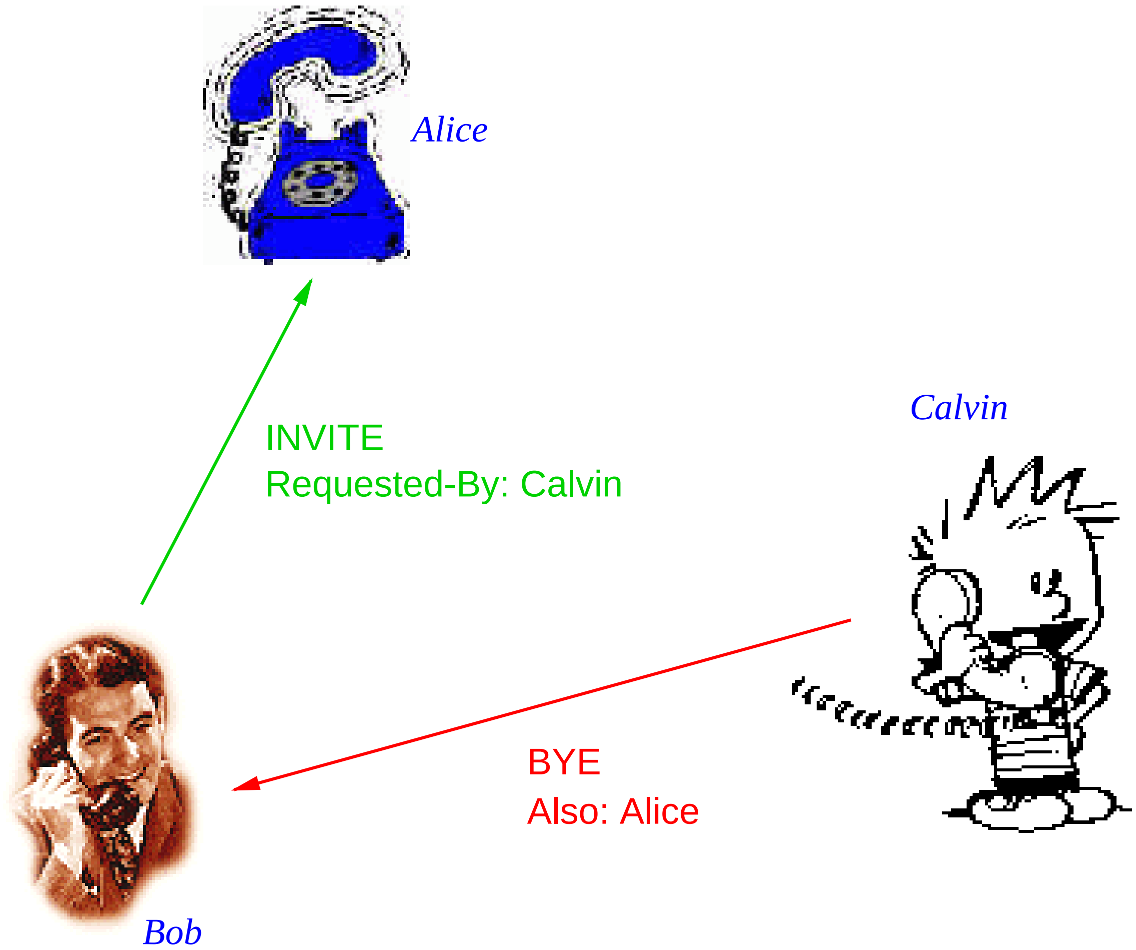
- give *caller* input in forwarding and selection decisions
- "caller proposes, callee disposes"
- examples:
  - forward to home or office
  - type of call: video, fax, chat, . . .
  - mobile or landline
  - queue or forwarding to secretary or voicemail
  - languages spoken

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:

