

ICE-09

Jonathan Rosenberg
Cisco Systems

What is ICE-9?

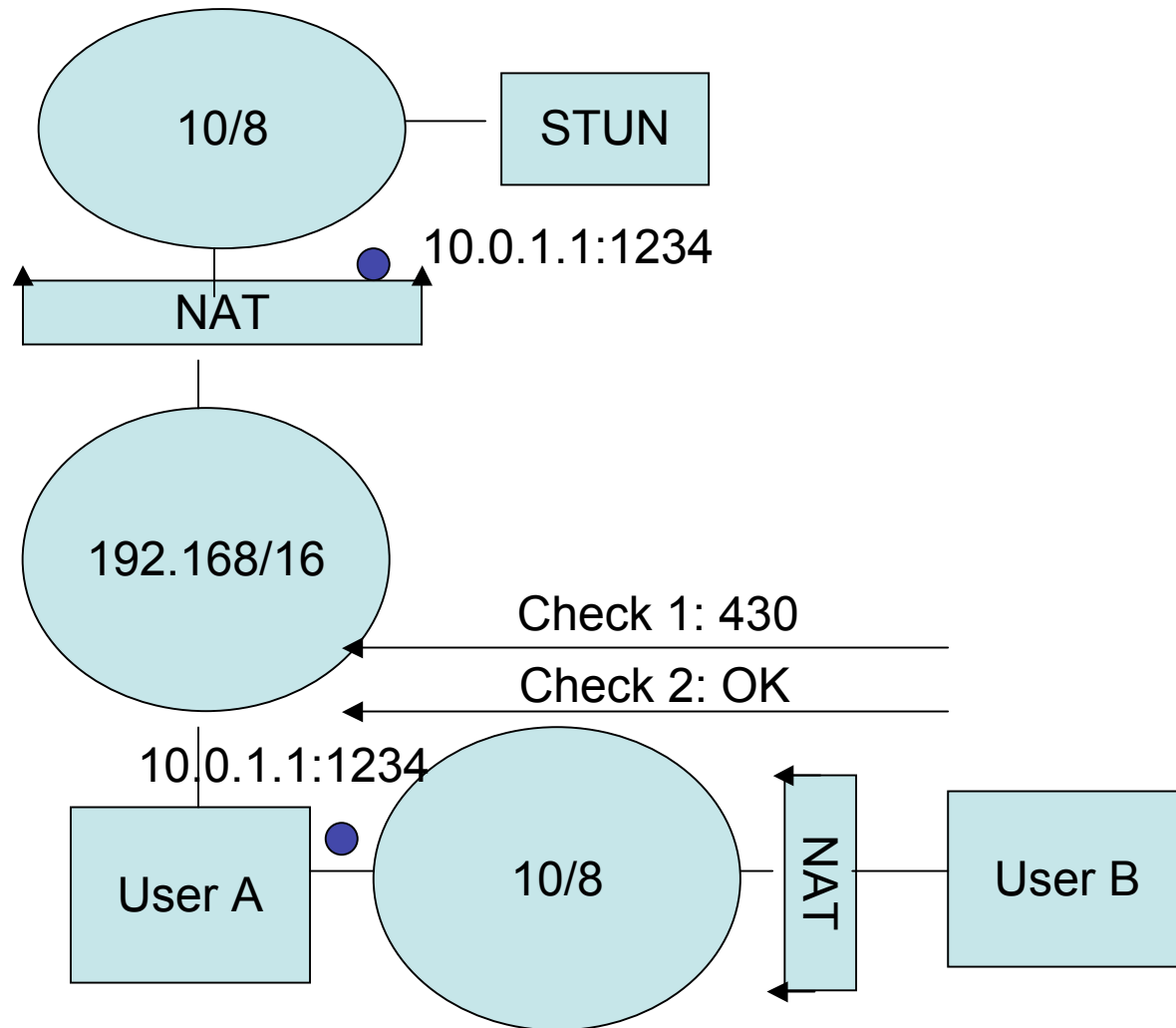
- **Ice-nine** is a fictional material conceived by [science fiction](#) writer [Kurt Vonnegut](#) in his novel [Cat's Cradle](#). It is supposed to be a special [allotropic](#) form of [ice](#) that only melts at 114.4 degrees [Fahrenheit](#) (45.8 °C); when it comes into contact with liquid water, it acts a crystal "seed", and will [catalyze](#) the solidification (freezing) of any normal water at ambient temperatures—thus being capable of destroying the world as we know it.

