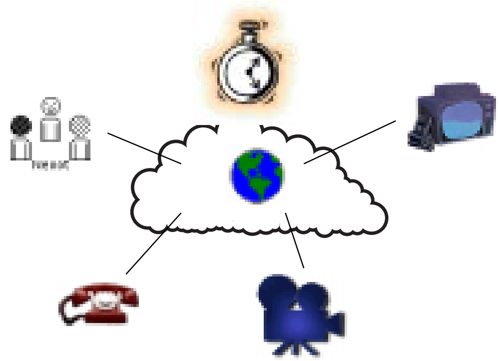
Internet telephony or what's hard about replacing 600 million telephones

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



NAE *Frontiers of Engineering Symposium*, Bremen, April 13-15, 2000

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

Overview

- what is Internet telephony?
- why replace the existing phone system?
- components of Internet telephony
- differences between IP telephony and traditional telephony
- quality of service
- programmability
- reliability

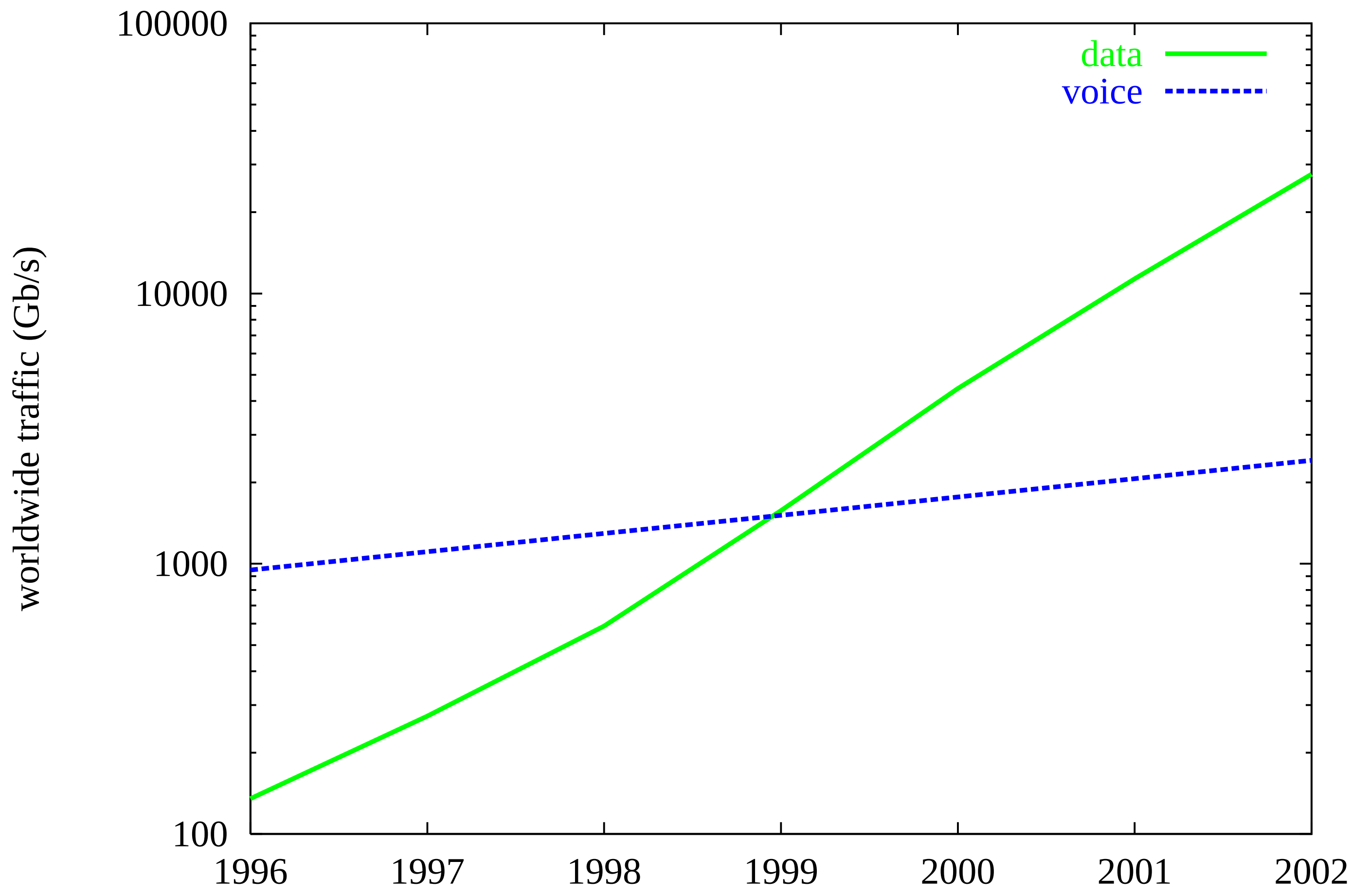
Historical perspective

