RTP Topology Considerations for Offer/Answer-Initiated Sessions

`draft-lennox-avtcore-rtp-topo-offer-answer-00`

AVTCore, IETF 83, 26 March 2012

Jonathan Lennox

jonathan@vidyo.com

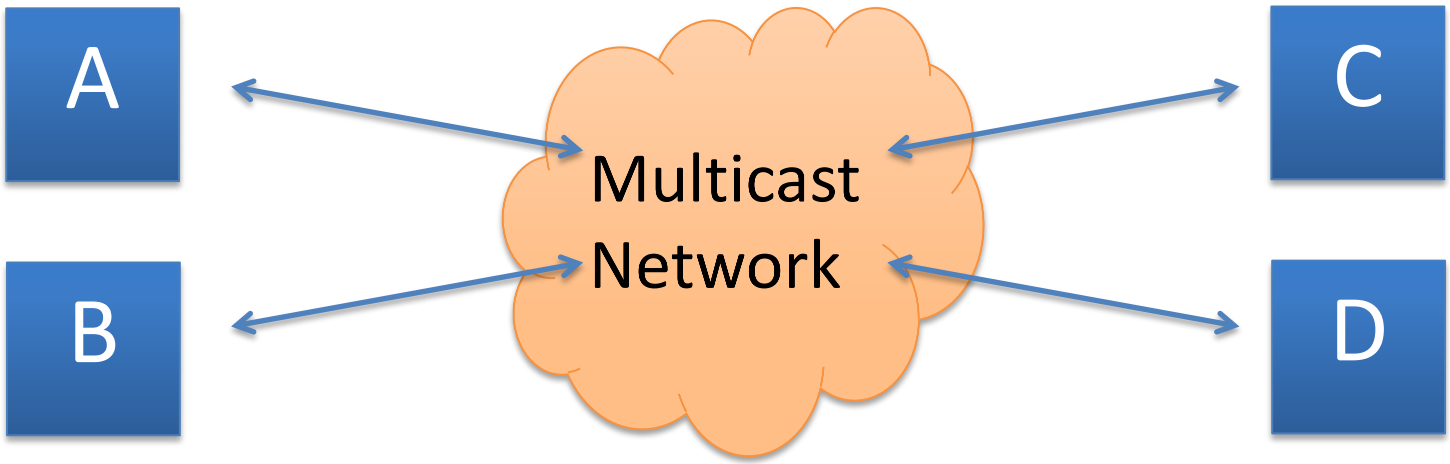
Offer/Answer in one Slide

- SDP Offer/Answer has two modes
  - Multicast
    - Single view of session
    - If transport address is multicast
    - Not used very much
  - Unicast
    - Two separate views of session
    - Mostly, each side specifies what it wants to receive
      - (Oversimplification)

RTP Topologies (RFC 5117)

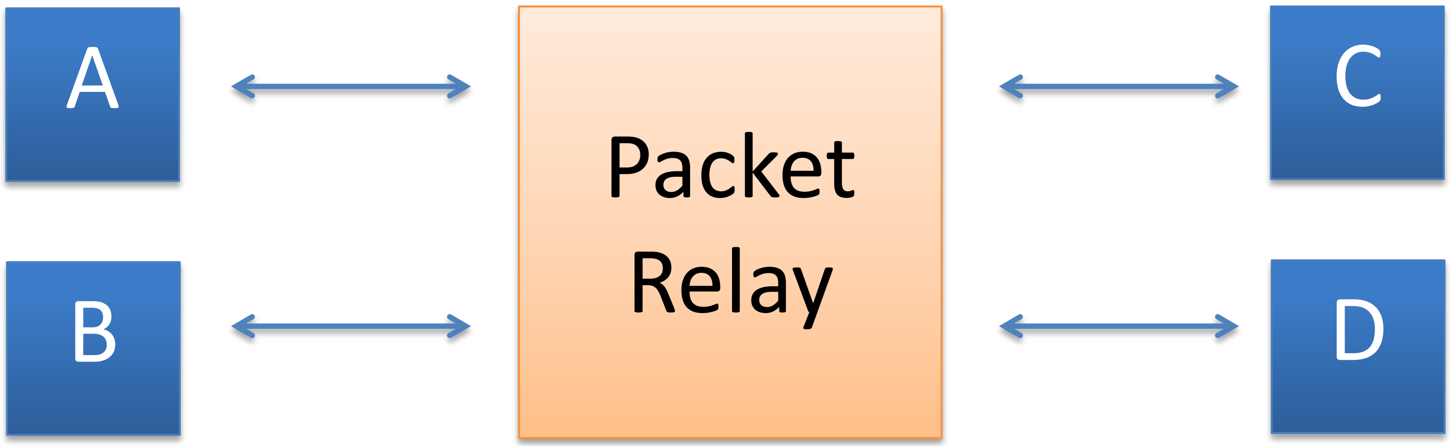
- RTP Sessions come in many topology types
- Topo-Multicast corresponds to Offer/Answer Multicast mode
- Topo-Point-to-Point corresponds most directly to Offer/Answer Unicast mode
- Other RFC 5117 topologies can also be assembled from Offer/Answer Unicast …
- … except for Topo-Transport-Translator!