- Wikipedia

Changes since -08

- Extended SDP candidate to indicate type of candidate
 - Diagnostics
 - Was needed for ICE processing for server reflexive
 - Needed for QoS mechanisms in 3gpp/Packetcable for relayed
- ‘Optimizations’
 - Prune candidate pair list if native addresses share the same origination address
 - Allow a connectivity check for one pair to validate another pair, which then invalidates the one that was checked
- 430 if request arrived on wrong local transport

430 Use Case



Changes

- Peer derived candidate checks
 - Previously – only start once all components showed up
 - Now, starts as soon as each shows up
- Candidate promotion to active description is simplified – based on states of elements higher on priority list
- Terminology change:
 - Active to operating
 - Avoids overlap with ‘active’ from tcp land
- A=remote-candidate used for selecting candidates in response, not just m/c line
- $T_{ws} = \max(0, 100\text{ms} + \text{RTT} - N * 50\text{ms})$
- T_b timer for connectivity checks, different for T_a for gathering
 - $T_b = 50\text{ms} * N$
 - Randomization factor to avoid sync
- Can't change passwords arbitrarily – all addresses must change

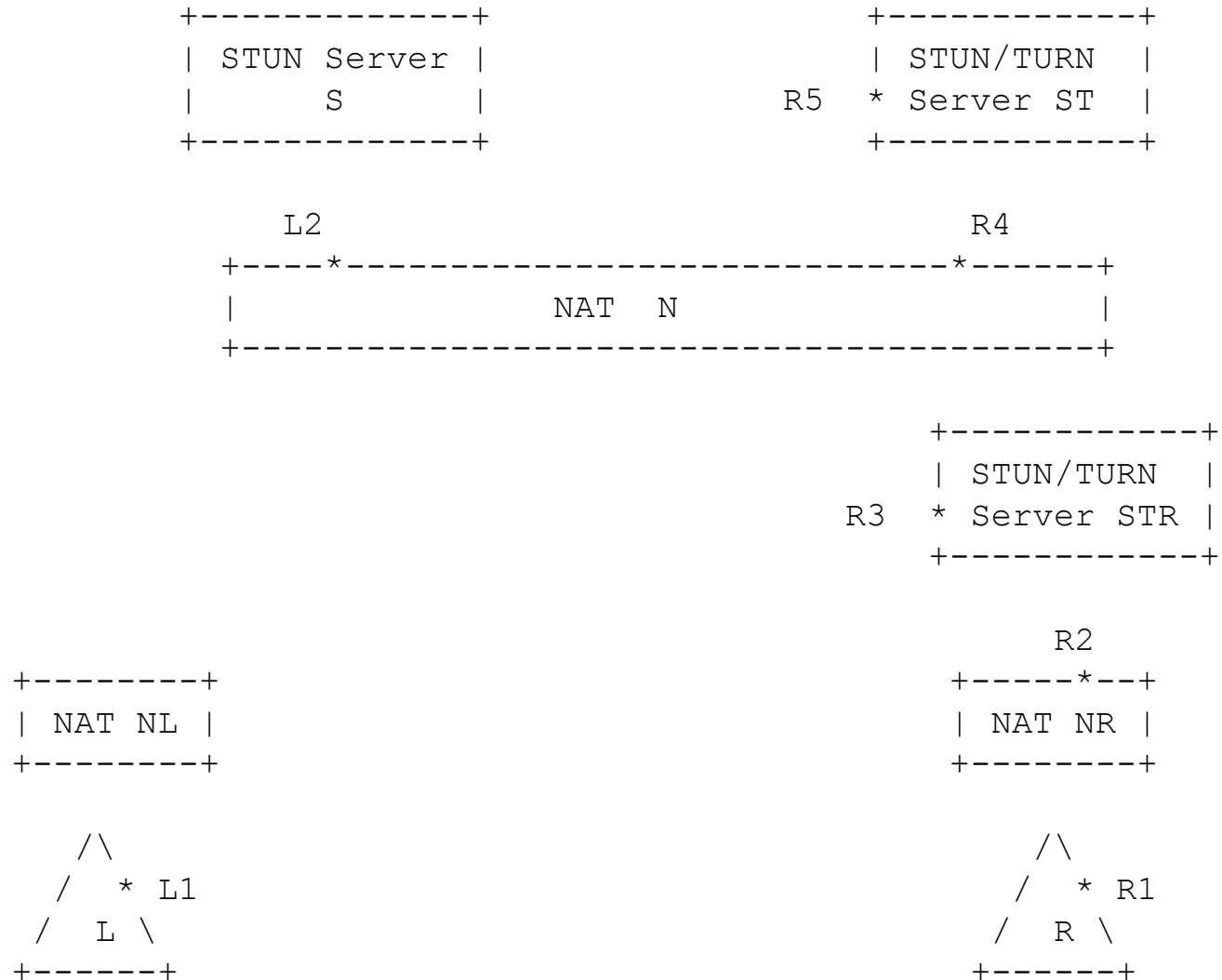
Changes

- To make sure keepalives are sent to non-ICE peers, use malformed RTP if nothing else