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
1974	real-time packet voice (USC/ISI and MIT/L)
1977	4ESS digital switch
1980s	Signaling System #7 (out-of-band)
1991	DARTnet voice experiments
1992	first IETF audiocast

Internet telephony

- Internet telephony = use of Internet technologies to provide telephony services
- can use "public" Internet, LANs or intranets
- also called Voice-over-IP, although video and application sharing are included
- examples: Microsoft NetMeeting, `dialpad.com`

Data vs. Voice Traffic



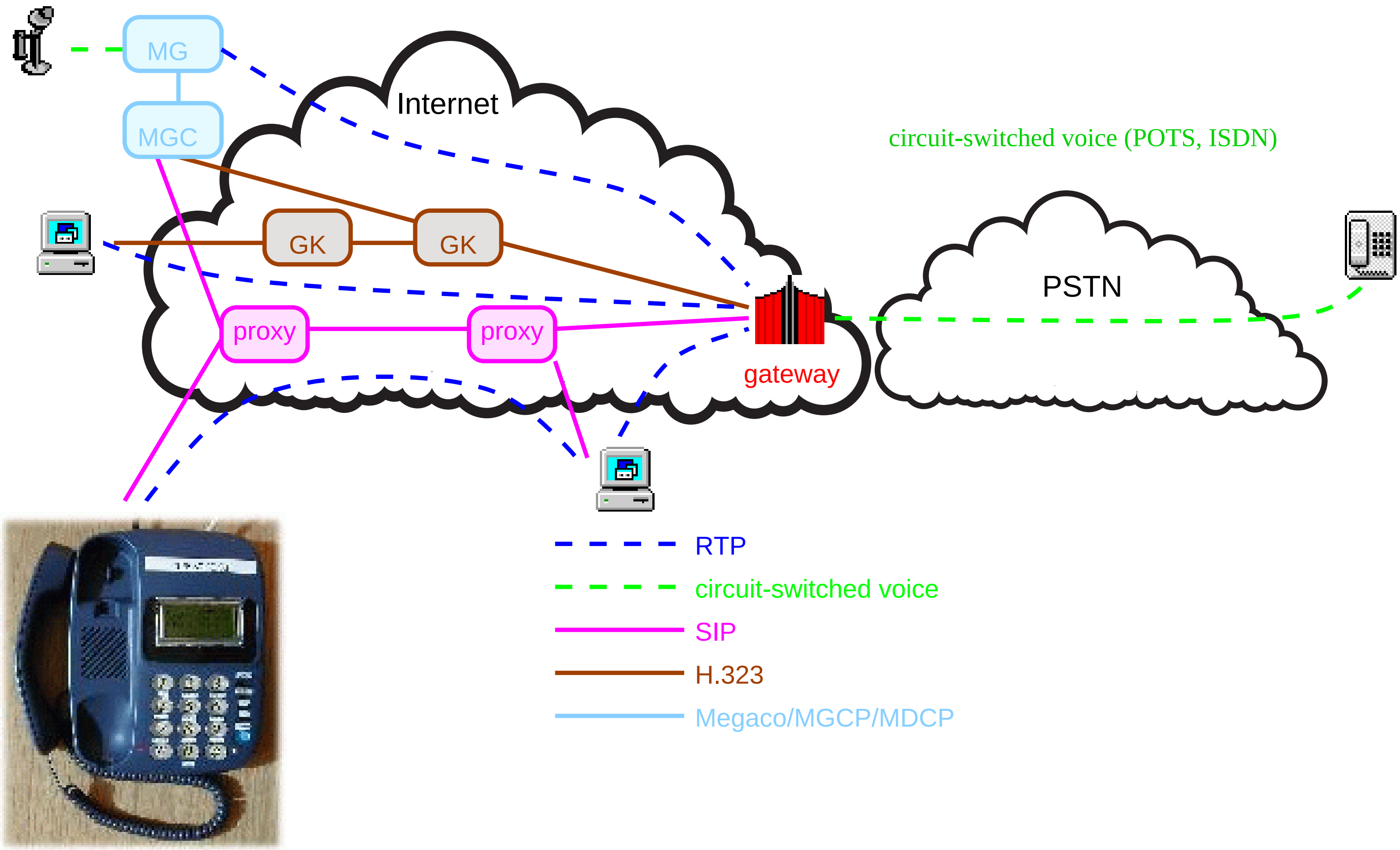
The phone works — why bother with VoIP?