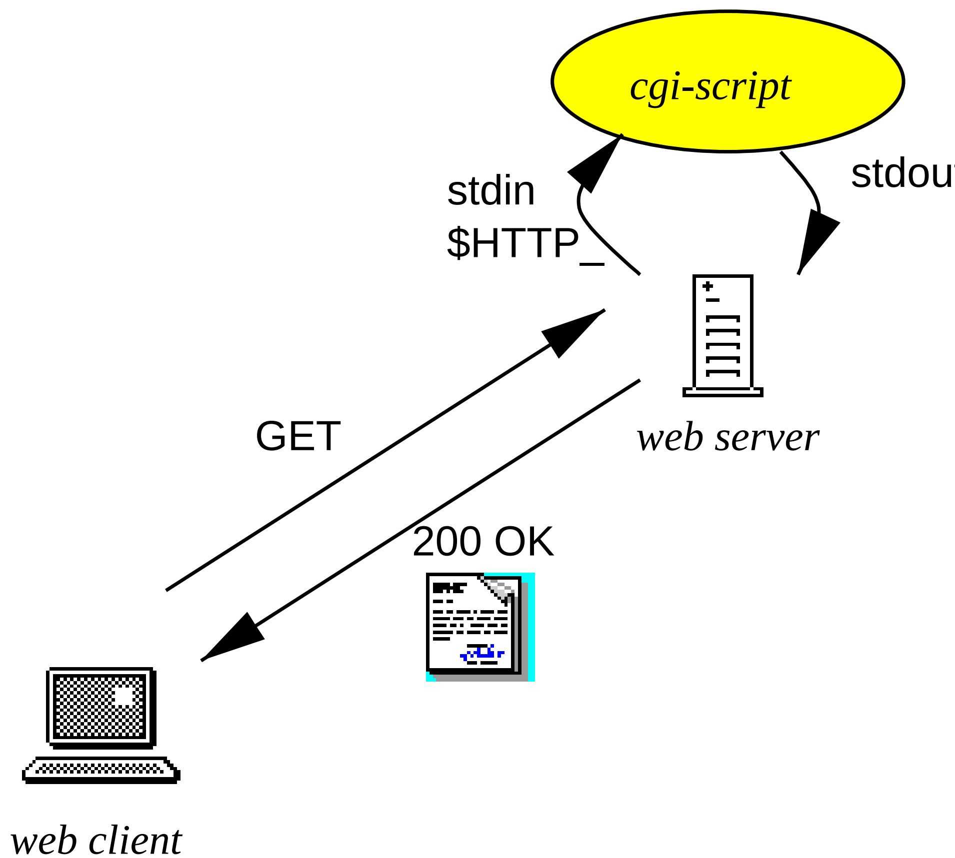
- crash server
- snoop
- calls directed to nowhere

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
- typically, Perl, but can be executable
- request (form) from client → server
- server forks process
- send form content via URL or stdin
- script writes web page to stdout