- Added some text to handle case of a NAT reboot mid-check
- Minor tweaks so that ICE through b2bua ends up using m/c-line
- Massive introduction rewrite
- Relaxed restriction on using ephemeral ports only for local candidates
- Clarified that candidate attributes with same candidate ID don't need to appear together
- Clarified ice-pwd I hand out is the one I expect to receive

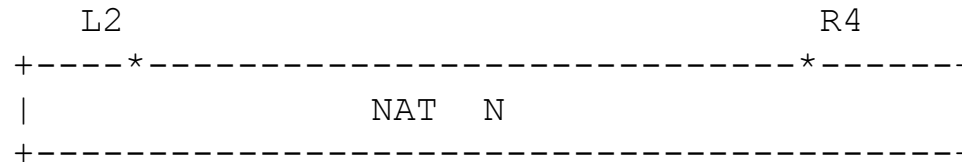
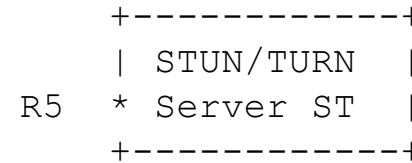
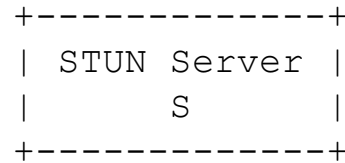
Open Issue #1: SBC Case

- What happens if an SBC is in the path and not ICE aware?
- Two likely possibilities
 - Changes m/c line and removes candidate attributes (case 1)
 - Changes m/c line and passes candidate attributes (case 2)
- Case 1 – will end up as if ICE was not there
- Case 2 – what should happen?
 - Proposal: end up as if ICE was not there
- Rationale
 - Don't want ICE to be an SBC bypass protocol
 - Should fallback to 'backwards compatibility' mode since network apparently doesn't support ICE
- Proposal
 - Keep current approach

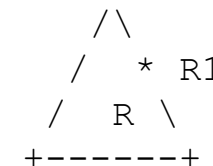
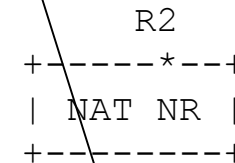
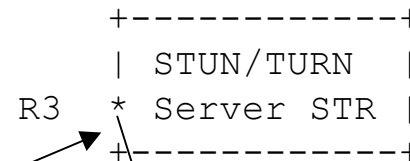
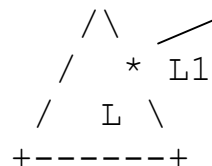
Open Issue #2: Pairing Peer Derived with other Candidates