user perspective	carrier perspective
tin can to broadcast quality	silence suppression ➠ traffic $\downarrow$
security through encryption	in-band signaling ➠ higher speed
caller, talker identification	shared facilities ➠ management, redundancy
better user interface	advanced services
TAT cable = \$0.03/hr	cheaper switching (9c vs. \$100s)
no local access fees (3.4c)	fax as data
no address scarcity	
programmability	
end-system capability labeling	
easy: video, whiteboard, . . .	

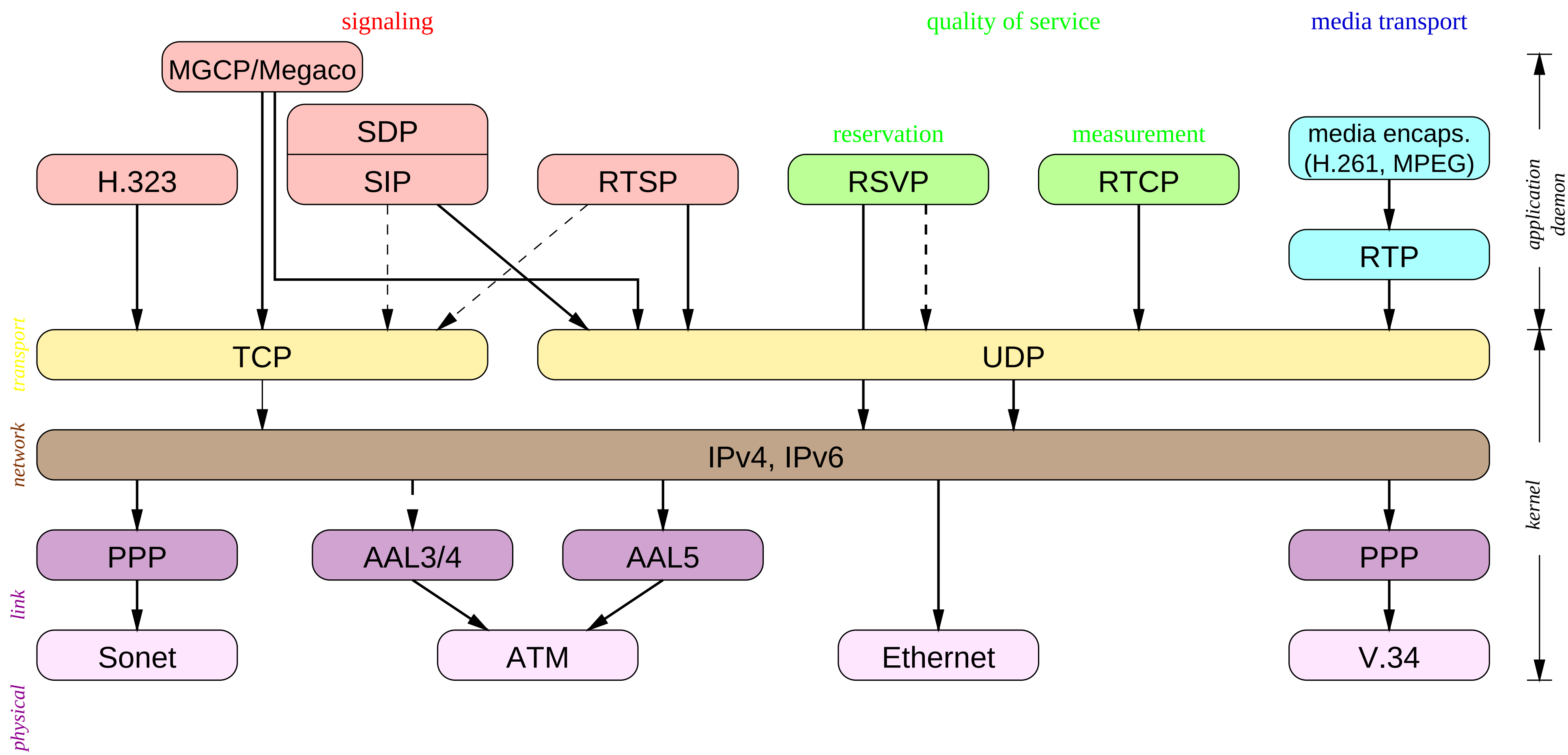
Differences: Internet telephony $\leftrightarrow$ POTS

- separate control, transport (UDP) ➠ no triangle routing
- separate connectivity from resource availability
- separate services from bit transport
- trust model
- physical location of end system?
- 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



Internet multimedia protocol stack



Telephone vs. Radio and TV

- now: separate infrastructure, technologies, regulation, . . .
- Internet: mostly the same, except

VoIP	media on demand	radio/TV
small groups	small groups	IP multicast “channels”
invitation, “ringing”	VCR commands	invitation by third parties
database	web page	web page, mc announcement
real-time	near real-time	delay tolerant
usually private	private or public	mostly public

- many hybrids: distributed couch, lectures, . . .
- longer term: single shared transport, wired and wireless

PSTN legacies to avoid

- telephone numbers $\longrightarrow$ email addresses as universal communications identifier?
- tones (e.g., failure indications)
- in-band signaling (“touch tones”)
- voice-only orientation
- integration of bit transport and services

$\longrightarrow$ confine PSTN knowledge to edge of network

Invisible Internet telephony

- currently: stand-alone application or PSTN phone
- chat applications
- distributed games
- virtual reality environments
- web pages and applets
- links in email messages

