

Internet Telephony — The second chance to invent the telephone

Henning Schulzrinne
Internet Real-Time Lab
Dept. of Computer Science
Columbia University
New York, New York
schulzrinne@cs.columbia.edu

Columbia University CS Open House

November 16, 1999

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

Overview

- Internet telephony: motivation and problems
- Internet telephony “appliances”
- Programming your telephone (service)
- Mobile services

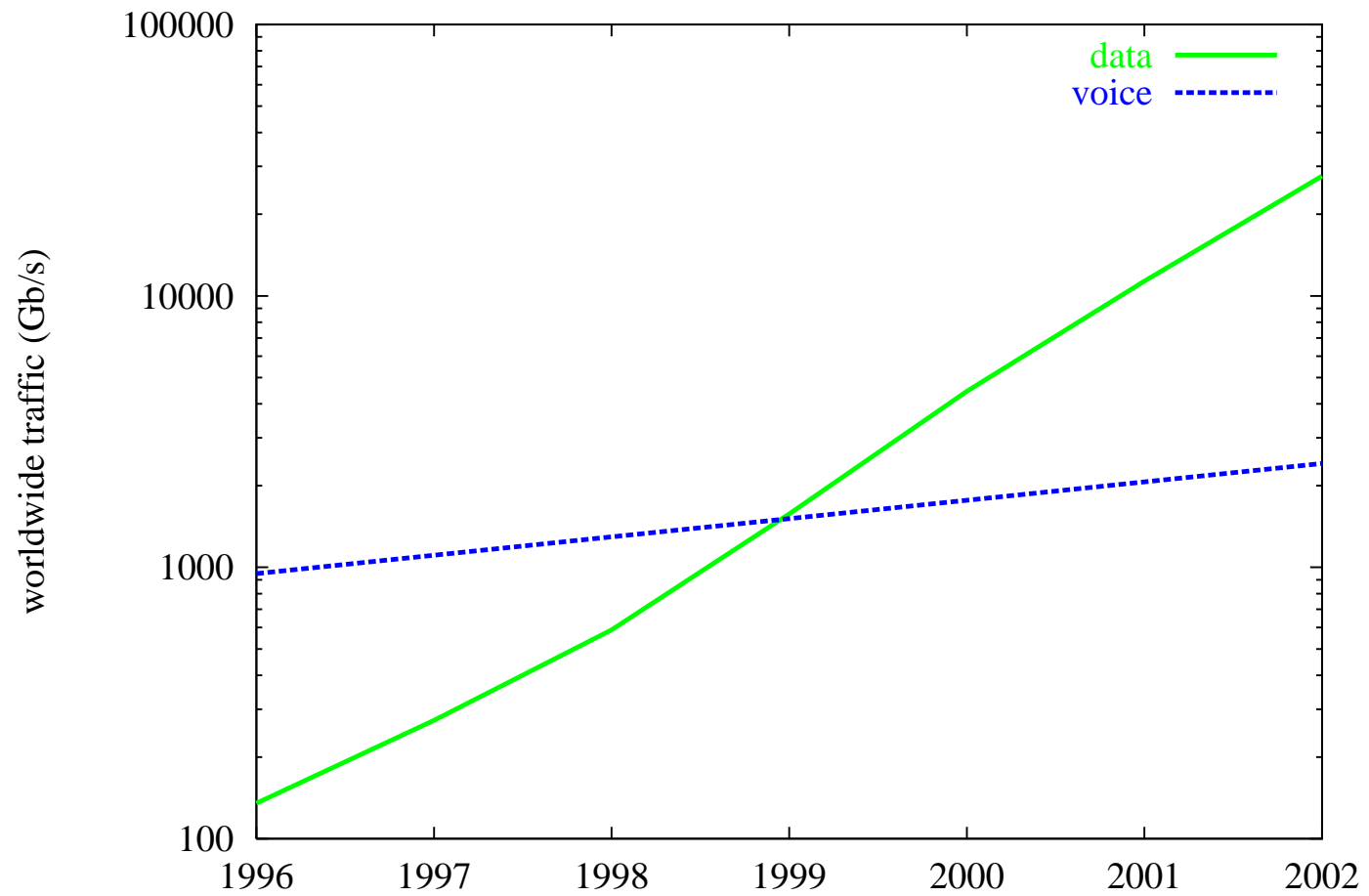
The phone works — why bother with VoIP?

- higher bandwidth efficiency
- programmable services
- multimedia (chess included), multicast
- cheaper switching
- unified messaging: voice mail → email
- only one network to manage
- single “phone number”: your email address?
- secure calls

mainframe → PC — CO switch → IP telephone

TV camera → camcorder

Data vs. Voice Traffic



Components of the New Phone Network

“telephones”: PCs, “Internet appliances”, Internet phones

“central offices”: routers, servers

gateways: to the legacy phone system

protocols: allow setting up and directing phone calls (SIP);
transport multimedia data (RTP); assure quality of service

Internet phone “appliance”

- phone = \$49.99; PC > \$600 (GPF included)
- *Ethernet phone* \Rightarrow no PBX for switching
- uses Session Initiation Protocol (SIP), Internet-standard for signaling
- MP3 radio
- interface to the world



Programmable phone service

- web = static pages → cgi-bin → Java
- “if calendar has meeting scheduled, don’t ring (unless it’s boss)”
- “if somebody is trying to call for the 3rd time, allow mobile”
- “try office and lab in parallel, if that fails, try home”
- “allow call to mobile if I’ve talked to person before”
- “if on telemarketing list, forward to dial-a-joke”
- phone computer-telephony integration = complex, not generally for end users
- “cgi-bin” for Internet telephones: allow scripts in end systems

“Active Phone Networks”

Develop a *service-creation* language:

- don't want Turing-complete language
- fail safe: make phone calls even if crashes
- predictable resource consumption
- hide parallelism (searches)
- hide timers
- execute in server or end system (or phone button)

▣▣▣▣▶ CPL, an XML-based language

Internet cellular phone

- centralized wireless carriers → every home a base station
- → no towers, low power, high bandwidth
- separate wireless protocols → integrated system with SIP for mobility

Open Issues

- Internet 911?
- reliability?
- integration of instant messaging
- service creation: just like a personal home page?

Conclusion

- transition from analog → digital → packet-switched
- “every telephone is a radio/TV station”
- we provide the infrastructure: signaling & transport protocols, programming languages, prototypes
- all packets, most of the time, in ten years?

More information

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

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

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