Topo-Multicast



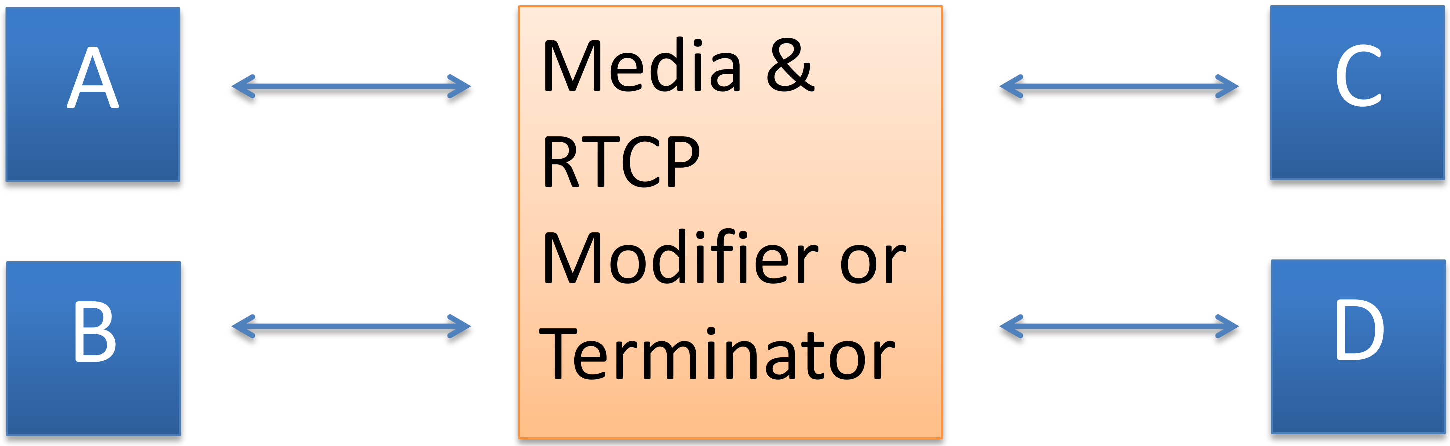
- Network forwards packets among all participants.
- Single view of session.
- SDP Offer/Answer Multicast mode.

Topo-Transport-Translator



- Middlebox forwards packets among all particpants.
- Single view of session.

Other RTP Topologies



- Topo-Media-Translator, Topo-Mixer, Topo-Video-Switch-Mixer, and Topo-RTCP-Terminating-MCU.
- Equivalent for this analysis.
- Middlebox modifies and forwards media and RTCP.

Why Offer/Answer doesn't work for Topo-Transport-Translator

- Topo-Transport-Translator assumes a single view of the session.
  - A middlebox that just forwards all RTP and RTCP, without modifications.
  - At session layer, looks almost identical to Topo-Multicast.
  - Everyone shares the same session configuration.
- But as mentioned: Unicast Offer/Answer allows each endpoint to establish its own view of the session.

