Open Source, real-time AV1 implementation

Dr. Alex

CoSMo / Millicast
I. AOM, AV1, Real-Time and background
AOM

- Based on quality, not quantity (SRT?),
- IP management integrated in the process from the start (avoid H265-like drama)
  - Legal due diligence to start
  - Membership agreement including protecting clauses
  - Legal defense fund to protect users
- An impressive list of Big Corporations.
- Main target: AV1 “bitstream” + encoder
- Secondary target: AV1 RTP payload
AOM :: USE CASES

- VOD, pre-recorded content, almost-live, live (3s): **CODEC ONLY**
  - Netflix, Youtube, ....
  - All the time in the world to encode,
  - Encoding / upload / storage / delivery all separated,
  - Main cost on storage and delivery (bandwidth)
  - Only delivery and decoding is time sensitive
  - Quality is often most important than latency
  - All ecosystem: cloud encoding, hardware encoders, decode-only, players, ...
AOM :: USE CASES

- VOD, pre-recorded content, almost-live, live (3s): **CODEC ONLY**
  - This group work was finished once the bitstream was specified in 2018.
  - The testing part boils down to checking that any bitstream generated by any encoder or transcoder is readable by at least the reference encoder,
  - There is no filtering between the encoder and the decoder.
  - For a lot of companies with that use case, the decoding speed and the encoder quality are the main concerns, but not the encoding speed.
    - It explains why we see so many decoding-only AV1 library out there.
      - Example: support of MPEG-DASH / HLS / CMAF in browsers or players.
AOM :: USE CASES

- **Real-time (<1s):** **CODEC** and **MEDIA TRANSPORT** with **SFU**

  - Cisco Webex, Poly, Vidyo, CoSMo, Facebook Msg/whatsapp, ….
  - Latency is king,
  - Simpler Encoder, single-frame encoding, no B-frames, ….
  - Encoder, Media Server, and decoder must ALL be real-time.
  - Real-Time requires end-to-end control, no storage, …
    - Need to define the Media Transport, which will shoulder some of the RT properties
    - Need to define everything with Media Server logic in mind
    - Codec is not enough.
  - Deliverable => RTP payload and Header Extensions
  - Non deliverable but needed IRL => BWE, CC, FEC, RED, RTX, ….
  - Test is challenging, since we now need end-to-end testing in RT conditions.
AOM :: USE CASES

- Real-time (<1s): **CODEC** and **MEDIA TRANSPORT** with **SFU**
  - This is more complicated
  - It could only really start implementing and testing once the main codec group was done
  - While encoder and decoder are apps, RTP in itself is not an app. What app to use?
  - Beyond the simple RTP encapsulation, management of SVC and SFU is the main target
  - Testing is much more challenging
  - Potentially some corner cases will be found wherever the SFU filtering generate bitstream that are compliant with the spec but never tested before (if it’s not tested, it’s broken).
AOM :: RT :: USE CASES

- Video Conference: e.g. Cisco Webex
  - Duplex
  - Everybody’s equal
  - But the Active speaker is more interesting
  - Optimizations possible based-on voice activity detection, and Active Speaker
  - Echo cancellation mandatory,
  - Scaling is quadratic with respect to the number of users.
  - Cascading is possible but not mandatory
  - “The cisco dilemma”: supporting as much as possible existing hardware-based devices.
Streaming: e.g. MilliCast

- One-way
- Source and viewers with very different logic and capacity
- No scaling optimization possible like in VC
- Scaling is linear with respect to the number of users.
- Cascading of servers is almost always needed
- Real challenges to keep quality and network resilience at scale
AOM :: RT :: USE CASES

- In p2p mode, there is no difference between VC and streaming
- In 1 server mode, there is also no difference
- When you start serving more than 1,000 viewers, and/or need more than one media server in the media path, things start becoming … interesting.
AOM :: RT :: Bandwidth Adaptation!

Sender-side (SVC) vs server-side (ABR)

- Both achieve the same thing: make several resolutions of the source stream available so that the viewers can switch between them depending on their capacity (Bandwidth, screen size, CPU, ...).

However

- Sender-Side is twice faster,
- Sender-Side allow for faster switching between resolutions,
- Sender-Side Enable both DRM and end-to-end encryption, transcoding step in ABR prevent any E2EE.
Bandwidth Adaptive Media Streaming Pipeline in practice - the usual
Bandwidth Adaptive Media Streaming Pipeline in practice - WebRTC end-to-end

No storage

1 less enc/dec = 50% load
Reminder: Multi-cast vs Simulcast vs SVC

**Multicast**
- Several tracks
- Decodable separately
- Bandwidth management separated

**Simulcast**
- Several tracks
- Coming from the same source
- Decodable separately
- Smart bandwidth management possible

**SVC Encoding**
- Several tracks
- Coming from the same source
- Not Decodable separately (Except base layer)
- Smart bandwidth management mandatory
- Less bandwidth, more resilience.
Simulcast: Use case for WebRTC 1.0

- Use case for WebRTC 1.0: SFU

- Browser send simulcast, does not receive simulcast (in WebRTC 1.0)
Recent history of AV1 with focus on RT

- 01 2018, Apple joins AOMedia.
- 03 2018, AOMedia announced the release of AV1 along with its reference implementation: libaom.
- 09 2018, chrome 70 and Firefox nightly had added some kind of support for decoding / playing AV1.
- 10 2018, CoSMo Software announced the first AV1 integration in RTP and WebRTC. Not real-time, no SVC support.
- 12 2018, AOMedia Sponsored dav1d encoder has been released. It is included e.g. in Firefox67, ....

