Internet Telephony: More than just re-inventing the telephone

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

University of Maryland

December 2, 1999

(Joint work with Ellen Hahne, Jonathan Lennox, Gautam Nair, Maria Papadopouli, Ping Pan, Lili Qiu, Jonathan Rosenberg, Xin Wang, Elin Wedlund, and Jianqi Yin)
Overview

- Internet telephony: motivation and problems
- protocol architecture
- quality of service:
  - light-weight resource reservation
  - forward error control
- services ➤ signaling
- the “programmable” telephone
- Internet telephony “appliances”
- mobile services
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)
1990s Advanced Intelligent Network (AIN) services deployed
Data vs. Voice Traffic

worldwide traffic (Gb/s)


100  1000  10000  100000

1000

data

voice
## The phone works — why bother with VoIP?

<table>
<thead>
<tr>
<th>user perspective</th>
<th>carrier perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>• variable compression: tin can to broadcast quality</td>
<td>• silence suppression $\Rightarrow$ traffic $\downarrow$</td>
</tr>
<tr>
<td>• security through encryption</td>
<td>• shared facilities $\Rightarrow$ management, redundancy</td>
</tr>
<tr>
<td>• caller, talker identification</td>
<td>• advanced services (email/web integration)</td>
</tr>
<tr>
<td>• better user interface (browser)</td>
<td>• cheaper switching ($0.005 \text{ vs. } $5/kb/s)</td>
</tr>
<tr>
<td>• internat. calls: TAT transatlantic cable = $0.03/hr</td>
<td>• fax uses 9.6 kb/s of 64 kb/s line</td>
</tr>
<tr>
<td>• no local access fees (3.4c)</td>
<td></td>
</tr>
<tr>
<td>• easy: video, whiteboard, …</td>
<td></td>
</tr>
</tbody>
</table>
Internet multimedia protocol stack

- **signaling**
  - H.323
  - SIP
  - RTSP

- **quality of service**
  - RSVP
  - RTCP

- **media transport**
  - media encaps. (H.261, MPEG)
  - RTP

- **transport**
  - TCP
  - UDP

- **network**
  - IPv4, IPv6

- **link**
  - PPP
  - AAL3/4
  - AAL5

- **physical**
  - Sonet
  - ATM
  - Ethernet
  - V.34

December 1, 1999
YESSION: Yet another Sender Session Internet Reservation

- RSVP: separate daemon, API
- integrate into application that needs it (embedded systems!)
- in-band easier firewall
- RTP: common data transport protocol for audio/video
- router alert option in RTCP packets
- resource demands: payload type, measurement, flow specs, ...
- soft-state + RTCP BYE
- partial reservations: add links as session ages ↔ fragmentation

December 1, 1999
**YESSIR**

**plain RTCP SRs or additional information:**

<table>
<thead>
<tr>
<th>IP Header with Router-Alert Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP Header</td>
</tr>
<tr>
<td>RTCP message:</td>
</tr>
<tr>
<td>Sender Report:</td>
</tr>
<tr>
<td>- sender information</td>
</tr>
<tr>
<td>- detailed report for each source</td>
</tr>
<tr>
<td>YESSIR message:</td>
</tr>
<tr>
<td>- reservation command: active/passive</td>
</tr>
<tr>
<td>- reservation style, refresh interval</td>
</tr>
<tr>
<td>- reservation flow specification</td>
</tr>
<tr>
<td>- link resource collection</td>
</tr>
<tr>
<td>- reservation failure report</td>
</tr>
<tr>
<td>Profile-specific extensions</td>
</tr>
</tbody>
</table>

**end-to-end refresh (vs. hop-by-hop)**
## RSVP and YESSIR performance

<table>
<thead>
<tr>
<th></th>
<th>setup</th>
<th>refresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>μs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSVP</td>
<td>1,105</td>
<td>624</td>
</tr>
<tr>
<td>YESSIR</td>
<td>356</td>
<td>344</td>
</tr>
</tbody>
</table>
Charging for Multimedia Services

