SIP Interoperability Test Events
(formerly known as SIP Bake-Offs)

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

Pulver SIP Summit – Richardson, TX
May 1st, 2001
Overview

- Goals
- What it is (and isn’t)
- History
- Traditions
- Capability levels
- Preparing for an interoperability event
- The future
What is it?

- gathering of small engineering teams
- bringing products and prototypes
- from companies, research labs and universities
- test interoperability and robustness of SIP-related software and hardware
Goals

- diverse, interoperable, robust SIP implementations
- unambiguous base and extension specifications
- best current practices
- community development
What it is not...

- competitive – no winner
- recruiting event
- press conference
- demo
- benchmarking test
- VON, IEEE conference
History

companies attending

companies

May 2001
# SIP Interoperability Test Events

## Events

Every four months:

<table>
<thead>
<tr>
<th>#</th>
<th>when</th>
<th>where/host</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>April 1999</td>
<td>Columbia University</td>
<td>New York, New York</td>
</tr>
<tr>
<td>2</td>
<td>August 1999</td>
<td>Pulver</td>
<td>Melville, New York</td>
</tr>
<tr>
<td>3</td>
<td>December 1999</td>
<td>Ericsson</td>
<td>Richardson, Texas</td>
</tr>
<tr>
<td>4</td>
<td>April 2000</td>
<td>3Com</td>
<td>Schaumburg (Chicago), Illinois</td>
</tr>
<tr>
<td>5</td>
<td>August 2000</td>
<td>Pulver</td>
<td>Melville, New York</td>
</tr>
<tr>
<td>6</td>
<td>December 2000</td>
<td>Sylantro/Sun</td>
<td>Santa Clara, California</td>
</tr>
<tr>
<td>7</td>
<td>March 2001</td>
<td>ETSI</td>
<td>Cannes, France</td>
</tr>
<tr>
<td>8</td>
<td>August 2001</td>
<td>Ubiquity</td>
<td>Cardiff, UK</td>
</tr>
<tr>
<td>9</td>
<td>December 2001</td>
<td>Nuera</td>
<td>San Diego, California</td>
</tr>
</tbody>
</table>
What can be (and has been) tested

Anything that speaks SIP:

- SIP phones, PC “soft clients”
- proxy servers
- test tools
- unified messaging servers
- conferencing servers
- RTP interoperability (mostly audio, some video)
What does the interoperability test event look like?

- lab ← conference room ← hotel ball room
- tables with Ethernet connectivity and power
- (SIP) phone or intercom system for communication
- separate rooms for “scenarios”
Traditions

- engineers and programmers, not laywers or marketing
- on-site bug fixing
- no written non-disclosure agreements or MoUs – mutual trust (and threat to be asked not to come again)
- detailed results remain confidential – no press releases touting individual achievements (“Our Acme SIP widgets were better than anybody else’s SIP widgets”)
- offered at or below cost, students free
Tests

- most tests are pair-wise, but also group scenarios
- scheduling semi-spontaneous, with scheduled scenarios
- based on capability self-assessment (basic, intermediate, advanced)
Typical test experience

First time (novice): make phone ring, avoid crashing on basic calls, test with relatively few (since lots of bugs to fix)

Second time (intermediate): test with more implementations, more features

Third time (veterans): participate in multi-group scenarios
Basic SIP capabilities for UAs

- send and receive INVITE over UDP
- generate ACK properly
- can accept or reject calls
- SDP with single m and c line, one codec
- To, From, Call-ID, CSeq, Via, Content-Length, Content-Type headers handled properly
- generate tags in To field
- send basic call termination with BYE via UDP
- receive BYE over UDP
- compact form for headers
- reject unknown request methods with 501 response
- send/receive RTP media, possibly without RTCP
Intermediate SIP capabilities for UAs

- support TCP for all messages
- Require, Proxy-Require
- handle packet loss for INVITE and BYE (with exponential backoff)
- pays attention to Contact header in INVITE and in 2xx response to INVITE (i.e., goes directly to peer for following requests)
- process CANCEL for INVITE
- Authentication for registrations: basic and digest
- allow redirection to web pages or email
- receive text or HTML in 3xx or 4xx responses
- Accept headers without SDP
- DNS SRV records
- register with periodic refresh to unicast address
Intermediate SIP capabilities for UAs

- understands redirection
- multiple codecs listed in SDP m line, finds common one with peer
- multiple SDP m= lines handled correctly
- unknown SDP m= media types handled correctly
- Domain name as well as IP address accepted in SDP c= line
- generate RTCP packets
- respond to OPTIONS request
- allows non-SIP URLs in REGISTER
- copy Record-Route from response into Route of request and route appropriately
- checks equality of action parameters on REGISTER
- can retrieve current registrations
- can clear registrations with Contact: * and Expires: 0
Advanced SIP capabilities for proxies

- forking proxies: sequential
- forking proxies with multiple 200 OK responses
- recursion on forking (fork response of 3xx triggers new branch)
- legal fork looping (with different request URIs)
- forking for non-INVITE
- digest and digest authentication for INVITE
- can detect loops
- can insert Record-Route
- drops request when Max-Forwards is zero
- obeys Expires in INVITE
- third party registration
- registration proxying
- process multicast REGISTER
- multicast INVITE
- received parameter in Via field
- can always redirect (configurable)
- IPsec support
- TLS support
Test scenarios

Typically require more than two participants:

- tel: URL translation
- multi-stage proxy
- media changes (re-INVITE)
- multicast REGISTER
- loop detection
Preparing your implementation for a SIP test event

- liberal in what you receive, conservative in what you send
- pass SIP parser torture tests
- follow standard SIP call flows
- use one of the SIP phones
- use free/cheap SIP implementations
- on-line servers (Lucent, Nortel, Ubiquity, 3Com, Worldcom, Columbia University, …)
The future

- evolving event – balance between informality and structure
- “users of SIP”: 3GPP, CableForum specs, …
- presence and messaging
- services and features
- third-party call control
- CPL and sip-cgi
- interworking with H.323, soft switches
Summary

- cooperative effort of lots of vendors to improve reliability and interoperability
- help to make specifications clearer and less ambiguous
- create a community of SIP developers