Principal IETF VoIP Protocols

RTP/RTCP: data transport and QoS feedback

SIP: call setup

SDP: session/media description

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

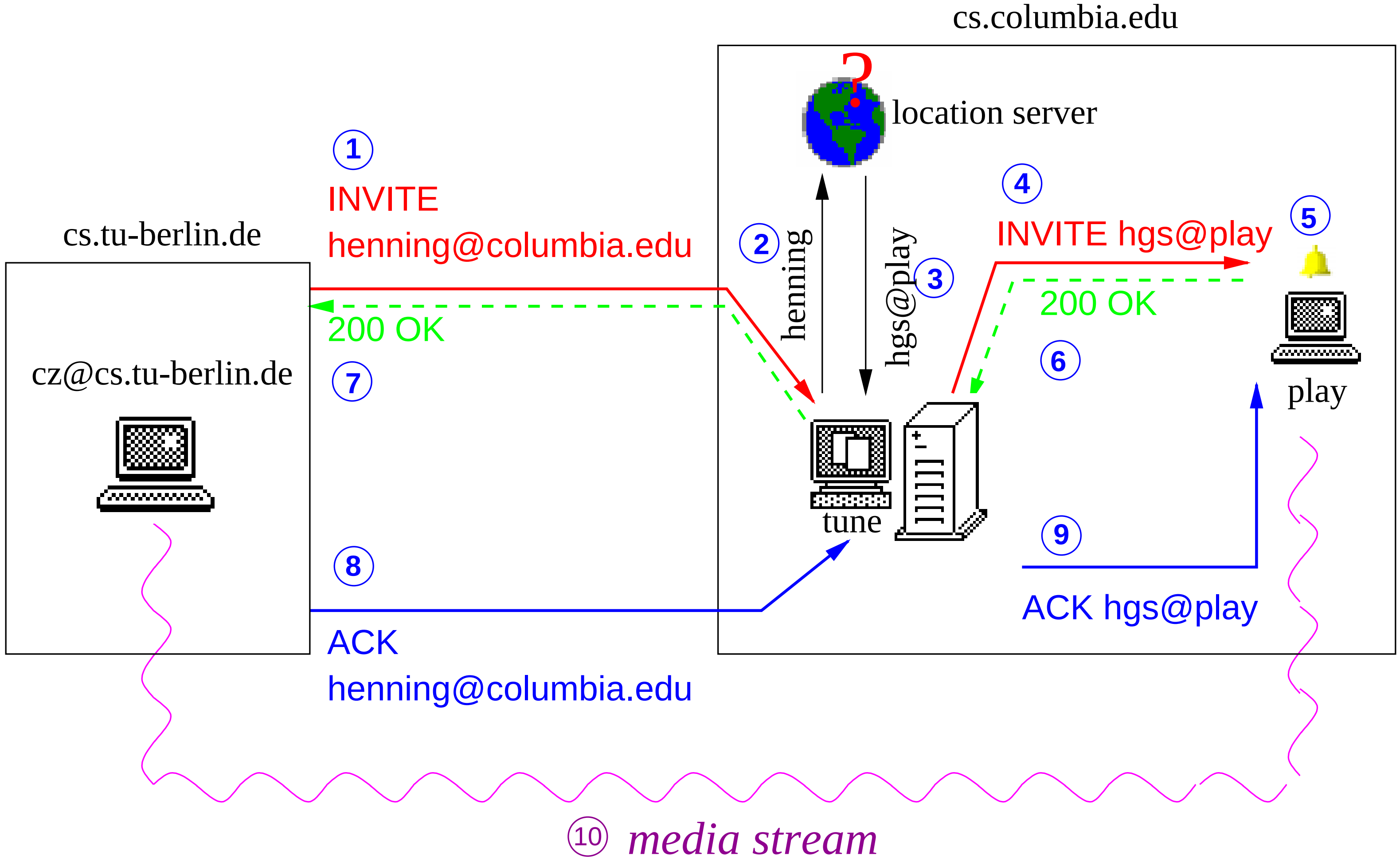
TRIP: finding “cheap” PSTN gateways, BGP-like

RTSP: voice mail, announcements

Session Initiation Protocol (SIP)

- call setup protocol
- support for user and terminal mobility
- genetically related to HTTP
- mechanisms: proxying (“forking”) and redirection
- feature negotiation mechanisms
- 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