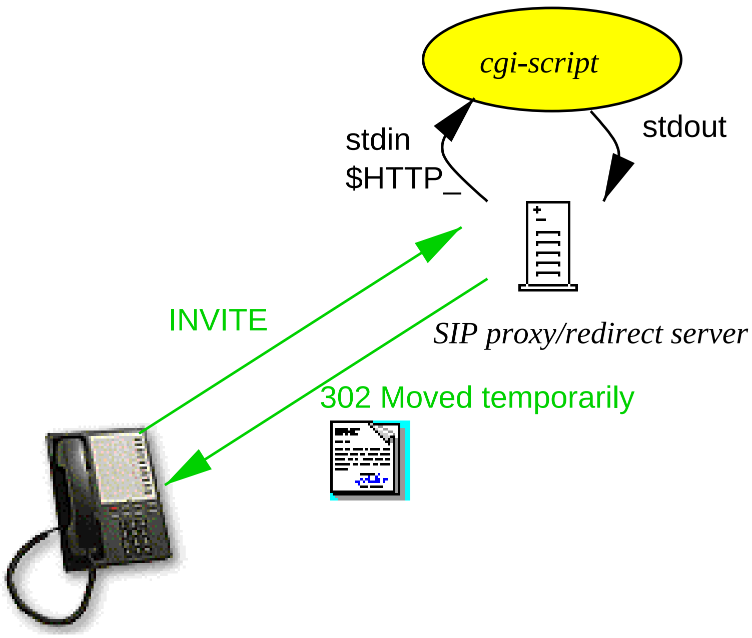


SIP cgi

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

Example perl script

- “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

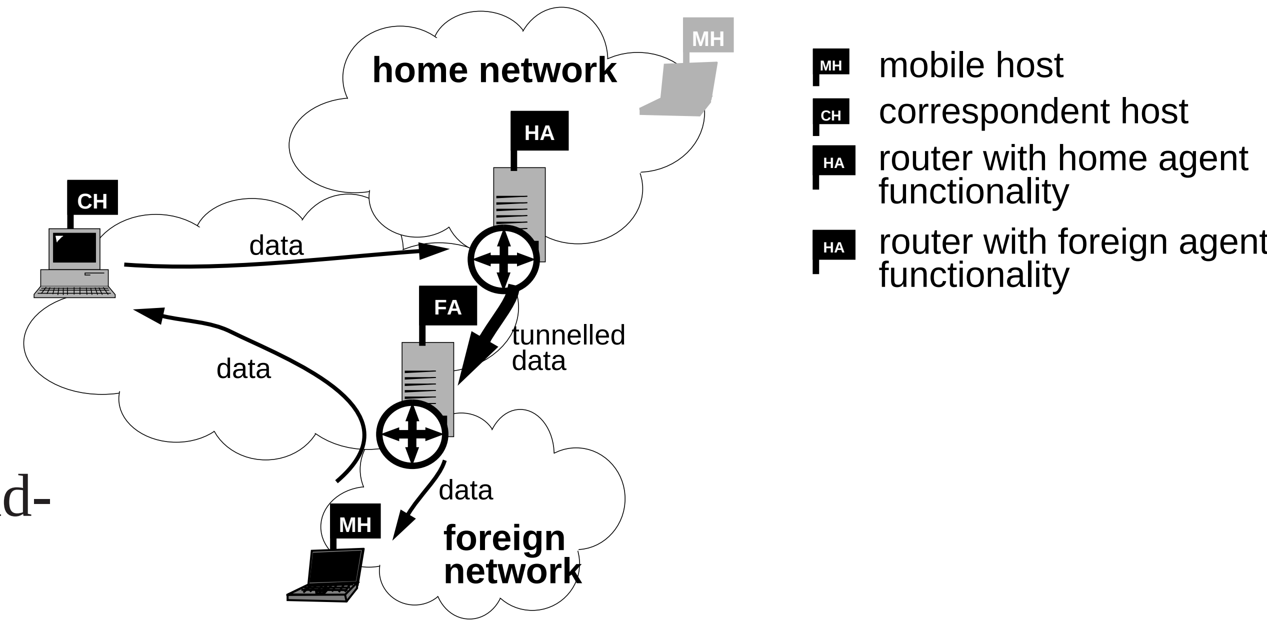
```
<call>
  <location url="sip:jones@jonespc.example.c
     <proxy timeout="8s">

       <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 $\longrightarrow$ cgi script $\longrightarrow$ CPL, sip-cgi
