What can we learn from 20+ years of SIP and VoIP?

Henning Schulzrinne - Columbia University

STIR/SHAKEN Enterprise Summit - October 17, 2022
Exploring past and future

What landscape did SIP emerge from?

What (likely) made it successful?

Why aren’t Zoom/Teams/WebEx/… using SIP & RTP (mostly)?

What have we learned about video calls & conferencing since 1964?

What could be next?
Where did we start?
Web vs. VoIP

- Request routing is static, initiated by the client, mostly DNS + some HTTP redirect.

- Fairly sophisticated price-based routing, largely provider-driven.

- Diagram showing CDN, contract, and various services connected to clients, PBX, and carriers.
SIP: Session Initiation Protocol

Abstract

Many styles of multimedia conferencing are likely to co-exist on the Internet, and many of them share the need to invite users to participate. The Session Initiation Protocol (SIP) is a simple protocol designed to enable the invitation of users to participate in such multimedia sessions. It is not tied to any specific conference control scheme, providing support for either loosely or tightly controlled sessions. In particular, it aims to enable user mobility by relaying and redirecting invitations to a user’s current location.

This document is a product of the Multiparty Multimedia Session Control (MMUSIC) working group of the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at confctrl@isi.edu and/or the authors.
SIP could be explained on a (small) napkin

C->S: INVITE sip:watson@boston.bell-tel.com SIP/2.0
   Via: SIP/2.0/UDP kton.bell-tel.com
   From: A. Bell <sip:a.g.bell@bell-tel.com>
   To: T. Watson <sip:watson@bell-tel.com>
   Call-ID: 3298420296@kton.bell-tel.com
   CSeq: 1 INVITE
   Subject: Mr. Watson, come here.
   Content-Type: application/sdp
   Content-Length: ...

v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
s=Mr. Watson, come here.
c=IN IP4 kton.bell-tel.com
m=audio 3456 RTP/AVP 0 3 4 5
What (might have) been reasons for success

- **Timing**: early enough before proprietary or ITU-T solutions could catch up
  - ISDN compatibility was never incentive enough
  - but SIP was close enough to feature parity to digital PBX and analog phones (parallel forking!)
- **Scope**: Competitor H.323 was focused on conference rooms, not calls
  - remained niche market
- **Familiarity**: HTTP-like syntax and re-use
  - could be stateless (until SBCs took over)
- **Low barrier to entry**: text-based, UDP, copy-paste examples
  - pass the “assign as homework” test
  - H.323 had mix of Q.931 bit-based TLV & ASN.1 (H.225.0 & H.245)
But these also proved to be troublesome

- **UDP transport**: significant edge-case complexity
  - embedding retransmission adds complexity (multi-hop)
  - mixed transport protocols add failure modes
  - lots of SIP headers + larger bodies + TLS ⇒ bad idea

- **SDP for media**: offer-answer has been trouble
  - hard to add structured alternatives and parameters
  - SDPng never made it (2nd system syndrome...)

- **Protocol encoding**: interoperability issues (code to example, not spec)
  - angry fruit salad of SMTP, HTTP/1.1, base-64 JWTs, SDP, MIME multipart, ...
  - relatively few libraries → HTTP/3 binary mode affects few
  - “While these exchanges are human readable, using whitespace for message formatting leads to parsing complexity and excessive tolerance of variant behavior” (RFC 9114)
Many (most?) SIP vulnerabilities are parser-related.

- **CVE-2022-31031**: PISSIP is a free and open source multimedia communication library written in C language implementing standard based protocols such as SIP, SDP, STUN, TURN, and ICE. In versions prior to and including 2.12.1, a stack buffer overflow vulnerability affects PISSIP users that use STUN in their applications, either by: setting a STUN server in their account/media config in PJSU/PJSUA2 level, or directly using "pjsb-util/stun\_simple" API. A patch is available in commit 450baca which should be included in the next release. There are no known workarounds for this issue.

- **CVE-2022-31001**: Sofía-SIP is an open-source Session Initiation Protocol (SIP) User-Agent library. Prior to version 1.13.8, when parsing each line of a sip message, \"rest = record + 2\" will access the memory behind \"0\" and cause an out-of-bounds write. An attacker can send a message with evil sip to FreeSWITCH, causing a crash or more serious consequence, such as remote code execution. Version 1.13.8 contains a patch for this issue.

- **CVE-2022-31002**: Sofía-SIP is an open-source Session Initiation Protocol (SIP) User-Agent library. Prior to version 1.13.8, an attacker can send a message with evil sip to FreeSWITCH, which may cause a crash. This type of crash may be caused by a URL ending with \"\". Version 1.13.8 contains a patch for this issue.