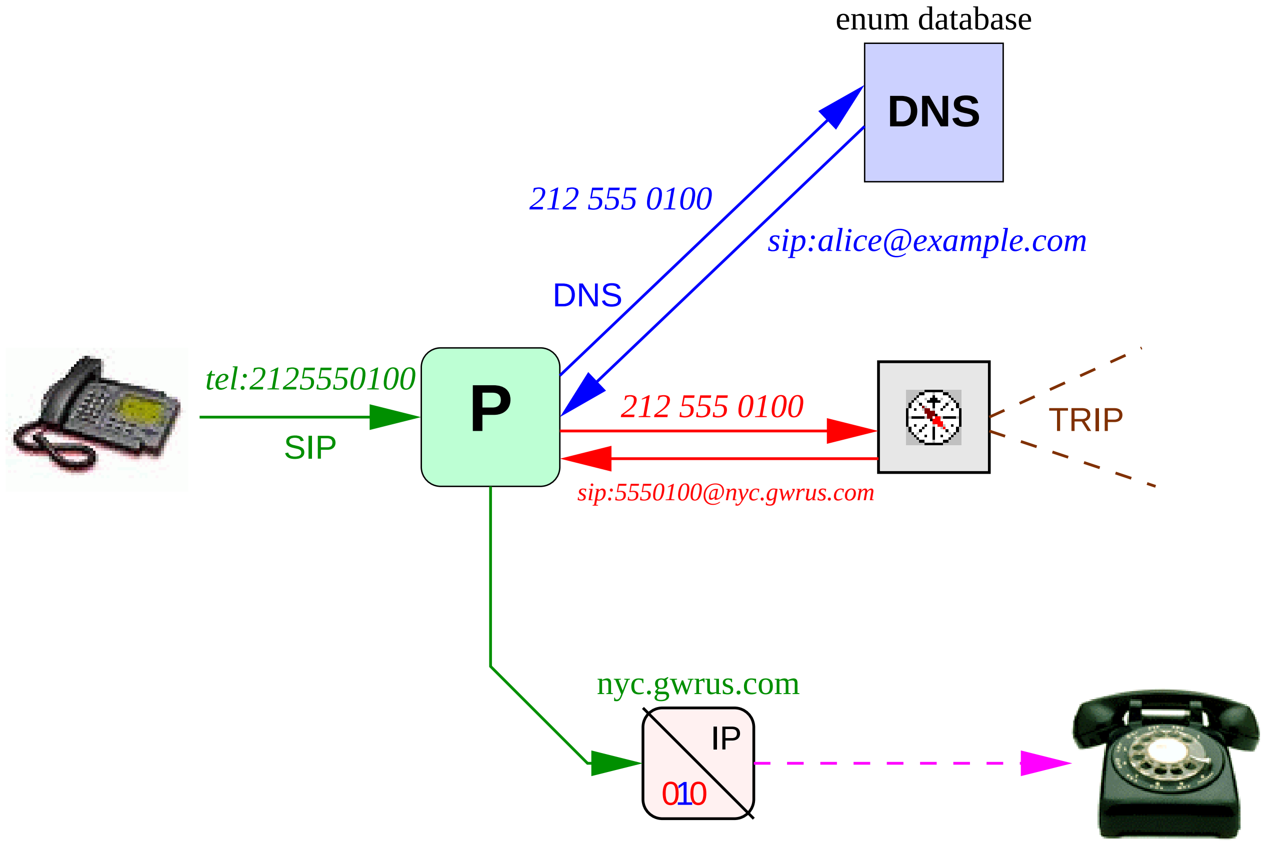
Integrating VoIP with the web

Everything linked together:

- telephone URLs: `tel:+1-212-555-0100`
- email: send SIP via email, redirect calls to email
- web: links to and actual content (HTML, XML, audio clips, . . .)
- chat and presence
- media streaming

Calling legacy phones

Internet telephony gateways – mostly local numbers



Charging model

- can't replicate existing $/minute PSTN model
- abolishes service monopoly by bit provider
- variable bit rate, not necessarily reserved
- service-independent to avoid masquerading
- advertising supported: 0.6 to 6 US cents/impression
- fixed charges or congestion-adaptive?

