IETF Video Standards
A review, some history, and some reflections

Colin Perkins
Internet Engineering Task Force

“The goal of the IETF is to make the Internet work better”

Technical development of protocol standards: an open, international, vendor-neutral forum
Internet Engineering Task Force

Multimedia standards → AVT, MMUSIC, and successor working groups (SIP, CLUE, WebRTC, …)
• Media: secure RTP, WebRTC data channel
• Session descriptions: SDP
• Different control protocols for different purposes
  • Announcing multicast sessions: SAP
  • Control of streaming media: RTSP
  • Control of interactive conferencing: SIP
  • Control of telepresence: CLUE
  • Control of web-based interactive media: JSEP (WebRTC)
• Path discovery: ICE, STUN, TURN
Media: RTP

• Separate data and control channels
  • RTP – media payload formats
  • RTCP – source description, reception quality feedback, codec control

• Payload formats
  • Codec-specific packet formats → application level framing → robust, but complex
  • Each frame packetised for independent use, for low latency
  • IETF media codecs: Opus + NetVC

• Profiles
  • Standard + feedback + security

• Other extensions
  • XRBLOCK → extended monitoring
  • Codec control and other feedback
  • Circuit breakers and congestion control

• RFC 1889 → RFC 3550
  Dozens of extensions, payload formats, etc.

• Widely used: voice telephony, video conferencing, telepresence, IPTV…
Media: WebRTC Data Channel

- Direct peer-to-peer data channel between browsers – operates without central server once connection established
- SCTP in secure UDP tunnel:
  - Tunnel → easy to deploy, incompatible with SCTP-level multihoming support
- Transparent data delivery:
  - Message-oriented abstraction
  - Multiple sub-streams
  - Full or partial reliability
  - Congestion controlled
- Potentially highly disruptive → trivial to build P2P applications with WebRTC and the data channel
Session Descriptions: SDP

v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000

• Control protocols need to describe session to be controlled
  • Media transport + payload formats
  • Addresses and ports
  • Originator and purpose of session
  • Options and parameters
• SDP provides a standard format for this data → declarative mode
  • SDP very effective in this use case
Multicast Session Announcement: SAP

- Initial use case: multicast sessions on the Mbone
  - Session directory – multicast declarative SDP
  - Multicast RTP media – broadcast and interactive
  - Any source multicast (ASM)

- Experimental
  - ASM didn’t scale for inter-domain use, security issues
  - Replaced by source-specific multicast → intra-domain IPTV deployments
Managed IPTV: Multicast Delivery, Unicast feedback

- Evolution of multicast conferencing
- Source-specific IP multicast media
  - Provisioned and managed multicast in edge networks, but not interdomain
  - One multicast group per TV channel
  - Replicates cable TV experience, using low latency, efficient, multicast delivery
  - Provisioned set-top boxes decode media → managed service
- Media transport using MPEG-TS in RTP; unicast quality feedback and repair/catch-up
  - Aggregate reception quality feedback up the tree, giving overall view statistics

- Managed multicast IPTV service can offer very high quality and low latency, but requires provisioning and managed clients – inflexible
Control of Streaming Media: RTSP

- Control protocol for real-time OTT streaming
  - Re-use existing IETF standards: declarative SDP and RTP media flows
  - Control protocol influenced by parallel development of HTTP and SIP
  - Originally media ran on UDP and control over TCP → extensions multiplexed media and control on a single TCP flow for ease of deployment

- Moderate commercial success
  - RealPlayer; 3GPP MBMS
  - Requires custom server infrastructure → expensive and doesn’t integrate with web CDN
  - RTP media over UDP → very low latency; robust; unicast or multicast
Session Descriptions: SDP Offer/Answer

• Declarative SDP works for broadcast
  • Server announces a session
  • Clients join, based on description in announcement

• Interactive sessions require negotiation
  • An offer to communicate: lists codecs, options and addressing details, identity of caller
  • The answer subsets codecs and options to those mutually acceptable, supplies addressing details, and confirms willingness to communicate
  • RTP-based media then flows, peer-to-peer

• IETF re-used SDP as the negotiation format
  • SDP not designed to express options and alternatives
  • Insufficient structure in syntax, semantic overloading
  • Complex → but complexity not initially visible; too entrenched for alternatives to take off
Control of Interactive Conferencing: SIP

• SIP trapezoid – inter-domain conferencing framework
  • SIP provides identity, location, and negotiation
  • Uses offer/answer model with SDP to negotiate media flows, codecs, addressing, etc.