- service models:

  **best-effort**: not predictable, all drop below threshold
  **adaptive, “TCP-friendly”**: no incentive to adjust
  **reserved**: long-lived connections $\rightarrow$ blocking $\uparrow$

  thus, we define *adaptive reserved services* with pricing incentives
RNAP: Architecture

Access Network

--- RNAP messages

<--- Intra-domain Messages

December 1, 1999
RNAP: Pricing

**holding cost:** opportunity cost for holding resources (= price of next-lower quality level)

**usage cost:** infrastructure cost amortized over usage

**congestion cost:** discouragement mechanism
RNAP

Adjustment with different utility functions:

![Bandwidth Reservation Graph]

- session1
- session2
- session3
- system
- affordable total bandwidth

December 1, 1999
Integrating packet FEC into adaptive voice playout buffers

- playout buffer: trade loss (2...20%) for delay (50...500 ms)

- \( (n, k) \) FEC: add \( n - k \) additional packets for total of \( n \)

- algorithms:
  - exponential average and fast exp-avg
  - minimum delay
  - spike delays
  - window: spike mode + \( q^{th} \) quantile
FEC: virtual delay

- virtual delay = min(arrival, recovery) – departure
- playout delay ≈ \( \alpha \cdot \sigma \)
- if loss < target loss, \( \alpha \leftarrow \alpha + \delta \)
- recover lost and late packets
- 20% loss: application loss/5, delay * 2
FEC loss and delay

Exp-avg vs. Its Extension

<table>
<thead>
<tr>
<th></th>
<th>Application Loss Probability</th>
<th>Average End to End Delay (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp-avg</td>
<td>0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22 0.24 0.26</td>
<td></td>
</tr>
<tr>
<td>Exp-avg (add (N-1)*pkt-length)</td>
<td>0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22 0.24 0.26</td>
<td></td>
</tr>
<tr>
<td>Exp-avg Ext</td>
<td>0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22 0.24 0.26</td>
<td></td>
</tr>
</tbody>
</table>
New playout algorithm: delayed optimal

- after talkspurt, one knows optimal delay
- use optimal combination for next talkspurt
- $D_i = \alpha D_{i-1} + (1 - \alpha) D_{opt}$
- user perception function: minimal delay that achieves loss target
New playout algorithm: Analytical

- assume independent loss $p$, delayed randomly $d$
- compute playout probability
- tabulate delay distribution histogram
- at end of talkspurt, find matching playout delay for target loss rate
Playout algorithm comparison

Trace 1

Application Loss Probability

Target Loss Probability

- Exp-avg Ext
- Spk-det Ext
- Window Ext
- Prev-opt (Bin)
- Analytical
- Optimal

December 1, 1999
SIP: Session Initiation Protocol

- call user
- re-negotiate call parameters
- manual and automatic forwarding
- call center: reach first (load distribution) or reach all (department conference)
- *personal mobility* (complements data link/IP mobility) → change of terminal (PC, digital cordless, palmtop), location
- “forking” of calls: one person, multiple locations
- terminate and transfer calls
- web security, cookies
SIP addresses food chain

yellow pages
common names
host-independent
host-specific
IP address

“president of the United States”
“Bill Clinton, Whitehouse”
president@whitehouse.gov
sip:bubba@oval.eop.gov
sip:+1-202-456-1111@net2ph.com
198.137.241.30
SIP: basic operation

1. use directory service (e.g., LDAP) to map name to \textit{user@domain}
2. locate SIP servers using DNS SRV, CNAME, A
3. called server may map name to \textit{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

1. INVITE henning@columbia.edu
2. 200 OK
3. INVITE hgs@play
4. 200 OK
5. play
6. ACK hgs@play
7. 200 OK
8. ACK henning@columbia.edu
9. "media stream"
SIP operation in redirect mode

1. INVITE henning@ieee.org
2. 302 Moved temporarily
   Contact: hgs@columbia.edu
3. ACK henning@ieee.org
4. INVITE hgs@columbia.edu
5. 200 OK
6. ACK hgs@columbia.edu

