Cultural Learnings of ICE/DTLS

IETF 92

Questions

- What is an ICE "virtual connection"?
- Do existing RFCs say the right thing about using DTLS with ICE?
- What causes a new DTLS handshake?

Terminology

- 5245 sayeth:

> Component: A component is a piece of a media stream requiring a single transport address; a media stream may require multiple components, each of which has to work for the media stream as a whole to work. For media streams based on RTP, there are two components per media stream -- one for RTP, and one for RTCP.

- Not the clearest definition, but from its usage in 5245 it's clear this is what we are looking for.
- Note that "selected pair" is a property of a component, and NOT the same thing.

RFC5763

- 5763 gets it right:

6.7.1. ICE Interaction

Interactive Connectivity Establishment (ICE), as specified in [RFC5245], provides a methodology of allowing participants in multimedia sessions to verify mutual connectivity. When ICE is being used, the ICE connectivity checks are performed before the DTLS handshake begins. Note that if aggressive nomination mode is used, multiple candidate pairs may be marked valid before ICE finally converges on a single candidate pair. **Implementations MUST treat all ICE candidate pairs associated with a single component as part of the same DTLS association. Thus, there will be only one DTLS handshake even if there are multiple valid candidate pairs.** Note that this may mean adjusting the endpoint IP addresses if the selected candidate pair shifts, just as if the DTLS packets were an ordinary media stream.

RFC5764

- 5764 ignores ICE altogether:

> **5.1.2. Reception**
>
> In some cases, there will be multiple DTLS-SRTP associations for a given SRTP endpoint. For instance, if Alice makes a call that is SIP forked to both Bob and Charlie, she will use the same local host/port pair for both of them.
>
> ...
>
> Because DTLS operates on the host/port quartet, the DTLS association will still complete correctly, with the foreign host/port pair being used, to distinguish the associations.

Conclusion

- Since 5763 is clear on this point, not sure anything needs to be said
- Could restate in rtcweb-transports to make it abundantly clear

Can app request new DTLS handshake?

- TLS 1.3 prohibits rehandshake; discouraged in earlier versions as well
- No SDP language to control doing a new DTLS association; 5763 forbids a=connection:new
- Suggest that we simply not support this at all

What about forking?

- Offerer could get answer from both A and B
  - Demuxing is not a problem; an ICE component with unique lfrag:rfrag and corresponding DTLS association can be created for both A and B
- In WebRTC, a PRANSWER from A will establish ICE/DTLS to A; the ICE/DTLS state will be discarded upon a PRANSWER from B
  - Implies that new remote ufrag in answer causes existing ICE/DTLS context to be discarded (only for initial offer?)

What about certificate change?

- If a new fingerprint is signaled (e.g. 3PCC), this will require a new handshake
  - Simplest way to handle this is via INVITE-with-replaces, which would be a new PeerConnection in WebRTC
- Can also be done via DTLS break-before-make
  - Upon receiving offer with new fingerprint, discard DTLS context and switch to new DTLS association immediately
  - Recommend that WebRTC not support this approach