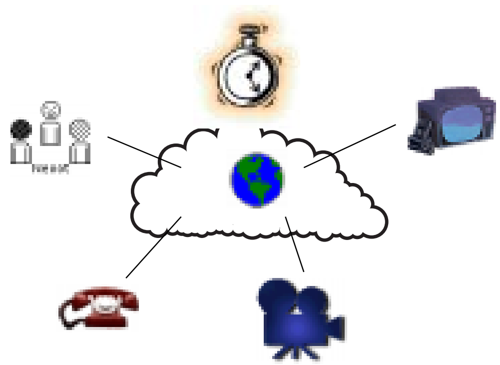
Making the Phone Ring: Signaling for Internet Telephony

Henning Schulzrinne
Internet Real-Time Lab
Columbia University
hgs@cs.columbia.edu



Telcordia General Colloquium, April 7, 2000

(Joint work with Jonathan Lennox, Gautam Nair, Ping Pan, Jonathan Rosenberg, Kundan Singh, Elin Wedlund, Xin Wang, and Jianqi Yin)

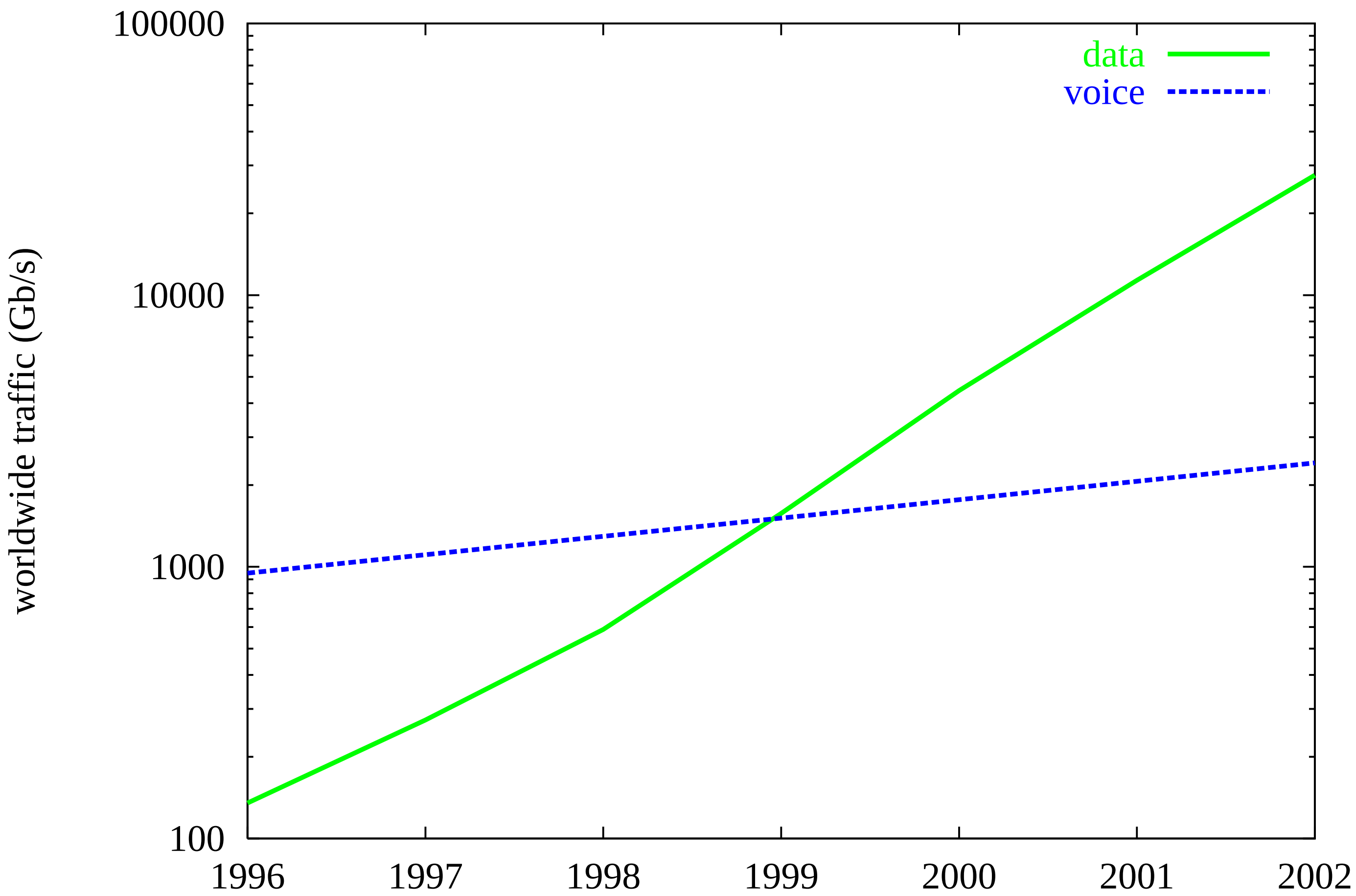
Overview

- architecture
- programmable services
- Internet phone appliances aka Internet phones
- standards-based unified messaging

Historical perspective

1876	invention of telephone
1915	first transcontinental telephone (NY–SF)
1920’s	first automatic switches
1956	TAT-1 transatlantic cable (35 lines)
1962	digital transmission (T1)
1965	1ESS analog switch
1977	4ESS digital switch
1980s	Signaling System #7 (out-of-band)