- **CVE-2022-29855**: Sofía-SIP is an open-source Session Initiation Protocol (SIP) User-Agent library. Prior to version 1.13.8, an attacker can send a message with evil sip to FreeSWITCH, which may cause a crash. This type of crash may be caused by \"advertise\" mismatched (m|c) (there are \m|c = 0\) which will make m/c logger and trigger out-of-bounds access when \"1522,0659,0659\" (m|c) - 1 = 0\) which will make m/c logger and trigger out-of-bounds access when \"1522,0659,0659\" (m|c) - 1 = 0\).

- **CVE-2022-29855**: Mitel 6800 and 6900 Series SIP phone devices through 2022-04-27 have "undocumented functionality." A vulnerability in Mitel 6800 Series and 6900 Series SIP phones excluding 6970, versions 5.1 SP (5.1.0.0816) and earlier, and 6.0 (6.0.0.386) through 6.1.64 (6.1.0.165), could allow an unauthenticated attacker with physical access to the phone to gain root access due to insufficient access control for test functionality during system startup. A successful exploit could allow access to sensitive information and code execution.

- **CVE-2022-29338**: Mising access control in the backup system of Teleos' VitalPbX before 3.2.1 allows attackers to access the PISSIP and PISSIP extension APIs. This allows an attacker to remotely execute code without authentication via a crafted SIP packet that contains malicious SDP data.

- **CVE-2022-26379**: On FS BIG-IP 16.1.x versions prior to 16.1.2.2, 15.1.x versions prior to 15.1.5, and 14.1.x versions prior to 14.1.4.6, when a Session Initiation Protocol (SIP) message routing framework (MRF) application layer gateway (ALG) profile is not configured for an application, undisclosed requests may be performed. Software versions which have reached End of Technical Support (EoTS) are not evaluated.

- **CVE-2022-23508**: PISSIP is a free and open source multimedia communication library written in C language implementing standard based protocols such as SIP, SDP, RTP, STUN, TURN, and ICE. In versions up to and including 2.12.1 when in a dialog set (or forking) scenario, a hash key shared by multiple UAC dialogs can potentially be permanently freed when one of the dialogs is destroyed. The IPv4 may cause a dialog set to be registered in the hash table multiple times (with different hash hash) leading to undefined behavior such as dialog list collision which eventually leading to endless loop. A patch is available in commit db3339935ba2d62d80276ca510f6ca756143 which will be included in the next release. There are no known workarounds for this issue.

- **CVE-2022-23035**: On BIG-IP version 16.1.x before 16.1.1, 15.1.x before 15.1.4, 14.1.x before 14.1.4, and all versions of 13.1.x, when a SIP profile is configured on a virtual server, undisclosed requests can cause the Traffic Management Microkernel (TMM) to terminate. Note: Software versions which have reached End of Technical Support (EoTS) are not evaluated.

- **CVE-2022-22704**: An Improper Release of Memory before removing Last Reference vulnerability in the Session Initiation Protocol (SIP) Application Layer Gateway (ALG) of Juniper Networks Junos OS allows unauthenticated network-based attacker to cause a partial Denial of Service (DoS). On all MX and SRX platforms, if the SIP ALG is enabled, receipt of a specific SIP packet will cause a state SIP entry. Sustained receipt of such packets will cause the SIP call state to eventually fill up and cause a DoS for all SIP traffic. The SIP call usage can be monitored by "show security alg sip", to be affected the SIP ALG needs to be enabled, either implicitly / by default or by way of configuration. Please verify on MX with: usershow ssh show security alg status | match sip | Enabled Please verify on MX whether the following is configured: [services ... rule <rule-name> (form <term-name> from/match application/application-set name..) ] where either a. name = junos-sip or an application or application-set refers to SIP: b. [ applications application <name> (form <term-name> from/match application/application-set name..) ] or c. [ applications application <name> application-protocol sip ] or d. [ applications application <name> application-protocol sip ] This issue affects Juniper Networks JUNOS OS on SRX Series and MX Series: 20.4 versions prior to 20.4R3-S2; 21.1 versions prior to 21.1R2-S2; 21.2 versions prior to 21.2R2-S2; 21.2 versions prior to 21.2R3; 21.3 versions prior to 21.3R2; 21.4 versions prior to 21.4R2. This issue does not affect Juniper. SIP is not aware of any malicious exploitation of this vulnerability.

