SIP: Affordable Convenience

An Update on SIP Applicability in 2003

Jiri Kuthan, `iptel.org`
sip:jiri@iptel.org
May 2003

Outline

- Backgrounder: SIP and iptel.org's SIP operational experience
- SIP Motivation?
- Technical Aspects
  - What (does not) work(s) today
  - Operational Challenges and Our Answers to Them
- Conclusions

SIP Refresher

- IETF's signaling protocol (RFC3261)
- Primary use: VoIP Signaling
- Other uses: Instant Messages and Presence
- Other possible uses: video conferencing, home appliances, games, etc.
- Technical properties: HTTP-like, textual, client-server protocol, using email-like addresses
- Gory details: tutorial tommorrow

iptel.org Background

- Very originally, iptel.org was consultancy organization, part of Fraunhofer.
- iptel.org has been running SIP services on the public Internet since 2001. Users are able to pick an address username@iptel.org and a numerical alias.
- Mostly used applications: VoIP, instant messaging and presence, voicemail2email.
- The infrastructure serves public subscribers as well as internal users with additional privileges.
- Increase in population size since introduction of Windows Messenger.
- Services powered by iptel.org's open-source SIP server, SER.

Shall I spend time with SIP at all?

- Keep expectations well managed.
- Facts
  - Services attract and there are **convenience services** which are finding user adoption.
  - SIP infrastructure can be very **affordable** and power multiple services
- Vendor propaganda:
  - "There is infinite space for killer applications" – there is, but you do not get any killer application from the vendors.
  - Enormous number of application creation platforms (frequently Java-based), most of them missing the key property: Efficiency.

Convenience Services

- The service driver is convenience: don't let users get hands off their notebook and you will make them happy.
- Applications demanded and deployed are mostly about service integration:
  - E-mail: replacement of IVR annoyance with voicemail-2-e-mail
  - Web: read list of missed calls from your webpage (both off-line and on-line)
  - Web: online phonebook
  - Instant Messaging and Presence, Notification services (T-sturm alarm), SMS delivery
  - Telephony: conferencing (to be deployed at iptel)

Affordability

- Setting up SIP Infrastructure is inexpensive:
  - Open-source software available (SER: www.iptel.org/ser)
  - Server Hardware: SER can deliver hundreds of Calls per Second (CPS) on an IPAQ and thousands of CPS on a MediaMarkt PC
  - Cheap SIP phones underway (~ $80)
  - PSTN gateways: open-source PC solutions emerging, low-density gateways at ~ $1k
- SIP has been designed in a way that allows single infrastructure to deliver multiple services

Our Customers' Use Cases

- Most iptel.org customers from ISP crowd:
- IP telephony offered as a part of bundle services
- Residential users mostly attracted by cheap telephony (some give up their telco lines), instant messaging and presence
- Enterprise users mostly after effectivization tools.
- Most happines reported from ISPs charging for bandwidth utilization.

Technical Feasibility: Good News

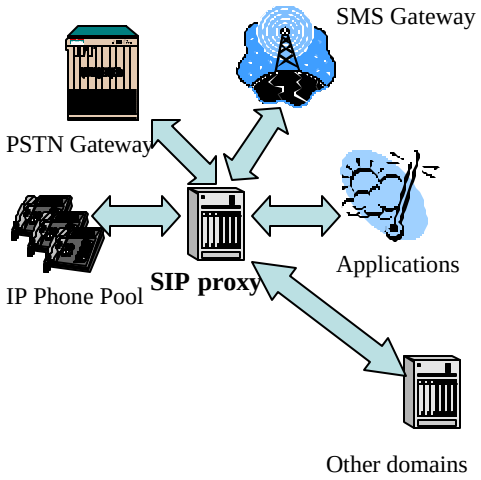
- Basic VoIP services work, so do complementary integrated services such as instant messaging, voicemail, etc.
- Billing machinery works too: Accounting easy, though not standardized. Gateways with accounting support exist today
- Numbering plans easy to maintain and they complement domain names well.
- QoS mostly pleasant.
- Solid SIP implementations interoperate fairly well.
- Interoperation with other technologies works too:
  - PSTN gateway market established (single-vendor dominance too)
  - Gateway to Jabber instant messaging and SMS up and running
  - Commercial H.323 gateways exist

Concern Stack

- Performance – are you really able to process all the crap messages you receive over the public Internet?
- Routing complexity – SIP is great in linking service component. Are you sure you linked them right?
- Application programming – is it easy?
- Nightmare – NATs.
- Why My Wife Doesn't Use VoIP: Reliability.