• Initially simple framework, became complex and inflexible
  • Innovation at the speed of standardisation
  • How much complexity is inherent in the problem domain?
    • Multiparty calls inherently complex – option negotiation, addressing, call setup
    • User location and call setup inherently complex – multiple answers for a single user, which to accept?
  • How much due to interoperability with PSTN?
    • Considerable – e.g., early media, fax-over-RTP, DTMF
    • Lessons for standardisation…
Control of Web-based Interactive Media: WebRTC

- Expose standard control API rather than standard signalling protocol – innovation above that JavaScript API, rather than by changing the protocol

- Features:
  - Media transport using modern RTP stack
  - Peer-to-peer data channel: SCTP over UDP
  - Javascript Session Establishment Protocol with custom applications

- Complexity of bundled media, JSEP signalling, and exposed SDP
- Obvious uses and extensions:
  - low-latency live unicast streaming
  - multicast IPTV
Control of Telepresence: CLUE

- SIP extensions for high-quality, multiscreen, telepresence
- The inflexibility of SIP coupled with the complexity of WebRTC bundled media and data channel
Path Discovery: ICE, STUN, and TURN

- Multimedia standards developed before wide deployment of NATs and firewalls
  - Assumed every host had a public IP address, that could be sent via SDP
  - Similar assumption to FTP
- This is no longer accurate – need NAT traversal
  - STUN: determine NAT bindings
  - TURN: relay traffic via public server
  - ICE: systematic algorithm for use of STUN and TURN to find usable path
- Complicates offer/answer
  - Don’t know the addresses to use in the offer until ICE has completed
  - Don’t know candidates to use in ICE until offer/answer has completed
- Essential in modern deployments
Review of Internet Multimedia Standards Development

- Long-term development – evolving standards
  - Network voice protocol (RFC 741; Nov. 1977)
  - Current framework (RTP, etc.): 1992 →

- Architectural focus on reusable protocols
  - Community has not favoured common components
    - Continued fight against point solutions
    - Ad-hoc developments → complexity
  - The architecture was designed for a network that no longer exists
  - Adaptive media, application level framing – very robust, low latency, if you can afford the complexity

- Signalling is harder than everyone realises
An Alternative Architecture: HTTP Adaptive Streaming

• Reaction to the complexity of the Internet multimedia architecture
  • RTSP effective, not economically viable for initial deployments
  • Efficiency and scalability becoming much less critical
  • Lack of understanding of the RTSP, SDP, and RTP stack by web community
HTTP Adaptive Streaming

- Video encoded in multiple chunks
  - Independently decodable; 2-10 second duration; multiple encodings of each at different rates
  - Manifest file provides index
  - Client pull via cache hierarchy (CDN)
    - Monitor download rate, and choose what encoding rate to fetch next
    - Standard HTTP downloads

- Easy to deploy, but challenges:
  - Low latency streaming
  - Rate adaptation for congestion control
  - Impact of HTTP/2
  - Impact of QUIC
  - These are pushing in a direction RTP tried to solve
Concluding Remarks

• HTTP adaptive streaming succeeded because bandwidth is cheap and plentiful – and it could leverage commodity CDN infrastructure

• The Internet multimedia standards trade some complexity for lower latency and robustness to loss
  • Application level framing, with intelligent endpoints
  • That used to make sense, and still does for some use cases – interactive; RTSP+RTP have much relevance for modern streaming

• To develop the next generation video architecture, we need:
  • A de-ossified, multiplexed, path layer above which transport can evolve – WebRTC has conclusively shown the limitations of the current approach
  • Interoperability between different media transport models – a content centric view?