- **CVE-2022-22198**: An Access of Uninitialized Pointer vulnerability in the SIP ALG of Juniper Networks Junos OS allows an unauthenticated network-based attacker to cause a Denial of Service (DoS). Continued receipt of these specific packets will cause a sustained Denial of Service condition. On all MX and SRX platforms, if the SIP ALG is enabled, an MS-PMC or MS-MC, or SP-S will crash if it receives a SIP message with a specific contact header format. This issue affects Juniper Networks JUNOS OS on MX Series and SRX Series: 20.4 versions prior to 20.4R3-S1; 21.1 versions prior to 21.1R2-S1; 21.1 versions prior to 21.1R2-S1; 21.2 versions prior to 21.2R2-S1; 21.3 versions prior to 21.3R2. This issue does not affect Juniper. SIP is not aware of any malicious exploitation of this vulnerability.

- **CVE-2022-22178**: A Stack-Buffer Overflow vulnerability in the flow processing daemon (flowd) of Juniper Networks JUNOS OS on MX Series and SRX Series allows an unauthenticated networked attacker to cause a flow processing daemon (flowd) crash and thereby a Denial of Service (DoS). Continued receipt of these specific packets will cause a sustained Denial of Service condition. This issue can be triggered by a specific Session Initiation Protocol (SIP) invite packet if the SIP ALG is enabled. Due to this, the PIC will be rebooted and all traffic that traverses the PIC will be dropped. This issue affects: Juniper Networks JUNOS OS 20.4 versions prior to 20.4R3-S1; 21.1 versions prior to 21.1R2-S1; 21.1 versions prior to 21.1R2-S1; 21.2 versions prior to 21.2R2-S1; 21.3 versions prior to 21.3R2; 21.3 versions prior to 21.3R2. This issue does not affect Juniper Networks JUNOS OS versions prior to 20.4R1.

- **CVE-2022-22175**: An Improper Locking vulnerability in the SIP ALG of Juniper Networks JUNOS OS on MX Series and SRX Series allows an unauthenticated networked attacker to cause a flowprocessing daemon (flowd) crash and thereby a Denial of Service (DoS). Continued receipt of these specific packets will cause a sustained Denial of Service condition. This issue can occur in a scenario where the SIP ALG is enabled and specific SIP messages are being processed simultaneously. This issue affects: Juniper Networks JUNOS OS on MX Series and SRX Series: 20.4R3-S1; 21.1 versions prior to 21.1R2-S1; 21.2 versions prior to 21.2R2-S1; 21.3 versions prior to 21.3R2; 21.2 versions prior to 21.3R2; 21.3 versions prior to 21.3R2.
SIP should have anticipated NAT

- SIP and IPv6 evolved at roughly the same time
  - assumption: NATs = nuisance all temporary
- High-speed home access for VoIP didn’t exist (~ 3-4% of US households)

---

### High-Speed Services for Internet Access: Subscribership as of June 30, 2000 (FCC)

<table>
<thead>
<tr>
<th>Types of Technology*</th>
<th>December 1999</th>
<th>June 2000</th>
<th>% Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSL</td>
<td>369,792</td>
<td>950,590</td>
<td>157%</td>
</tr>
<tr>
<td>Other Wireline</td>
<td>609,909</td>
<td>747,028</td>
<td>22%</td>
</tr>
<tr>
<td>Coaxial Cable</td>
<td>1,414,183</td>
<td>2,248,981</td>
<td>59%</td>
</tr>
<tr>
<td>Fiber</td>
<td>312,204</td>
<td>307,151</td>
<td>n.m.</td>
</tr>
<tr>
<td>Satellite &amp; Fixed Wireless</td>
<td>50,404</td>
<td>65,615</td>
<td>n.m.</td>
</tr>
<tr>
<td>Total Lines</td>
<td>2,756,492</td>
<td>4,319,365</td>
<td>57%</td>
</tr>
</tbody>
</table>
SIP design: NATs continue to constrain

Lots of streams
- audio, video, screen sharing
- signaling, BFCP, XCON, ...

Good for
- modularity (choose different protocols)
- QoS (apply different treatments)
- selective forwarding vs. mixing

NATs
- need to get public IP for each
- hard to re-use inbound connections

Firewalls
- hard to have port rules

Multiplexed media over HTTP (or TCP)
- ugly, but that's what Zoom does
What would a “SIP” version 3 look like?

HTTP/3-based → RIPT (*)

- different trade-off between standards and local software
- asymmetric (client-server)

Unclear whether current common SIP use cases would be significantly improved

- e.g., unlikely to achieve UE interoperability for complex video scenarios

(*) draft-rosenbergjennings-dispatch-ript-00
Shouldn’t STIR/SHAKEN been done in 2002?

Yes, but RFC 4474 was published in 2006!