SIP Routing

- One of primary benefits of SIP: Ability to link various service components speaking SIP together.
- The "glue" are signaling servers. Their primary capability is routing requests to appropriate services.
- Issues:
  - Routing flexibility – how to determine right destination for a request
  - Troubleshooting when routing failures occur

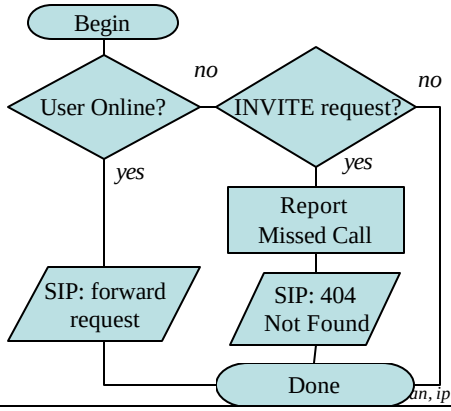


Routing Policy

- SIP request-routing decision can depend on a variety of factors. Iptel.org example:
  - address-based routing – requests to numeric destination are forwarded to PSTN gateway, whereas others to IP phones
  - Policy-based processing – calls to international PSTN requests require authentication and privileges
  - Method-based routing – requests to numerical destinations are split by method between SMS and PSTN gateway
  - Further factors include request's transport origin, address claimed in From header field, content of Contact, etc.
- **Operational observation: mighty tools for specification of routing policy are needed.**

Routing Language

- Request routing flexibility needed to link SIP components (voicemail, PSTN gateway, logging facility, etc.) together
- Answer: request routing language (features conditions, URI-rewriting, request modification, replying, etc.)
- Example: reporting missed calls



SER Routing Language

```
/* user online ? */
if (lookup("location")) {
     t_relay();
     break;
};
if (method=="INVITE") {
     /* report to syslog */
     log("ACC: missed call\n");
};
sl_send_reply("404","Not Found");
```


Performance Concerns

- New applications, like presence, are very talkative
  - Presence status update frequent
  - Each update ventilated to multiple parties
- Broken or misconfigured devices account for a fair part of load; few of many real-world observations:
  - Broken digest clients resend wrong credentials in an infinite loop → heavy flood
  - Mis-configured password: a phone attempted to re-register every ten minutes (factor 6) →2400 messages a day
  - Mis-configured Expires=30 (factor 120)
  - Keeping NAT bindings up – SIP request each 20 seconds
- Replication, Boot avalanches

Achievable Scalability

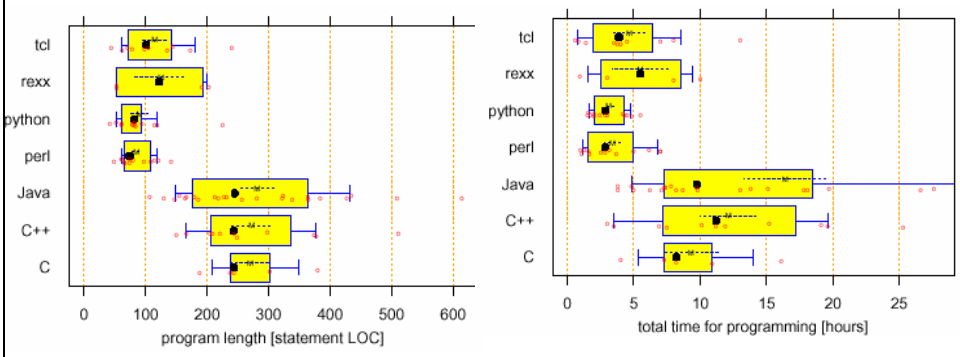
- Good news: well-designed SIP servers can cope with load in terms of thousands of calls per second (CPS)
  - *Example: lab-tuned version of SIP Express Router able to process 5000 Calls Per Second to a static destination statefuly on a dual-CPU PC – capacity needed by telephony signaling of Bay Area*
- Pending concern: denial of service attacks
  - *Example: hundreds of megabytes of RAM can be exhausted in tens of seconds with statefull processing*