December 1, 1999
SIP protocol design: robustness

SIP is designed to be robust against server failures:

- no state in proxy servers during call
- responses are “selfrouting”
- subsequent requests and retransmissions can take different path (backup server)
- proxy servers can “lose memory” any time ⇒ still function
- UDP ⇒ less state than TCP, no time-wait
## Invitation modes

<table>
<thead>
<tr>
<th>invitation</th>
<th>conference</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>unicast</td>
<td>telephony</td>
<td>Internet TV session</td>
</tr>
<tr>
<td>multicast</td>
<td>reach first</td>
<td>dept. conference</td>
</tr>
</tbody>
</table>
SIP user location

- SIP is independent of mechanism to locate user
- examples:
  - local multicast of invitation
  - login-based via NFS
  - recursive “finger”-traversal
  - name translation: Alexander.G.Bell ➔ agb
  - active badges
- SIP:
  - REGISTER announces location, with time limit
  - REGISTER + Location sets new location
  - forwarding within host (≠ standard port)
Implementations

- 33 vendors at December 1999 SIP “bake-off”

- Columbia sipd:
  - registration via unicast and multicast
  - handles mailing lists (ug-students@cs), ambiguous names (lee@cs)
  - maps names (b.clinton@whitehouse)
  - Apache (httpd)-style configuration and logging
  - “basic” authentication
  - how many servers for 2300 requests/second?
Signaling ← event notification

- call queueing . . . buddy lists . . . event notification
- SUBSCRIBE to events
- server NOTIFY
- can use forking
- handle subscriptions using CPL
- transition to multicast if large group of subscribers
Programmable phone service

- “caller proposes, callee disposes, administrator decides”
- web = static pages → cgi-bin → Java
- “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: CTI = complex, not generally for end users
- “cgi-bin” for Internet telephones: generate requests, proxy, responses ▶️ sip-cgi, complete control
“Active Phone Networks”

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 callee’s proxy server or end system (or phone button)

▶ CPL, an XML-based language
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

December 1, 1999
CPL example

<call>
  <location url="sip:jones@jonespc.example.com">
    <proxy timeout="8s">
      <busy>
        <location url="sip:jones@voicemail.example.com" merge="clear" id="voicemail">
          <proxy />
        </location>
      </busy>
    </proxy>
  </location>
  <noanswer>
    <link ref="voicemail" /> 
  </noanswer>
</call>
e*phone, an Internet phone “appliance”

- phone = $49.99; PC > $600 (GPF included)
- *Ethernet phone* no PBX for switching
- minimal operating system: threads and event flags (CRTX, 2 kB)
- DSP for voice coding limited memory (128 kB SRAM!)
- implemented minimal IP stack (IP/UDP/RTP, DHCP, SIP)
- DNS and TCP not absolutely needed
- MP3 radio
- interface to the analog world: sensors, X10, ...
Internet cellular phone

- mobile IP: mask mobility to TCP connections
- imposes overhead:
  - all registrations to home agent
  - triangle routing (mostly)
  - encapsulation, address filtering problems
- use SIP and RTP for mobility management
SIP mobility

1. SIP INVITE
2. SIP 302 moved temporarily
3. SIP INVITE
4. SIP OK
5. data

- MH: mobile host
- CH: correspondent host
- Redir: SIP redirect server
Other work: signaling

- touch-tone transmission
- interoperation of SIP with SS7, ISDN and POTS
- large-scale IPtel gateways
- locating IPtel gateways (and other wide-area resources)
- charging for (adaptive) services and resources
- Internet voice mail
Internet mobile services

Global Network

WLAN

NoD

public area

Gateway

proxy

CDPD

Gateway

ADSL

PSTN

Ricochet

train

WaveLan

Internet Telephony
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

- Internet telephony = first new service since web
- last new/old service?
- touches QOS, signaling, programming
- deployment inside out or outside in?
- operational issues: billing, 911, CALEA, ...
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