Just like for SSL/TLS (1995)

- in 2013, only 23% of European websites had encryption
- Let’s Encrypt, AWS ACM, … probably mattered more than protocols

https://doi.org/10.1145/2872518.2888605
Regulatory push & pull likely matter more than technology

STIR/SHAKEN infrastructure, not protocol

Calls% A - 15.79%

Calls% B - 2.79%

Calls% C - 2.87%

Calls% Errors - 1.63%

Calls% Unsigned - 76.91%

https://transnexus.com/blog/2022/shaken-statistics-september/
Signing is a very good predictor of robocalls.

https://transnexus.com/blog/2022/shaken-statistics-september/
Why did standards fail for video conferencing (mostly)?
Not news: Lots of people spend lots of time on video

~ 5.7 M people on 24/7
300M participants per day
AT&T videophone 1995 ($1,499 or $30/day)
Video relay service: VP-100 (2000)

H.323 (TV)

Reach users by E.164 phone number

Now: SIP-based
Probably largest interoperable, public video network
(IETF RUM working group working on profile)
The landscape of IP video communications

2-party phone call, spontaneous

- Differentiated roles (organizer, panelist, audience)
- Some audience participation
- Up to 50,000 participants

CuSeeMe (1992)

Unified Communications (UC)

Multi-party streaming (Mbone, YouTube, FB Live, Livestream)
- One way, except chat & comments
Lessons learned since 1964

- Two-party video is rarely useful except for specialty applications (telemedicine & adult entertainment)
  - But popular for environment sharing (“let me show you my new apartment”)
- Most video “calls” are scheduled → call signaling by calendar invite and SMTP, not SIP
- Chat and screen sharing are the most useful Zoom features
- The most useful video conferencing accessory is a better *microphone* (and maybe a ring light)
Such a mobility turn in video communication enables participants to show something to their interlocutor. Thirty percent of mobile video conversations seem to unfold around the intent of one of the participants to show something to the other, which is probably an underestimate because showing also occurs in video calls that do not have that as an initial goal. From what we observed in the Skype part of our own corpus, the numbers should be much in the same range also for Skype interactions. With the possibility of video communication technologies being able to show something during a call, these at last seem to fulfill their early and heretofore unkept promise that they would allow remote conversationalists to share their environments. A related line of research has looked at “video-as-data,” that is, how some part of the ongoing activity could be recorded and made available in real time to provide a shared field of interaction in collaborative situations. In such a configuration, the participants work to articulate video and speech occurrences in a way that is relevant to the unfolding interaction.
What we think Zoom is...
The hard part for interoperable video interaction but also a reason people do audio-only Zoom calls!
Aside: What’s wrong with the Zoom video model?

See Jeremy N. Bailenson (Feb. 2021):

- **Eye gaze at a close distance (cf. elevator gaze aversion) - no zoom on Zoom!**
  - “long stretches of direct eye gaze and faces seen close up” (~50 cm)
  - for mid-sized meetings, everybody looks at every other non-speaker

- **Constant self-monitoring**
  - “centering oneself in the camera’s field of view, nodding in an exaggerated way for a few extra seconds to signal agreement, or looking directly into the camera”)
  - side glances are misconstrued

- **All day mirror**

- **Reduced mobility**
  - moving out of camera view is seen as sign of non-attention

- **VR may make this worse, e.g., by confusing positional cues**
  - who “sits” where? How do I see the person’s face if covered by VR goggles?
Video (and audio) are a small part of the system!

- **Call signaling**
  - to hardware (SIP, H.323)
  - to software

- **Floor control** (hand-raising, muting others, ...) (BFCP)

- **Media transport** (RTP)
  - echo cancellation

- **Media quality feedback** (RTCP)

- **Bandwidth adjustment**

- **Packet loss recovery**

- **“API”: set up sessions, functions XCON (RFC 5239+)**

- **Required, available and widely used**

- **Available, but not widely used**

- **Not available or limited functionality**

- **No (interoperability) need for standards**

- **Text chat (incl. reactions)** (RCS, T.140, MSRP, XMPP)

- **Screen sharing**

- **G.711 (4 kHz)**
  - G.722 (8 kHz)
  - OPUS (HQ)
  - G.723, G.729, iLBC, ...