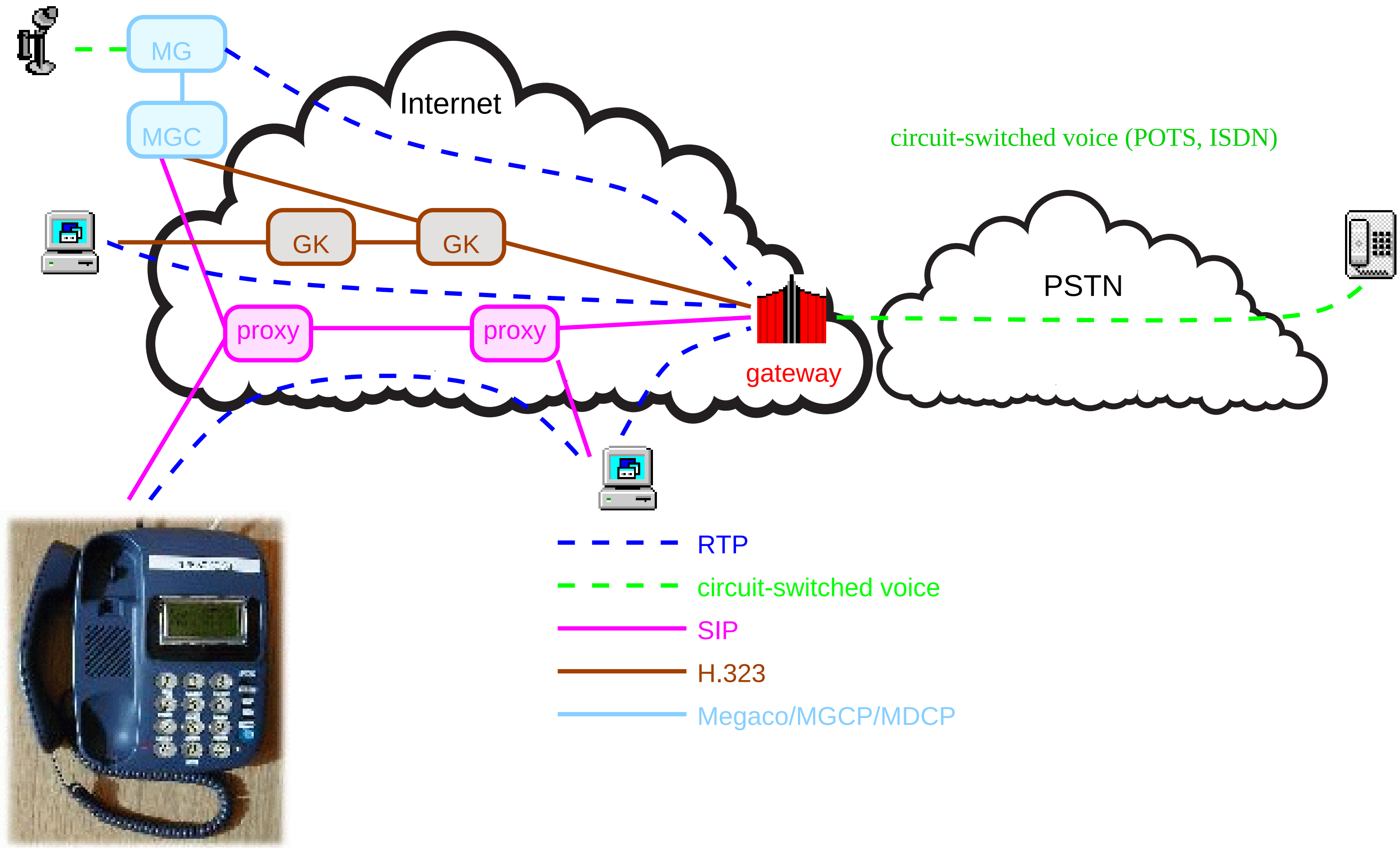
Data vs. Voice Traffic



Differences: Internet telephony $\leftrightarrow$ POTS

- separate control, transport (UDP) ➠ no triangle routing
- separate connectivity from resource availability
- separate services from bit transport
- datagram service ➠ less bootstrapping
- in-band signaling ➠ higher speed
- features "network" $\rightarrow$ end system: distinctive ringing, caller id, speed dialing, number translation, ... ➠ scaling
- features: intra-PBX = inter-LATA and general
- protocols: user-network = network-network signaling