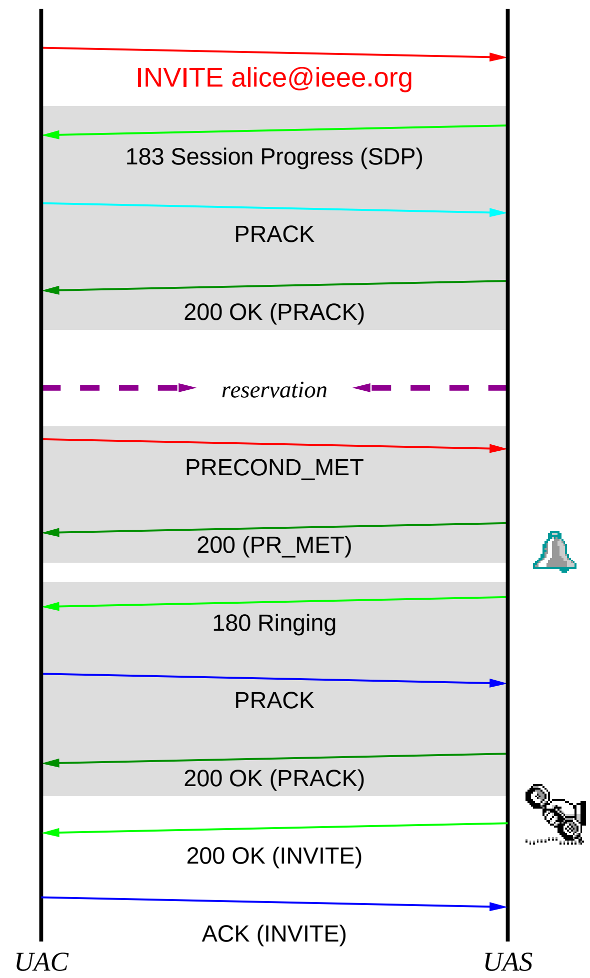
Quality of service

		admission state	
		flow	class
scheduling	flow	IntServ	doesn't make sense
state	class	ietf-diffserv-rsvp, BGRP	DiffServ

- best effort → classes → classes with reservation → adaptive reservations → fixed per-flow reservation
- modest gain for QoS routing
- connection-oriented Internet through back door?

Coupling of signaling and QoS

- traditional (H.323) approach: use signaling to set up QoS
- but: separation of signaling and data flow
- SIP approach: security and QoS *preconditions*



Reliability

- need “5 nines” reliability = 5 minutes/year
- currently have maybe 99.5%
- reasons: protocol design?
- lots of independent entities for DNS, routing, servers, OS, . . .
- lack of in-service software upgrades
- configuration problems

Feature interaction

- amateur feature designers
- cooperative and adversarial interactions
- request forking (voice mail)
- camp-on and call forward on busy
- outgoing call screening and call forwarding
- incoming call screening and polymorphic identity
- incoming call screening and anonymity

Internet phone "appliances"

- need small, cheap end systems (cf. PBX: $550/seat)
- *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

Mobile Internet telephony

- user and terminal mobility are related
- mobile applications: mostly UDP (DNS, multicast) or short TCP transactions (SMTP, POP, IMAP)
- should make applications restartable
- little mobile-IP deployment
- use SIP to support mobile multimedia applications
- mobile IP and SIP mobility are complementary

Programmable services

- fixed service menu $\longrightarrow$ programs
- equipment vendor $\longrightarrow$ administrator, user, service providers
- several models:
  - APIs (Parlay, Jain)
  - SIP servlets
  - **sip-cgi**
  - **dedicated languages**: CPL
  - mobile code
- related to active networks and agents