- **H.264 (MPEG-4) AVC**
Standards = technology translator

● Similar in some ways to textbooks
● “accepted technology”
  ○ lower/known risks ("vetted")
  ○ infrastructure ("eco system")
  ○ libraries, test tools, text books, certification, …
  ○ reduce cost of picking among roughly equal choices
  ○ sometimes reduce IPR risks ("patent pool", RAND)
● requires expertise and broader training
  ○ many CS standards don’t have either
  ○ example: HTTP/1.0, HTML 1.0, 802.11 WEP
Interoperability: indifferent, cooperative, competitive

[Doctorow, CACM 10/2021]

- Indifferent interoperability
  - company A does not care that B makes a complementary product

- Cooperative interoperability
  - typically via standards
  - but may play favorites

- Competitive (or adversarial) interoperability
  - "third-party inkjet ink, DVRs that record anything"
  - see copyright-for-API (Google vs. Oracle)
When do we get standards

<table>
<thead>
<tr>
<th>Condition</th>
<th>VoIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect different industries</td>
<td>PBX + carriers; mobile + landline; device + carrier</td>
</tr>
<tr>
<td>Industries with different emphasis</td>
<td>Hardware (incl. niches) vs. software vs. operations</td>
</tr>
<tr>
<td>Non-dominance of single vendor or operator</td>
<td>lots of local, niche &amp; national carriers (unlike browser)</td>
</tr>
<tr>
<td>Minimize interconnection preparation</td>
<td>don’t want to install new software (with new UI) for each call</td>
</tr>
<tr>
<td>Interoperability with legacy technology</td>
<td>150 years: analog, SS7, ISDN</td>
</tr>
</tbody>
</table>
### SIP Standards

#### Core SIP Documents

<table>
<thead>
<tr>
<th>RFC</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2543</td>
<td>SIP: Session Initiation Protocol (obsolete)</td>
</tr>
<tr>
<td>3261</td>
<td>SIP: Session Initiation Protocol</td>
</tr>
<tr>
<td>3262</td>
<td>Reliability of Provisional Responses</td>
</tr>
<tr>
<td>3263</td>
<td>Locating SIP Servers</td>
</tr>
<tr>
<td>3265</td>
<td>SIP-Specific Event Notification</td>
</tr>
<tr>
<td>5954</td>
<td>Essential Correction for IPv6 ABNF and URI Comparison in RFC 3261</td>
</tr>
</tbody>
</table>

#### SDP-Related Documents

<table>
<thead>
<tr>
<th>RFC</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2327</td>
<td>Session Description Protocol (SDP) (obsolete: see RFC 4566)</td>
</tr>
<tr>
<td>3264</td>
<td>An Offer/Answer Model with the Session Description Protocol (SDP)</td>
</tr>
<tr>
<td>3266</td>
<td>Support of IPv6 in SDP</td>
</tr>
<tr>
<td>3388</td>
<td>Grouping Media Lines in SDP (obsolete: see RFC 5888)</td>
</tr>
<tr>
<td>3407</td>
<td>Session Description Protocol (SDP) Simple Capability Declaration</td>
</tr>
<tr>
<td>3524</td>
<td>Mapping of Media Streams to Resource Reservation Flows</td>
</tr>
<tr>
<td>3556</td>
<td>SDP Bandwidth Modifiers for RTCP Bandwidth</td>
</tr>
<tr>
<td>3605</td>
<td>Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)</td>
</tr>
<tr>
<td>3890</td>
<td>A Transport Independent Bandwidth Modifier</td>
</tr>
<tr>
<td>4091</td>
<td>An Alternative NAT Semantics for SDP</td>
</tr>
<tr>
<td>4145</td>
<td>TCP-Based Media Transport in the SDP</td>
</tr>
<tr>
<td>4566</td>
<td>Session Description Protocol (SDP)</td>
</tr>
<tr>
<td>4567</td>
<td>Key Management Extensions for SDP and RTSP</td>
</tr>
<tr>
<td>4568</td>
<td>SDP Security Descriptions for Media Streams</td>
</tr>
<tr>
<td>4570</td>
<td>SDP Source Filters</td>
</tr>
<tr>
<td>4572</td>
<td>Connection-Oriented Media Transport over TLS in SDP</td>
</tr>
<tr>
<td>4574</td>
<td>SDP Label Attribute</td>
</tr>
</tbody>
</table>

- Roughly 300 with SIP in title (RFC editor)
- IMS 23.228: 329 pg.
- RCS 5.1: 482 pg.
Standards can take a looong time (and RFCs are decreasing)

### Conference Information Data Model for Centralized Conferencing (XCON)

<table>
<thead>
<tr>
<th>Status</th>
<th>IESG evaluation record</th>
<th>IESG writeups</th>
<th>Email expansions</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Versions</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- [draft-no-von-xcon-common-data-model](#)
- [draft-ietf-xon-xcon-common-data-model](#)
- [rfc6501](#)

- [Publication rate per year](#)
  - [Number of RFCs](#)
    - [Year](#)
It takes a lot of people to do the work
Simple core protocols have acquired technical debts