VoIP Architecture



Principal IETF VoIP Protocols

RTP/RTCP: data transport and QoS feedback

SIP: call setup

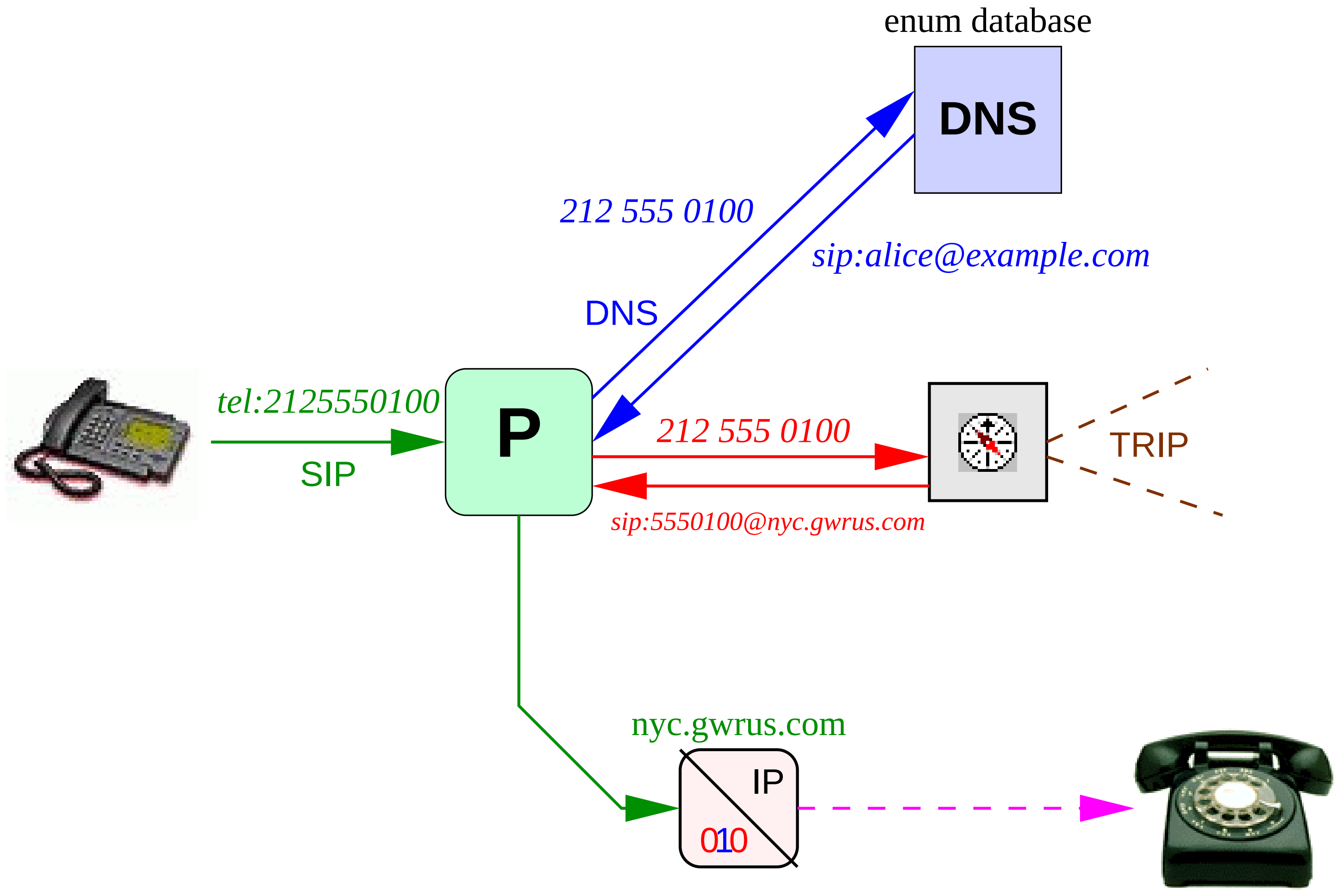
SDP: session/media description

enum: (DNS) E.164 $\longrightarrow$ URLs

TRIP: finding “cheap” PSTN gateways, BGP-like

RTSP: voice mail, announcements

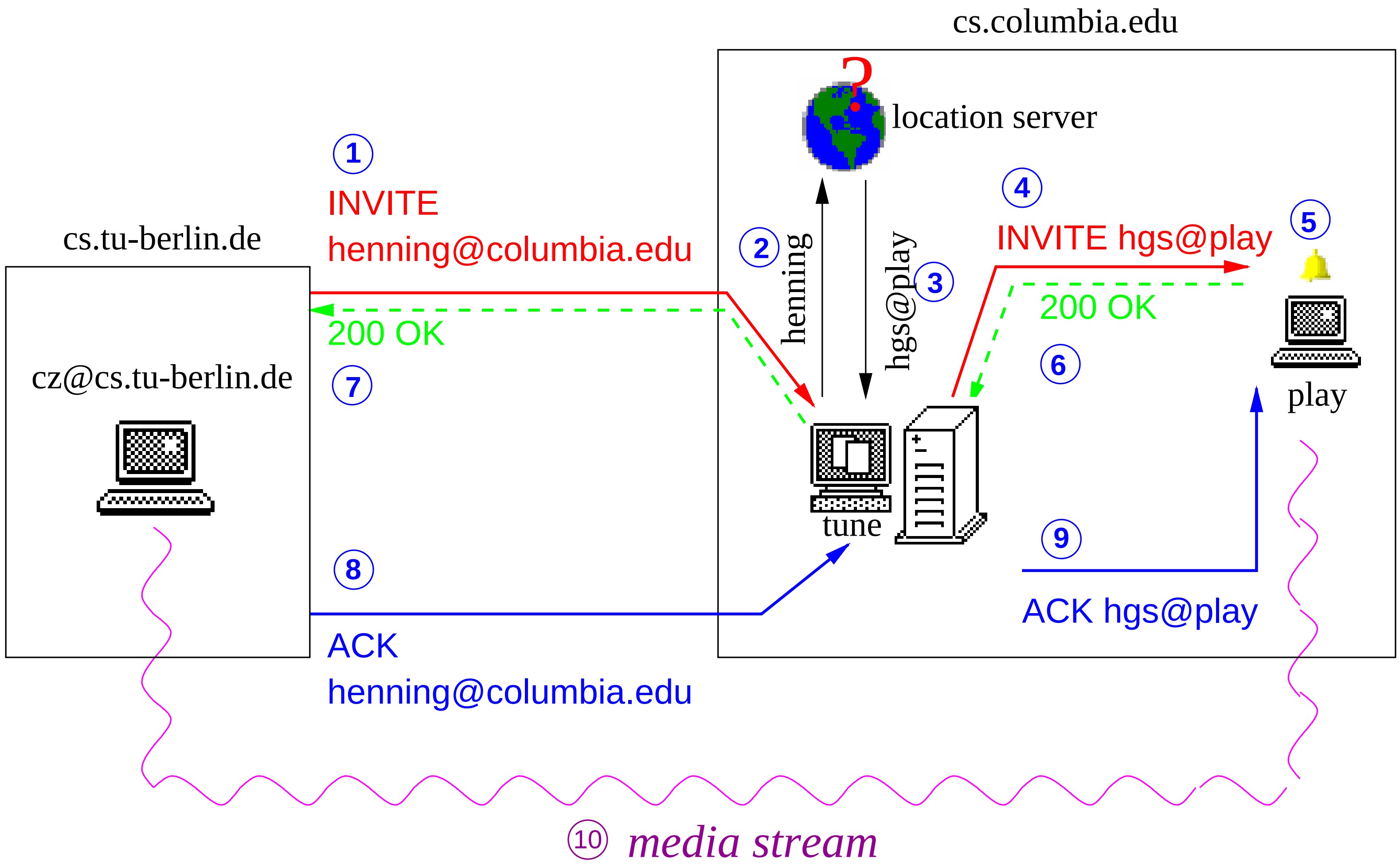
Number mappings



Session Initiation Protocol (SIP)

- call setup protocol
- support for user and terminal mobility
- genetically related to HTTP
- mechanisms: proxying (“forking”) and redirection
- multicast and unicast signaling
- caller preferences: “no voice mail, please”, “Spanish-speaking operator, please”
- establish security and QoS preconditions for call

SIP operation in proxy mode



Web and email integration

- invisible phone calls in VRML, web pages, avatars meeting, …
- use email address as universal communications identifier
- web page instead of “press 1 for hell, 2 for purgatory”
- forward calls to instant messaging, email, web page, …
- integration with calendar (vCal), address book (LDAP), …

Programmable services

- fixed service menu $\longrightarrow$ programs
- equipment vendor $\longrightarrow$ administrator, user, service providers
- several models:
  - APIs (Parlay, Jain)
  - applets
  - **sip-cgi**
  - **dedicated languages**: CPL
  - mobile code

