ICE Mobility at IETF 96

Network Cost, Renomination, and Candidate Removal
Big question

How much incremental improvement to ICE is the WG interested in standardizing?

1. Network cost
2. Renomination
3. Candidate removal
4. ???
**TURN Mobility is coming**

<table>
<thead>
<tr>
<th>Versions</th>
<th>00</th>
<th>01</th>
<th>02</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>draft-ietf-tram-turn-mobility</strong></td>
<td>00</td>
<td>01</td>
<td>02</td>
</tr>
</tbody>
</table>

- **Types**: Active Internet-Draft (tram WG)
- **Last updated**: 2016-07-12 (latest revision 2016-04-04)
- **Stream**: IETF
- **Intended RFC status**: Proposed Standard

**Stream** | **WG state** | Submitted to IESG for Publication
ICE Mobility

1. Use only TURN candidates
2. Use TURN mobility

Done!
ICE Mobility is needed (goals)

- Prefer cheaper (WiFi) over expensive (cell)
- Switch back and forth over time
- Minimize disruption when switching
- Avoid too many TURN candidates
General Choice

Option A: Improve ICE incrementally

Option B: Call ICE "done" and start over with ICENG

Option C: Both
Incremental Improvements

1. Network cost
2. Renomination
3. Candidate removal
Network Cost
Problems

- If 4 candidate pairs work, which do you select?
  - WiFi <-> WiFi
  - WiFi <-> Cell
  - Cell <-> WiFi
  - Cell <-> Cell

- If the remote side *changes* the network interface using TURN mobility, how do you know?

- If you re-select a different candidate pair, how do you know if the network interface changed? (BWE)
Solutions

● New candidate attribute signaled:
  ○ a=candidate .... network-id=X network-cost=Y

● New candidate STUN attribute
  ○ COST = network-id, network-cost
  ○ 2 uint16s
Why not candidate priority?

- Forces change in check order
  - We want to be able to order by priority and select by cost
- Can't change (without ICE restart)
- Network cost wants to be more important than all the other things with candidate priority
- ...
Renomination
Problems

- You can only nominate once
- Backup candidate pairs not possible
- Continual (un-ending) gathering not possible
- Timing of nomination tricky with passive-aggressive
  - Too soon, stuck with worse pair
  - Too late, no convergence for a while
Solutions

- New ICE option
  - a=ice-options=renomination
- New STUN attribute
  - NOMINATE = X
  - On controlling side, increment X
  - On controlled side, biggest wins
Candidate Removal
Problems

- Continual (never ending) gathering piles up candidates forever
- With WiFi/cell and ipv4/v6, there are lots of TURN candidates
  - TURN mobility is a nice solution, but requires server support
  - Removing candidates can help with a client-only change
Solutions

- PeerConnection.oncandidateremoved
- PeerConnection.removecandidate
Implementation experience

We've implemented these things (mostly) and deployed them in limited form, and it seems to work well so far.
Big question

How much incremental improvement to ICE is the WG interested in standardizing?

1. Network cost
2. Renomination
3. Candidate removal
4. ???