<table>
<thead>
<tr>
<th>RFC</th>
<th>Type</th>
<th>Status</th>
<th>Title</th>
<th>Bind Prot</th>
<th>Names</th>
<th>Ops</th>
<th>RR</th>
<th>Proxy</th>
<th>Stub</th>
<th>Auth</th>
<th>Res</th>
<th>YF</th>
<th>RFCSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>882</td>
<td>Obsolete</td>
<td></td>
<td>Domain Names – Concepts and Facilities</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>883</td>
<td>Obsolete</td>
<td></td>
<td>Domain Names – Implementation and Specification</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>920</td>
<td></td>
<td></td>
<td>Domain Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>973</td>
<td>Obsolete</td>
<td></td>
<td>Domain System Changes and Observations</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1032</td>
<td></td>
<td></td>
<td>Domain Administrators Guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1033</td>
<td></td>
<td></td>
<td>Domain Administrators Operations Guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1034</td>
<td>Standard</td>
<td></td>
<td>Domain Names – Concepts and Facilities</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1035</td>
<td>Standard</td>
<td></td>
<td>Domain Names – Implementation and Specification</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td></td>
<td></td>
<td>DNS Encoding of Network Names and Other Types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1123</td>
<td>Standard</td>
<td></td>
<td>Requirements for Internet Hosts – Application and Support</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1178</td>
<td>Informational</td>
<td></td>
<td>Choosing a Name for Your Computer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DNS: ~143 active RFCs
Sidebars: XCON and CCMP

IETF attempt in 2008-2012 to standardize basic conference management

Data model for conference (XML)

  e.g., user admission, sidebars (breakout rooms), floors

API (operations) on data model → CCMP

Left out polling, advanced breakout functions, waiting rooms, ...
Addressing - vision & reality

Original idea: SIP URLs (sip:user@domain) or tel URLs (tel:+1-201-555-0123) still exists and useful for hardware

Current reality: web URLs via web page, email, calendar, Slack, IM, SMS, ...
Beyond protocols - what do users expect?

**Video conferences:**
- NAT traversal
- Cross-domain authentication and authorization
- Calendar interface
- Media routing
- Scalable capacity (tens to thousands per session)
- End-to-end security
- Media gateways (phone, room systems)
- Polling
- Recording and playback
- Transcription (accessibility, records)
- Language translation
- Managing abuse (“Zoom bombing”, criminal activity, extremism)

**Webinars:**
- Attendee management
- Connect to YouTube, Facebook Live, ...
- Monetization
- Polling and “engagement”
Operational models

Enterprise

PBX heritage
“Unified communications”
Hosted in corporate data center

Carrier

Early Skype architecture
Common elsewhere: SMTP, XMPP, IRC*, Usenet
but usually large user/server ratio

Cloud

SIP-based: RCS (mostly messaging)
struggled with higher-quality audio (HD audio)

Peer-to-peer

Rooted in corporate heritage
Struggling with consumer use (and abuse)
Not quite peer-to-peer: “permissioned” networks

<table>
<thead>
<tr>
<th>IRC</th>
<th>today</th>
<th>yesterday</th>
<th>network</th>
<th>users Ø</th>
<th>channels Ø</th>
<th>servers Ø</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1.</td>
<td>Libera Chat</td>
<td>36564</td>
<td>18711</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>2.</td>
<td>IRCnet</td>
<td>20115</td>
<td>10685</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>3.</td>
<td>Undernet</td>
<td>14574</td>
<td>6065</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>4.</td>
<td>EFnet</td>
<td>11765</td>
<td>6892</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>5.</td>
<td>OFTC</td>
<td>11623</td>
<td>2327</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>6.</td>
<td>Rizon</td>
<td>11511</td>
<td>8803</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>7.</td>
<td>QuakeNet</td>
<td>9909</td>
<td>8780</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>8.</td>
<td>DALnet</td>
<td>7839</td>
<td>3861</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>9.</td>
<td>Spoonet</td>
<td>4262</td>
<td>5734</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>10.</td>
<td>GIMPnet</td>
<td>3352</td>
<td>368</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>11.</td>
<td>KampungChat</td>
<td>3197</td>
<td>459</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>12.</td>
<td>hackint</td>
<td>3195</td>
<td>1753</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>13.</td>
<td>GeekShed</td>
<td>3175</td>
<td>219</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>15.</td>
<td>SimosNap</td>
<td>2631</td>
<td>522</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>16.</td>
<td>OltreIRC</td>
<td>2596</td>
<td>30</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>17.</td>
<td>ExplosionIRC</td>
<td>2591</td>
<td>61</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>18.</td>
<td>18.</td>
<td>EsperNet</td>
<td>2430</td>
<td>2533</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td>19.</td>
<td>GameSurge</td>
<td>2122</td>
<td>1639</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td>20.</td>
<td>synIRC</td>
<td>2092</td>
<td>1103</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td>21.</td>
<td>Abjects</td>
<td>2074</td>
<td>341</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td>22.</td>
<td>SceneP2P</td>
<td>1771</td>
<td>68</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td>23.</td>
<td>IRCHighWay</td>
<td>1445</td>
<td>661</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td>24.</td>
<td>EurooNet</td>
<td>1353</td>
<td>983</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td>25.</td>
<td>OpenJoke</td>
<td>1095</td>
<td>51</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>27.</td>
<td>tilde_chat</td>
<td>1006</td>
<td>445</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