Sample services

- voice mail on busy/no answer
- intelligent user location
- call routing based on caller's language
- consult telemarketer database and reject
- only allow call-backs from those we have called before
- calendar – "please try again after 3 pm"

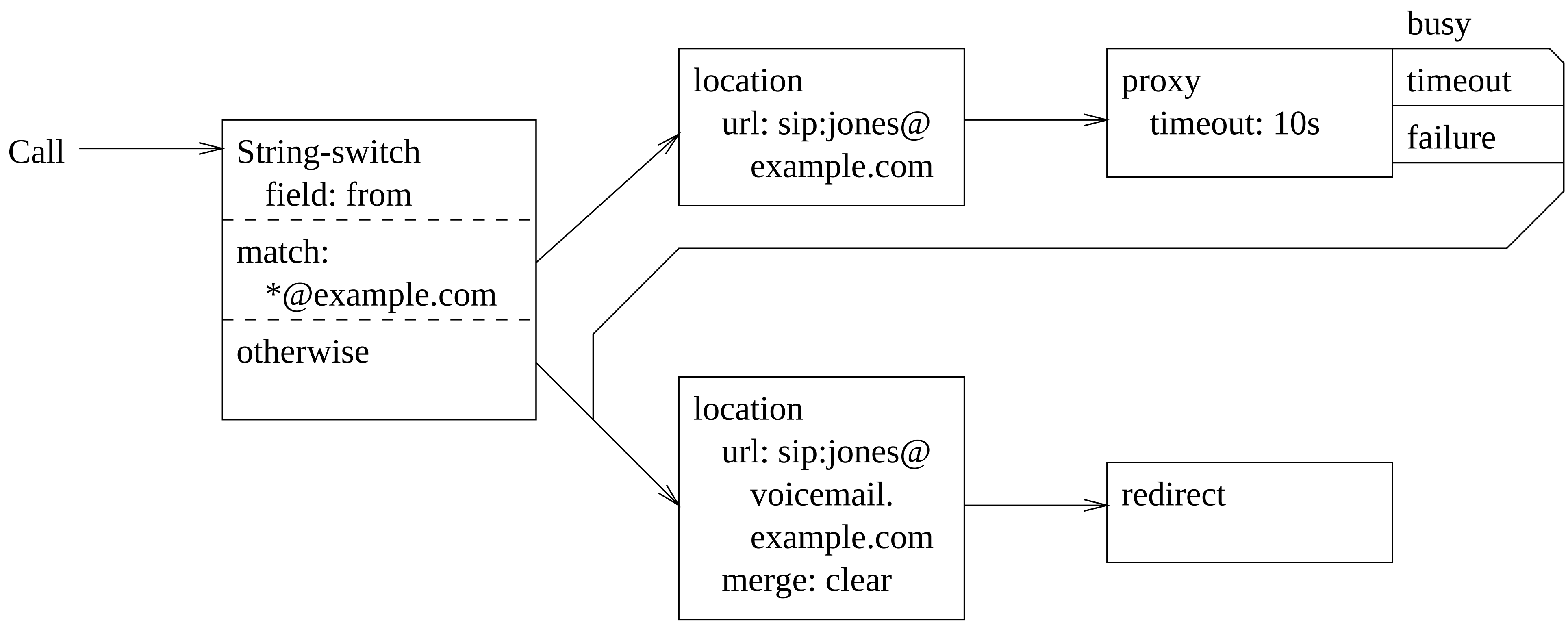
sip-cgi

- 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


busy
Call
String-switch
field: from
match:
*@example.com
otherwise
location
url: sip:jones@
example.com
proxy
timeout: 10s
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>
```

Signaling and event notification

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

Conclusion

- major protocol pieces in place
- operational issues: “911”, anonymity, billing, OSS for services, . . .
- not just replicating existing architecture and service
- programmability key – but how to make grandma a programmer?
- should become an invisible service
- need to keep low-end devices in mind