

# ICE Timing

## IETF 96

draft-jennings-ice-rtcweb-timing-01  
Cullen Jennings

# Disclaimer .....

Have you read it ?

Bunch of details about what browser plan to do are likely wrong.

Size estimates based on what browsers are doing not what spec says so are a bit small

# Back at last IETF .....

## Concerns with Changing $T_a$ from 20 ms to 5 ms

Two concerns

- 1) Can NATs created new mappings at this rate
- 2) Does it use too much bandwidth

# Tests

Ran on latest fastest high end NATs from Asus, DLink, Netgear, and Linksys

Key to understand this does not represent what is deployed today but what we might encounter in the future

# Mapping Results

Mapping speeds > 1000 mappings / second

Should work fine with a Ta at 5ms

## Bandwidth Results (evil JS)

Create 100 PC in parallel each with one data channel

Send the stun traffic to server under attack

Result: Creates too much traffic

Proposal: Need a global rate limit of non congestion controlled bandwidth

Chrome limits to 250 kbps and Firefox plans to do the same

## Bandwidth Results (not evil JS)

A single PC with 72 pairs to test and  $T_a=5$  ms will peak around 480 kbps

- This is a problem in that the 250 kbps limit will drop lots of STUN packets causing reduced ability to connect

Single PC, 72 pairs and  $T_a=25$  peaks about 180 kbps

- This works for one PC but not real great for multiple PC

(based on simulation courtesy of EKR - see sim.py at <https://github.com/fluffy/stunTiming/> )

# Global Rate limit Problems

If A and B are doing checks at different rates, there will be skew on timing of suicide packets

If the skew is too large, NAT closes mapping before incoming packet arrives

We don't have good measurements of how long this is (antidotal info is 1 second on some firewalls but likely longer on residential NATs )

# Proposal

- Add back to ICEbis the concept of a global limit
  - Set a baseline (perhaps 250 kbps) that is used unless a specific usage overrides it
- Concurrent ICE agents suffer
  - Let different usages pick different recommendations (rtcweb might recommend that apps that need lots of concurrent PeerConnections to do things like "Use TURN-only first then stagger non-TURN connections")
- Add to SDP a way for both side to indicate min Ta then they use the min acceptable to both sides (mmusic issue)
- Continue work on shaved ICE to reduce size