In particular (1): bandwidth

- An SDP offerer or answerer can specify a bandwidth (b= parameter) for any media stream (RTP session), indicating the bandwidth it wants the peer to use to send.
- This is independent of the peer's specified bandwidth.
- RTCP timings (and thus RTP membership timeouts) are a function of SDP b= values (AS, RS, RR…).

In particular (2): media types

- Participants can remove media types from the answer or updated offer.
- Participants can change (some) fmtp parameters.
- Though NOT RECOMMENDED, they can change the mapping between media types and RTP payload type numbers.
  - And there are sometimes good reasons to do this.

And others…

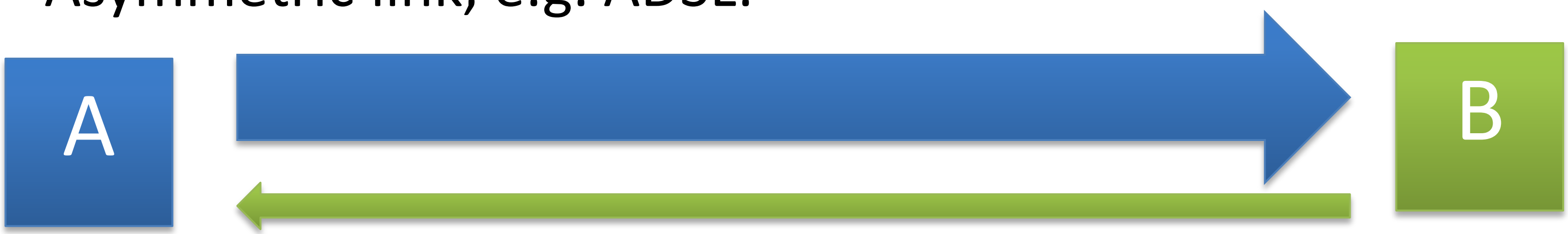
- Other reasons too, such as ptime or many SDP extensions.
- Don’t need to get into them now.

But this is a good thing!

- If you know that your media and RTCP is always received by only one other party …
  – Whether it’s consuming it directly, or forwarding it after re-writing
- Some very useful optimizations are possible

Offer/Answer Asymmetric Bandwidth

- Asymmetric link, e.g. ADSL:



- Bandwidths in each direction are very different, and non-rival at the bottlenecks.
- What RTCP timeout should you use?
- B sending RTCP less often because A has multiple senders makes no sense.
- If you can assume a single receiver: each party uses 5% (or whatever) of its own bandwidth for RTCP, calculates remote timeouts accordingly.

RTCP receiver reports

- For the benefit of Topo-Multicast and Topo-Transport-Translator, in RTCP, every session member sends receiver reports about every sender.
- This means (in a many-to-many session) the number of reception reports is quadratic in the number of session members.
- If RTCP is going directly to only one other party, this is useless.
  - A middlebox is likely doing its own retransmission and repair, and thus directly consuming a large fraction of the RTCP.
  - Excess reports consume RTCP bandwidth which could be used for timely feedback of relevant data.

RTCP receiver reports (2)

- If you have multiple co-generated sources (e.g. CLUE, Bundle, RTX)
  - Having them send reports about each other is pointless: zero loss, jitter.
  - Having them send all send redundant reports about remote sources isn't helpful.
- Reception reports from remote source A about another remote source B are rarely interesting or useful.

Normative recommendations

- For RTP sessions negotiated with unicast SDP offer/answer:
  - RTCP bandwidths, and timeouts, MUST be calculated independently in each direction.
  - Endpoints SHOULD NOT send reports from one of their own sources about another.
  - Endpoints SHOULD pick a single “reporting” source to send reception reports for each remote source.

Way forward

- What do people think of this?
  - I think this is important for BUNDLE, CLUE, some WebRTC models, and anything else doing source multiplexing…
  - The bandwidth issues apply even for single-source cases.
- Would this need to be a normative update to 3550? To 3264? Or just implementation guidance?
- Should this actually be more than one document, addressing different issues?