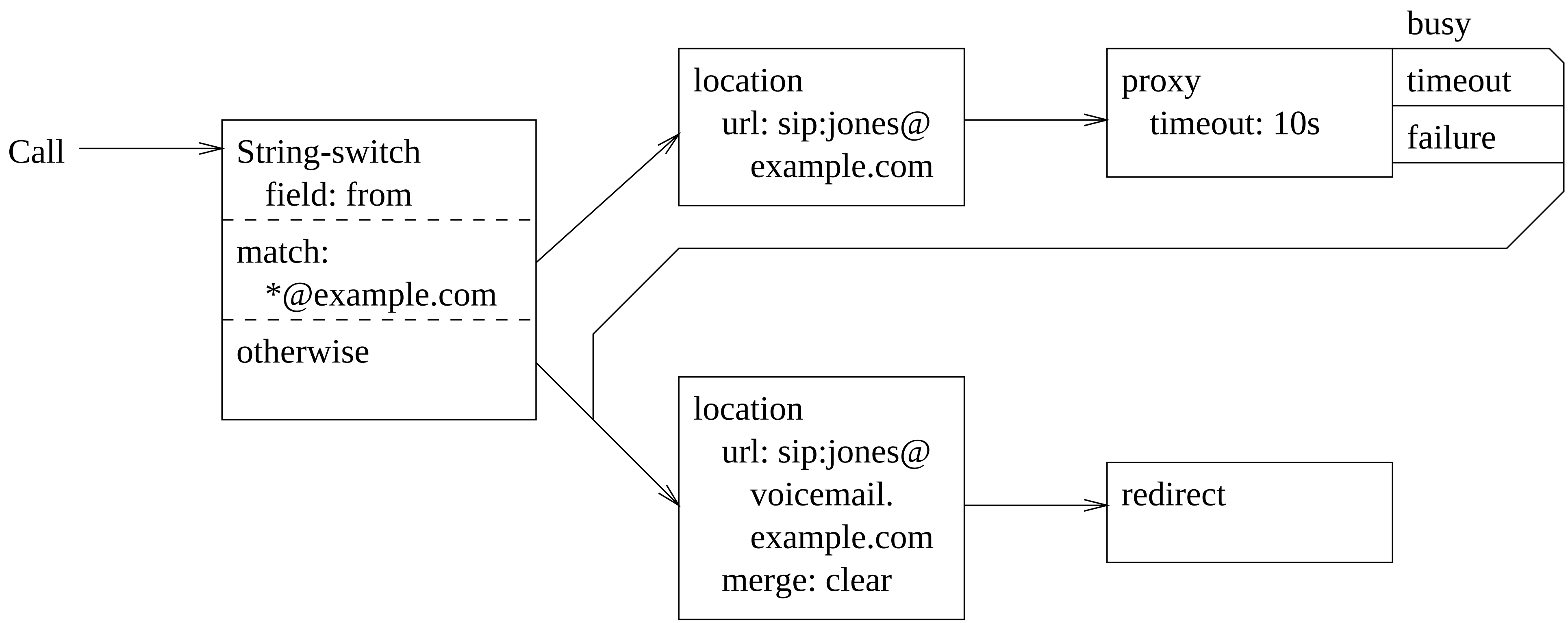
sip-cgi

- similar in spirit to cgi-bin scripts for web servers
- full access to all signaling functionality
- language-independent, typically scripting (Perl, Tcl, . . .)
- uses environment variables and stdin/stdout to communicate
- *reasonably* safe, but not for casual user

CPL

- safe: bounded run-time, no system access, provable
- creatable and editable by simple graphical tools
- independent of signalling protocol
- XML-based language, but not usually visible by user
- composable from building blocks
- minimize feature interaction by explicit specification

CPL example


Call
String-switch
field: from
match:
*@example.com
otherwise
location
url: sip:jones@
example.com
proxy
timeout: 10s
busy
timeout
failure
location
url: sip:jones@
voicemail.
example.com
merge: clear
redirect

CPL example

```
<subaction id="voicemail">
  <location url="sip:jones@voicemail.example.com">
    <redirect />
  </location>
</subaction>
<incoming>
  <address-switch field="origin" subfield="host">
    <address subdomain-of="example.com">
      <location url="sip:jones@example.com">
        <proxy>
          <busy> <sub ref="voicemail" /> </busy>
          <noanswer> <sub ref="voicemail" /> </noanswer>
          <failure> <sub ref="voicemail" /> </failure>
        </proxy>
      </location>
    </address>
    <otherwise> <sub ref="voicemail" /> </otherwise>
  </address-switch>
</incoming>
```

Internet phone “appliance”

- *Ethernet phone* ➠ no PBX for switching
- only DSP for voice coding and signaling ➠ limited memory
- minimal IP stack (IP, UDP, RTP, DHCP, SIP, DNS, IGMP)
- downloadable software (tftp)
- no TCP needed
- multicast & MP3 radio
- must be self-configuring
- personalize by user identification (i-button)
- interface to the physical world

