mmusic
Internet-Draft
Intended status: Informational
Expires: January 7, 2016

C. Jennings
Cisco

July 06, 2015

Proposal for Fixing ICE
draft-jennings-mmusic-ice-fix-00

Abstract

This draft raises some issues and proposes some directions for improving ICE. It is never meant to become an RFC but is for WG discussion.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 7, 2016.

Table of Contents

1. Problem Statement

ICE was developed roughly ten years ago and several things have been learned that could be improved:

1. It is spectacularly difficult to debug on analyze failures or successes in ICE or develop good automated tests. Many implementing have had significant bugs for long periods of time.

2. It is timing depended. It relies on both sides to to do something at roughly the same time and that ability to do this goes down with the number of interfaces and candidates being handled. Mobile interfaces and dual stack make this worse.

3. Many widely deployed devices do not implement aggressive nomination.

4. Many widely deployed devices do not implement normal nomination.

5. It does not discover asymmetric routes. For example UDP leaving a device may work just fine even thought UDP coming into that device does not work at all.

6. Almost all deployments include a TURN server. At the time ICE was done it was not understood if this would be too expensive or not so ICE works without TURN but better with it.

7. The asymmetric of the controlling / controlled sides has caused many interoperability problems and bugs.

8. Priorities are complicated in dual stack world and ICE is brittle to changes in this part of the algorithm

2. Possible Solutions

Lets consider what we could do if both sides had a back channel that always worked for sending information to the other side. I'll discuss how we get a back channel later in the draft.

Each side is responsible for discovering a viable path for it's incoming media. Each end sends media to where the other sides tells it. There is no controlling concept or aggressive or normal nomination. The candidates are exchanged as in normal ICE along with some attribute that explains the extensions in this draft are supported.

If both sides support this, each side performs it's check by sending outbound STUN pings to open local pin holes, then tracks the life times of theses and reopens them before they expire. It then tells the other side using the back channel to send it a STUN ping from a given location to one of a specific candidates. If this works, it knows it has a viable path.

One side can tell the other side which path to use (or not use) over the back channel. This can happen at any point of time time and there is not concept of ICE is "done", ice is running while the connection is up and at any point in time there some number (or perhaps zero) viable paths for media in each direction.

There are alms no case where ICE works but that TCP data through the TURN servers does not. The candidates that go though the TURN server would be used for the back channel and they would be assumed to work. There are environments where this back channel would not work but in those cases, it is very unlikely ICE would work.

Many parts of ICE were designed to support finding multiple ports for RTP / RTCP , voice video and so on. At this point in time we should fully optimize for just a single port and assume bundle and rtcp-mux.

A variation of this could be each side discover a path for it's outgoing media. Not clear where is best.

3. Advantages

3.1. Diagnostics

This makes it very easy to see which outbound connection were sent from side A to open a pin hole, then when side A asked B to send a test PING and if B received that. It becomes easier to set up a client with an automated test jig that tests all the combinations and

makes sure they work as you only need to test receiving capability and sender capability independently.

3.2. Timing

This more or less removes the timing complexity by allowing both sides to be responsible for their own timing. If it turns out that we can pace things much faster than 20ms (see Juberti's tests results when they come out), then this allows us to take advantage of that without both sides up grading at the same time. If we end up with a lot more candidates due to v6, mobile etc, this removes the issue we have today where a path might have worked but the two sides did not find it due to timing issues.

3.3. Asymmetric Media

This allows media to be sent in one direction over a path that does not work in the reverse direction.

4. Backwards Compatibility

At IETF 92 I thought it would be possible to design a backwards compatibly solution that did roughly this. That might be possible if all the major implementations fully implemented the current ICE spec but many of them do not. Even worse, they implement different parts. My proposal here is more or less make an ICE2. ICE2 advertises the same candidates as ICE but also adds some SDP signaling to indicate the device supports ICE and ICE2.

In the short term we would need device such as web browsers would be requires to support ICE and the ICE2 extensions here but in the future we could move to devices that only did ICE2.

The main mechanisms between ICE and ICE2 are largely the same but the way paths are chosen and used is somewhat different.

Author's Address

Cullen Jennings
Cisco

Email: fluffy@iii.ca