Freenode IRC staff resign en masse, unhappy about new management

Network boss Andrew Lee disputes claims made by those leaving the internet chat community

Thomas Claburn in San Francisco

Wed 19 May 2021 // 21:50 UTC

Most of the volunteer staff of Freenode, an internet relay chat (IRC) network dating back to 1995, have resigned in protest over what they describe as a hostile takeover of the chat service.

And many have launched an alternative service, Libera Chat.

Freenode, which has focused on serving as a real-time communication channel for free and open source software projects, currently has about 76,000 users and 42,000 chat rooms.

In a resignation letter, a staffer called Christian, who is also known as Fuchs on Freenode, said after 10 years helping with the network, he is leaving because he disagrees with the direction being taken by Andrew Lee, founder of VPN firm Private Internet Access (PIA), who acquired a controlling interest [PDF] in Freenode's holding company in 2017.
## What are the strengths of the operational models?

<table>
<thead>
<tr>
<th>Feature</th>
<th>Enterprise hosted</th>
<th>Peer-to-peer</th>
<th>Carrier</th>
<th>“VCaaS”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictable features</td>
<td>Mostly</td>
<td>Difficult</td>
<td>Unlikely (Android!)</td>
<td>Mostly</td>
</tr>
<tr>
<td>Cross-domain AA</td>
<td>guests with passwords</td>
<td>sybils</td>
<td>“roaming”</td>
<td>added SSO, but still mostly secret strings</td>
</tr>
<tr>
<td>Media routing</td>
<td>rare</td>
<td>challenging</td>
<td>usually national only</td>
<td>As far as the cloud will stretch</td>
</tr>
<tr>
<td>Scalable capacity</td>
<td>rare</td>
<td>freelancer problem</td>
<td>struggling with cloud</td>
<td>natural</td>
</tr>
<tr>
<td>End-to-end security</td>
<td>easy</td>
<td>easy for 2-party, no mixing</td>
<td>wiretapping laws</td>
<td>challenging with media mixing</td>
</tr>
<tr>
<td>Media gateways</td>
<td>PBX dial-in</td>
<td>nobody ever tried*</td>
<td>“we are the phone company!”</td>
<td>outsourced</td>
</tr>
<tr>
<td>Recording &amp; playback</td>
<td>with effort (rare)</td>
<td>nobody ever tried</td>
<td>struggling with cloud</td>
<td>easy</td>
</tr>
<tr>
<td>Transcription, translation</td>
<td>challenging</td>
<td>nobody ever tried</td>
<td>similar to VCaaS</td>
<td>in progress</td>
</tr>
<tr>
<td>Manage abuse</td>
<td>Challenging for smaller entities (schools, nonprofits)</td>
<td>lots of PhD theses were written</td>
<td>have fraud &amp; security departments, but “common carrier” tradition</td>
<td>incompatible with no-touch model; unexpected role</td>
</tr>
</tbody>
</table>
But it’s really the business model that killed interoperability

Old models: Open source, enterprise software license or built into phone

Open source: who is going to run the server → open source companies get bought by operations (“cloud”) companies (e.g., Jitsi)

Enterprise: who wants to run and maintain a PBX server?

see: email outsourcing

Caller pays is back: Caller (= host) pays for meeting; participants are free
VoIP clients need inbound connections for call signaling and media

Video conference clients rely on participants to initiate sessions and participation - outbound only signaling — but still may need inbound media

Late 1990s: The only users with enough bandwidth didn’t have NATs
Early 2000s: NATs are evil and IPv6 will kill them

https://anyconnect.com/stun-turn-ice/

Somebody has to provide the STUN and TURN servers
But not quite - Zoom uses P2P for two-party calls