  - web upload
  - upload via REGISTER

Mobility

- move to new network ➠ IP address changes (DHCP)
- mobile IP hides address changes
- but: little deployment
- encapsulation overhead
- dog-legged routing
- may not work with IP address filtering

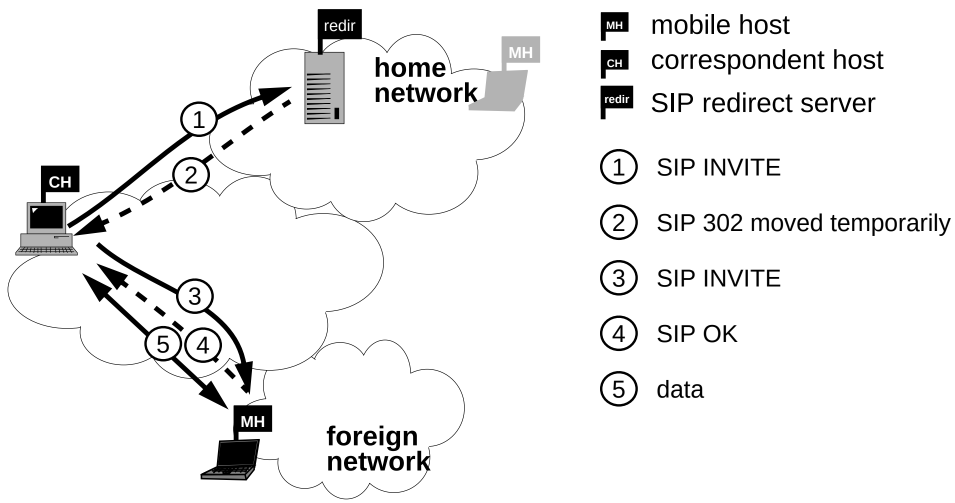


SIP mobility overview

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

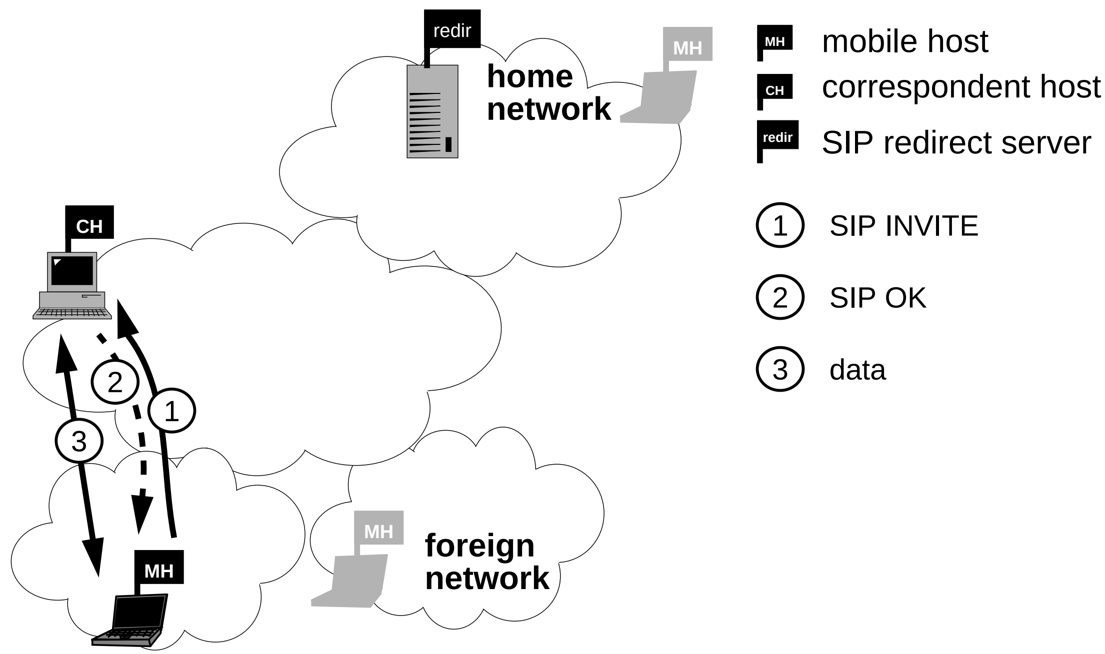
SIP mobility: pre-call

