

ICE AND WEBRTC

draft-thomson-mmusic-ice-
webrtc-01

- ▶ ICE makes the following assumptions:
 - ▶ that consent cannot be revoked
 - ▶ that there is only one ICE agent operating
 - ▶ that the signaling is created by an entity that is acting in good faith
- ▶ Only the first is being addressed

BAD ASSUMPTIONS

- ▶ Browsers allow for concurrent ICE agents
 - ▶ In the same tab/origin to accomplish varied tasks
 - ▶ Cross tab/origin
- ▶ Agents may be unaware of each other, even in the same tab
- ▶ Multiple ICE agents competing cause
 - ▶ Increased check volume
 - ▶ NAT bindings might be dropped (*research continuing)

CONCURRENT ICE AGENTS

- ▶ Bad signaling opens up interesting possibilities
 - ▶ e.g., A large ufrag can inflate the size of a check significantly
 - ▶ e.g., Adding bogus candidates can increase the number of checks
- ▶ In WebRTC we have to assume that the signaling is bad
 - ▶ We can't allow applications to cause browser to misbehave
 - ▶ Warning! Using ICE doesn't require user consent or action

BAD SIGNALING

- ▶ Quick calculations
 - ▶ 100 candidate pair limit
 - ▶ x A check every 20ms
 - ▶ x 384 (or 404) byte checks
 - ▶ x number of ICE agents
 - ▶ = A **lot** of packets (my current record is almost 3Mbps)
- ▶ That's assuming constant pacing; actual numbers can be higher

WHAT COULD POSSIBLY GO WRONG?

- ▶ Cap bandwidth, globally
- ▶ Calculations in the draft
- ▶ Attempt to define “legitimate use” for 1 Agent
 - ▶ “legitimate use” might be 64kbps
 - ▶ Suggested cap: 96kbps

OPTION 1: OOPS, HACK

- ▶ Define global pacing for all ICE agents
- ▶ This introduces some interesting interaction problems
- ▶ RTO needs looking at (ICEbis work perhaps?)

OPTION 2: HARD WORK

- ▶ RTO is calculated such that initial checks all go out before any retransmissions start
 - ▶ Not that many implementations respect this
- ▶ Competition between agents could delay RTO in unpredictable ways if this rule is observed
 - ▶ Either way, competition is potentially bad

FIX RTO

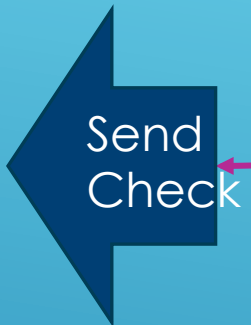
Pacing Timer



B'. if no checks outstanding

A. Normal Candidate Pairing Process + Asynchronous Trickling

B. if available



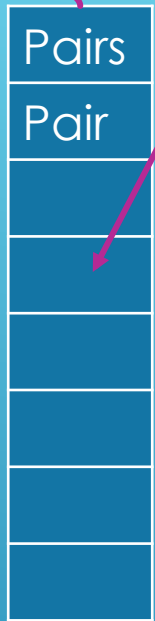
RTO Timer



C. On RTO



Candidate Pairs Awaiting Checks



Candidate Pairs Ordered Strictly By Priority

TRICKLE COMPATIBLE ALGORITHM

- ▶ Concurrent ICE agents compete
- ▶ Need to ensure that one tab/origin can't starve others out
- ▶ May want to hide activity from other origins
 - ▶ Definitely want to hide connectivity check status, but relying on the ufrag/password being different should suffice

DEALING WITH CONTENTION