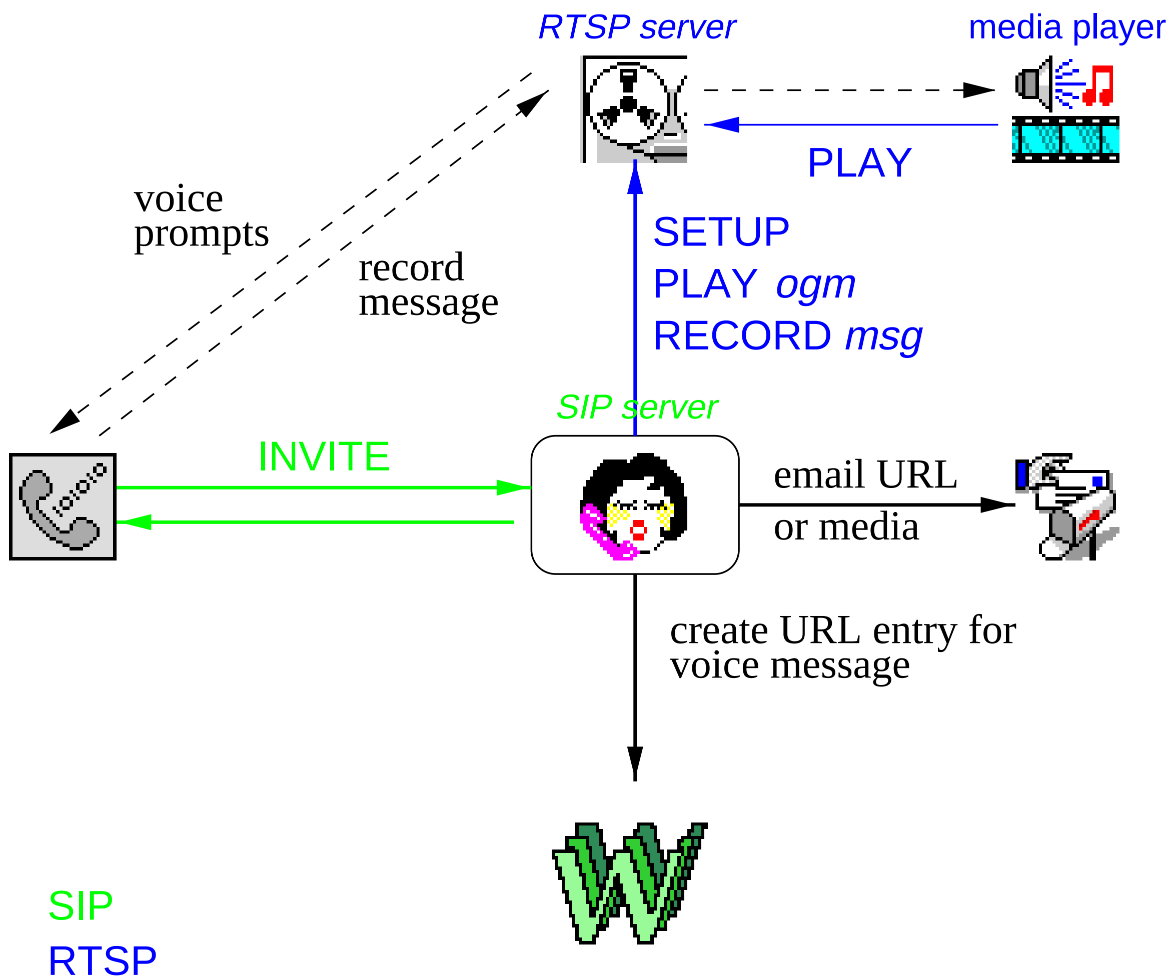
e*phone


128.59.19.38
D IDL 00:01:46 P
Jan 01 1900
SIP Phone 2.0_
NEW CALL
OPTIONS
MARK CALL
CALL LIST
CALL BACK
REMOVE
PROGRAM
FLASH
MUTE
HOLD
REDIAL
VOLUME
SPEAKER

Unified messaging

- voice/video mail to email or single web view
- deliver v/vmail as attachment: slow, particularly for video
- SIP VM server registers (and waits) or CPL sequential proxying
- use RTSP for generating OGM
- use email-delivered RTSP URL for streaming (e.g., QuickTime, RealPlayer G2)
- initial implementation with web interface

Unified messaging



Signaling and event notification

- traditional signaling: probe for availability
- event notification: presence, alarms, "beanie auction in progress", ...
- SIP extensions via SUBSCRIBE and NOTIFY
- allows forking of events and subscriptions
- unify recording and filtering

Resource reservation - another look

in-band: YESSIR $\longrightarrow$ use RTP to reserve resources

scaling: BGRP for sink trees

pricing: RNAP for congestion-adaptively priced reservations

Make money fast...

- `dialpad`: add-financed
- ad impression = 3.6c $\longrightarrow$ one add/minute
- pay for services: telemarketer filtering, voicemail, mobility – but: free part of portals?
- high-bandwidth reservations

Conclusion

- major protocol pieces in place
- operational issues: 911, billing, OSS for services, ...
- not just replicating existing architecture and service
- programmability key to web success
- should become an invisible service

More information

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

Papers: http://www.cs.columbia.edu/~hgs/research/irt

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

RTSP: http://www.cs.columbia.edu/~hgs/rtsp

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