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



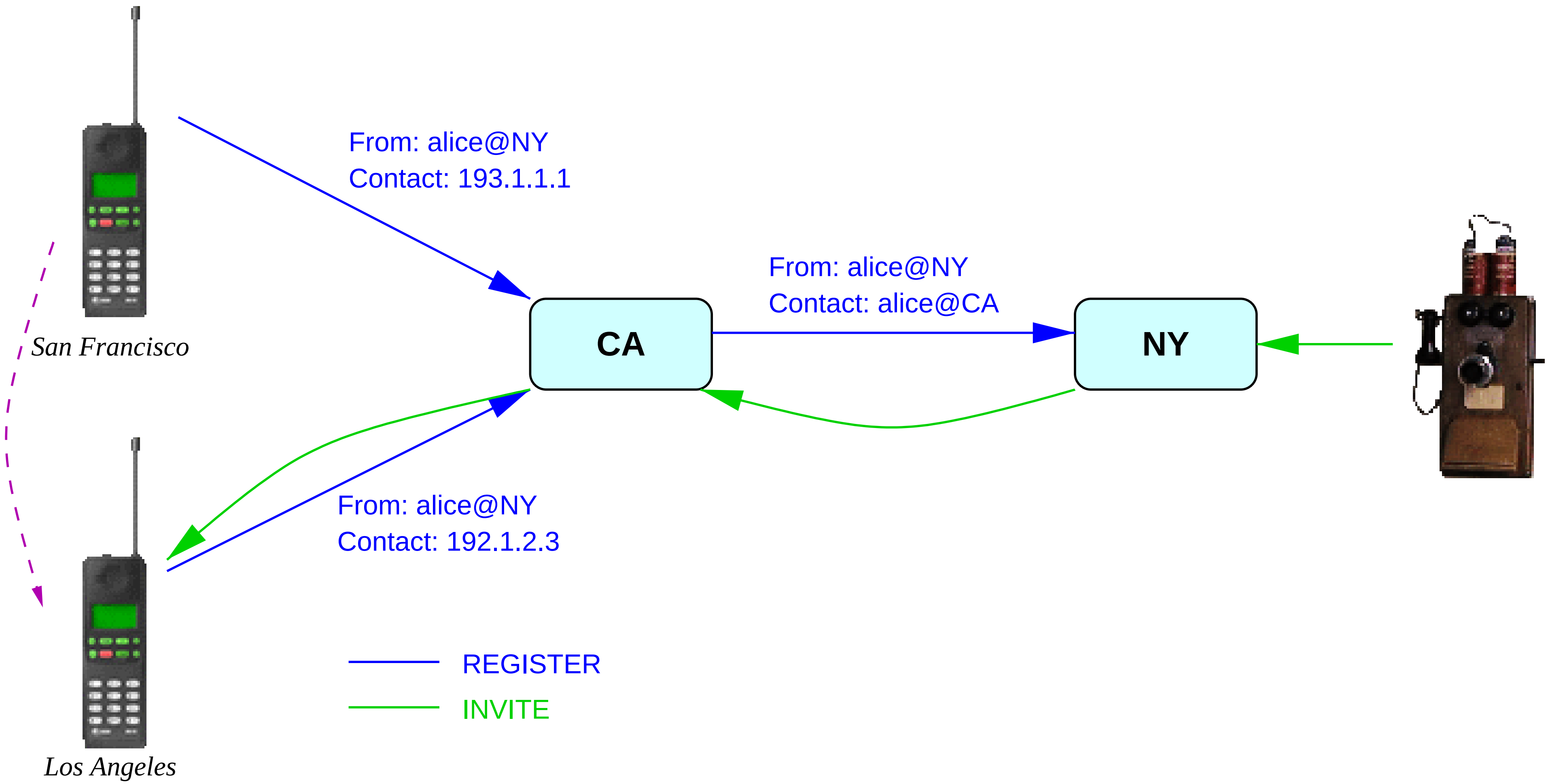
SIP mobility: mid-call

- 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



Conclusion

- SIP basic standard stable
- multiple interoperating implementations
- new backward-compatible features:
  - QOS
  - mobility
  - caller preferences
  - call transfer
- 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/~hgs/sip