- 01 2019, CoSMo Software joins AOMedia.
- 03 2019, Samsung joins AOMedia.
- 04 2019, INTEL and NETFLIX, announced their collaboration around the SVT-AV1 open-source codec.
- 04 2019, Allegro DVT announced its AL-E210 multi-format video encoder hardware IP, the first (?) hardware AV1 encoder.
- 05 2019, Realtek announced the RTD2893, its first integrated circuit with AV1 decoding, up to 8K.
- 06 2019, it announced the RTD1311 SoC for set-top boxes with an integrated AV1 decoder.
- 06 2019, Cisco demo of the first Real-Time AV1 integration in RTP and WebRTC (webex). No SVC, not open-source.
- 07 2019, CoSMo Software release a demo of Real-Time AV1 integration in RTP and WebRTC. Also no SVC.
- 08 2019, AV1 SVC Availability in MilliCast is announced at IBC, along with RT SSAI.
II. AV1 RTP Payload

- AV1 OBUs <==> RTP packets
  - Modes
  - Fragmentation
  - Reconstruction

This is trivial. The real deal is to use all the RTP features to enable SVC and network errors resilience.

- Extend AV1 modes: K-SVC
- Simplify the decoding / filtering: “DTI” Decoding Target Information
- Help Filtering without reading payload, Manage Encrypted payloads (EEME)
II. AV1 RTP Payload: OBUs

Temporal Units and Spatial Scalability
Operating Point Bitmask

The AV1 specification defines an operating point as a 12-bit mask that combines temporal and spatial IDs: S3 S2 S1 S0 T7 T6 T5 T4 T3 T2 T1 T0

Example: L2T2

<table>
<thead>
<tr>
<th>Nr.</th>
<th>S3</th>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>T7</th>
<th>T6</th>
<th>T5</th>
<th>T4</th>
<th>T3</th>
<th>T2</th>
<th>T1</th>
<th>T0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Low resolution, half fps</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Full resolution, half fps</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Low resolution, full fps</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Full resolution, full fps</td>
</tr>
</tbody>
</table>
S2T3: Two simulcast streams each with 3 temporal layers
II. AV1 RTP Payload

- AV1 OBUs <=> RTP packets
  - Modes
  - Fragmentation
  - Reconstruction
- Extend AV1 modes: K-SVC
- Simplify the decoding / filtering: “DTI” Decoding Target Information
- Help Filtering without reading payload, Manage Encrypted payloads (EEME)
Benefits Of Shifted Temporal Prediction Structures

- More balanced bit distribution on the wire (reduced congestion/delay)
  - Rate distribution with concurrent TLO’s: 54%, 15%, 16%, 15%
  - Rate distribution with shifted TLO’s: 36%, 21%, 28%, 15% (variance ratio 4.5:1)

- More balanced CPU usage at encoder
Example K-SVC mode: L4T7_KEY_SHIFT
II. AV1 RTP Payload

- AV1 OBUs <=> RTP packets
  - Modes
  - Fragmentation
  - Reconstruction
- Extend AV1 modes: K-SVC
- Simplify the decoding / filtering: “DTI” Decoding Target Information
- Help Filtering without reading payload, Manage Encrypted payloads (EEME)
Example: L2T3

- At Time 3, SFU decides to move from full resolution quarter framerate operating point to full resolution, full frame rate.
  - Problem: P2 cannot be forwarded if P1 wasn’t forwarded, S2 cannot be forwarded if S1 wasn’t forwarded. Need to wait for S0 switching up point ("B" flag set).
  - Can we determine this from the framemarking header alone (e.g. no access to operating point bitmask or scalability structure)?
  - “Discardable” marking for frame P1 depends on the operating point, which is not provided.
To the rescue: Decode Target Information

- **Decode target**: The set of frames needed to decode a coded video sequence at a given spatial and temporal fidelity.
- **Decode Target Information (DTI)**: Describes the relationship of a frame to a Decode target.
  - ‘not present’: The frame does not belong to the Decode target.
  - ‘discardable’: The frame will not be a referred frame for any frame belonging to that Decode target.
  - ‘switch indication’: all subsequent frames for that Decode target will be decodable if the the frame containing the indication is decodable.
  - ‘required’: The frame belongs to the Decode target and has neither a Discardable nor a Switch indication. A frame belonging to more than one Decode target may be Required for one Decode target, but not for another Decode target.
III. Open-Source Implementation
Minimum RT system (p2p)
AV1 Minimum Open Source System (p2p)
CoSMo implementation:
Code Name: DarkArtsWebrtc

- Libaom is the reference (compliance)
- It is also a production-quality library used e.g. by Youtube.
- New real-time mode.
CoSMo implementation:
Code Name: DarkArtsWebRTC

Codec support
libwebrtc
libaom
CoSMo
CoSMo implementation:
Code Name: DarkArtsWebrtc

RTP Engine
libwebrtc

GHD

Goog

AV1 Payload

“Simple” RTP payload
May be recoded by Google

SVC support
System Under tests (p2p)

Modified Libwebrtc
With AV1 support

Test the basics: packetization, unpacketization, all modes (unmodified), ....
System Under tests (SFU)

Test SFU based filtering, layer changing, etc ...

- Dedicated native app
- Sig. server (meedoze)
- Dedicated SFU (meedoze)
- Dedicated native app

Media

Media
KITE: SUT instrumentation + Testing automation

KITE Design

CoSMo Software
Test Scenarios

- Start with existing webRTC test suite and adapt
  - Simulcast / SVC
  - Layer switching logic
  - ...

- Could we test the matrix of all possible filtering exhaustively in reasonable time?
When will AV1 in webrtc be available?

- Tentative
  - 2 months for RTP payload + GHD usage first implementation upstream
  - 3~4 months of intense testing to make sure all modes and corner cases are working, and update the specification accordingly

End of Q1