The versioning problem

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aparté</td>
<td>BSD / Linux / macOS</td>
</tr>
<tr>
<td>AstraChat</td>
<td>Android / iOS / Linux / macOS / Windows</td>
</tr>
<tr>
<td>BeagleIM by Tigase, Inc.</td>
<td>macOS</td>
</tr>
<tr>
<td>blabber.im</td>
<td>Android</td>
</tr>
<tr>
<td>Bruno the Jabber™ Bear</td>
<td>Android</td>
</tr>
<tr>
<td>Conversations</td>
<td>Android</td>
</tr>
<tr>
<td>Converse</td>
<td>Browser</td>
</tr>
<tr>
<td>Dino</td>
<td>Linux</td>
</tr>
<tr>
<td>Gajim</td>
<td>Linux / Windows</td>
</tr>
<tr>
<td>Kaidan</td>
<td>Android / Linux / macOS / Other / Windows</td>
</tr>
<tr>
<td>Monal IM</td>
<td>iOS / macOS</td>
</tr>
<tr>
<td>Movim</td>
<td>Android / Browser / Linux / macOS / Windows</td>
</tr>
<tr>
<td>Poezie</td>
<td>Linux / macOS</td>
</tr>
<tr>
<td>Profanity</td>
<td>Linux / macOS / Windows</td>
</tr>
<tr>
<td>Psi</td>
<td>Linux / macOS / Windows</td>
</tr>
<tr>
<td>Psi+</td>
<td>Linux / macOS / Windows</td>
</tr>
<tr>
<td>Pâité</td>
<td>Browser</td>
</tr>
</tbody>
</table>
WebRTC as transition model

Standards-based client

WebRTC client

Application

multiple services, one client

no installation - one “page” per service
switch browsers & maybe platforms
no interoperability between services

No interoperability between services
WebRTC architecture

The web

Web API (Edited by W3C WG)

WebRTC

WebRTC C++ API (PeerConnection)

Session management / Abstract signaling (Session)

Voice Engine
- iSAC / ILBC Codec
- NetEQ for voice
- Echo Canceller / Noise Reduction

Video Engine
- VP8 Codec
- Video jitter buffer
- Image enhancements

Transport
- SRTP
- Multiplexing
- P2P STUN + TURN + ICE

Audio Capture/Render

Video Capture

Network I/O

API for web developers
API for browser makers
Overrideable by browser makers
Typical WebRTC architecture

- **STUN**
- **ICE**
- **websocket (bidirectional TCP)**
- **SRTP (secure media transport)**
- **Apache or nginx serve JS and HTML**

proprietary session signaling (can be SIP or XMPP)
Good for non-square UIs

gather.town

advantages to break-out rooms?
Or lower level still - browser as VM

**WebAssembly SIMD**: SIMD instructions, e.g., to replace video background

**WebTransport**: multiple cancellable streams: datagrams + bidirectional reliable streams

**WebCodecs API**: direct access to codecs
Zoom: vestigial standards compliance

Server-based Traffic

P2P Traffic

Value | Packet Type       | Offset
--- | ----------------- | ----
16   | RTP: Video       | 24
15   | RTP: Audio       | 19
13   | RTP: Screen Share| 27
34   | RTCP: SR + SDES  | 16
33   | RTCP: SR         | 16

makes it easier to interoperate with SIP and H.323 room systems!

---

TABLE II: Comparison of the RTC applications under test. Under *Redundant data*, “F” stands for FEC and “S” for Simulcast. Under *DNS domains*, “B” stands for easy to block, “C” for company-specific and “S” for social networks. Under *Other*, “N” means it uses less than four server-side ports and “T” means that PTs are used in a static fashion.

<table>
<thead>
<tr>
<th>Application</th>
<th>Protocols</th>
<th>Operation Redundant Data</th>
<th>Identification DNS Domains</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RTP</td>
<td>STUN/TURN</td>
<td>DTLS</td>
<td>P2P</td>
</tr>
<tr>
<td>Skype</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Google Meet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Jitsi Meet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WhatsApp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Telegram</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Facebook Messenger</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Instagram Messenger</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Facetime</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HouseParty</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Microsoft Teams</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Webex Teams</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Zoom</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GoToMeeting</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>


GoToMeeting = AVTP (IEEE 1722-2011)
Bifurcation

Communication out front applications: collaboration, social interaction, telemedicine

- challenge: hybrid interactions $\rightarrow$ AR with remote participants?
- challenge: more structured meetings (e.g., recorded votes)
- challenge: unwanted communications -- robocalls and QAnon

Video in back applications: monitoring (traffic, agriculture, security, ...) $\rightarrow$ consumers are ML applications
The uneasy coexistence of synchronous and asynchronous collaboration
Or maybe we’ll just be avatars

https://www.meta.com/work/workrooms/
And the typical group project has...

each with their own login, groups, privileges, ...
Conclusion

Video worked out quite differently than anticipated in the 1990s

probably the component everybody would ditch first for Zoom and kin

Standards-based communications survived where communication without prior arrangement is valued → phone, email, SMS

We think codecs and protocols → systems and operations

Moving from protocol standards to browser as hardware abstraction layer

happening with transport protocols, too (see QUIC)