Application Programming

- Status: vendor market dozens of APIs and service creation platforms
- Problem: most of these tools lack effectiveness: they are not simple, they are bound to specific languages and run-time environment and reuse of existing application uneasy
- Our approach: reuse of UN*X scripting legacy
- Results: we built a scripting interface which we use from both command-line tools and web pages:
  - Click-to-dial … 183 lines of code (bash), ~ the same for PHP
  - Weather alerts … 80 lines of code and reuse of a Linux weather application

Why Language Independence Matters

- No dependency on a particular programming language – developers can use what they best understand, including scripting languages
- Use of scripting languages makes code shorter and takes less time (graphs from [*] demonstrate complexity for a specific problem)



(*) Source of both graphs: Lutz Prechelt: "An Empirical Comparison of C, C++, Java, Perl, Python,RXX, and Tcl", March 2000

NAT Traversal

- NATs popular because they conserve IP address space and help residential users to save money charged for IP addresses.
- Problem: SIP does not work over NATs. Peer-to-peer applications' signaling gets broken by NATs: Receiver addresses announced in signaling are invalid out of NATted networks.
- Straight-forward solution: IPv6 – unclear when deployed if ever.
- There are many scenarios for which no single solution exists: STUN, ALGs, symmetric communication, media relay, UPnP, …

Murphy's Law Holds

Everything can go wrong.

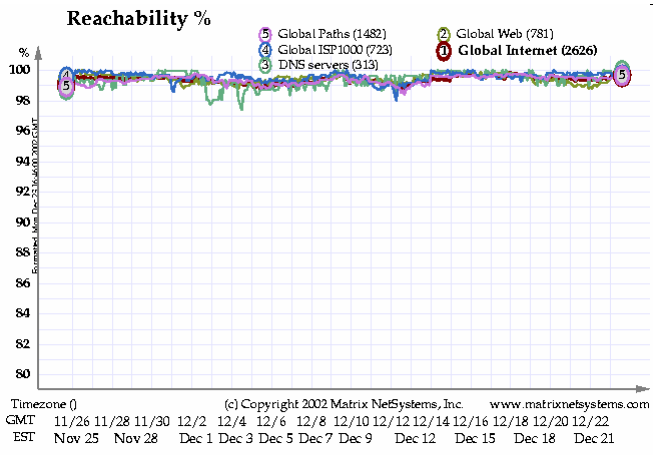
- Servers:
  - software/configuration upgrades
  - vulnerabilities
  - both SIP and supporting servers subject to failure: DNS, IP routing daemons
- Hosts:
  - power failures
  - hard-disk failures
- Networks:
  - line.
  - IP access

IP Availability: SLAs

- Industry averages for "Network Availability" SLAs are from 99.9% to 99.5% (an NRIC report)
- SLAs mostly exclude regular maintenance and always Acts of God
- Residential IP access rarely with SLAs

Availability (percent)	Actual Downtime (per year)
99.999	5 Minutes
99.9	9 Hours
99.5	1.8 Days

matrix.net's Reachability Statistics



- Minimum 98.69%
- Median 99.45%
- Maximum 99.84%
- Mean 99.40%

Reduncy Status

- *Replication of real-time data such as user location information*
  - doable as for today
- *Making clients use backup infrastructure on failure*
  - Doably in theory (specification) using DNS/SRV, only one SIP phone known today to get it right.
  - A variety of (cumbersome) work-arounds exist, mostly at IP layer: BGP manipulations, IP take-over

Concluding Observations

- Basic VoIP & complementary services up and running.
- Infrastructure can be set up in an inexpensive way
  - "commercial grade" solutions are propaganda of high-cost vendors (except legacy devices)
  - open-source solutions can handle large-scale installations (see SER).
- The operationally critical issues, performance and routing flexibility are delt with, at least with SER.
- Solutions for some compelling problems still in infancy: DoS, NAT traversal, solid fail-over (mostly phone vendor's guilt).

Information Resources

- Email: jiri@iptel.org
- IP Telephony Information: http://www.iptel.org/info/
- SIP Services: http://www.iptel.org/user/
- SIP Express Router: http://www.iptel.org/ser/