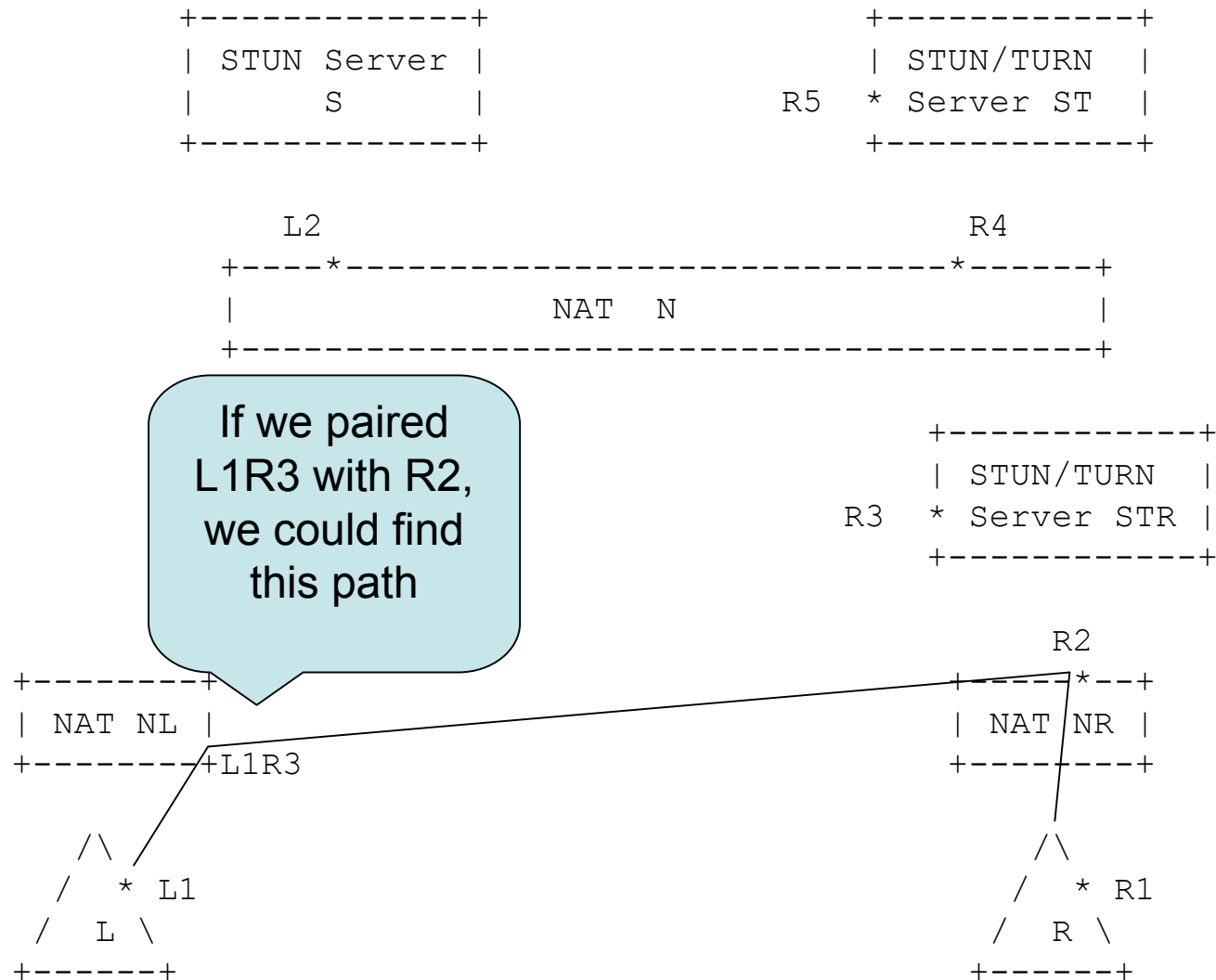
Open Issue #2: Pairing Peer Derived with other Candidates



Check causes new p-d candidate on NL to be discovered, paired only with R3



Open Issue #2: Pairing Peer Derived with other Candidates



Discussion

- Pairing new candidate with all other candidates would
 - Increase complexity of ICE further
 - Increase number of checks that are done
- Note that ICE still works – it just chooses suboptimal path
- Alternative proposal:
 - If a client cares about this, do a re-invite and add p-d candidate as a regular candidate
 - Requires no extra support from peer
- Ties in with the final open issue

Open Issue #3: Prioritization

- Current algorithm sorts pairs:
 - First in order of increasing MIN SN
 - Then in order of increasing MAX SN
- Consider
 - SN 1 is local
 - SN 2 is reflexive
 - SN 3 is relayed
- Consequence of this is that a pair {1,3} is preferred over {2,2}
 - Uses relay over two peer reflexive!

Open Issue #3: Alternatives

- Approach 1: Instead use this:
 - First order by increasing MAX SN
 - Then order by increasing MIN SN
 - This would prefer {2,2} over {1,3}
- Approach 2: Multiply q-values
 - Argument is its simpler
 - BUT, it doesn't work since q-values are cardinal
- Approach 3: Multiple sequence numbers
 - Seems promising, not clear if it would prefer relay over non-relayed
- Propose approach 1

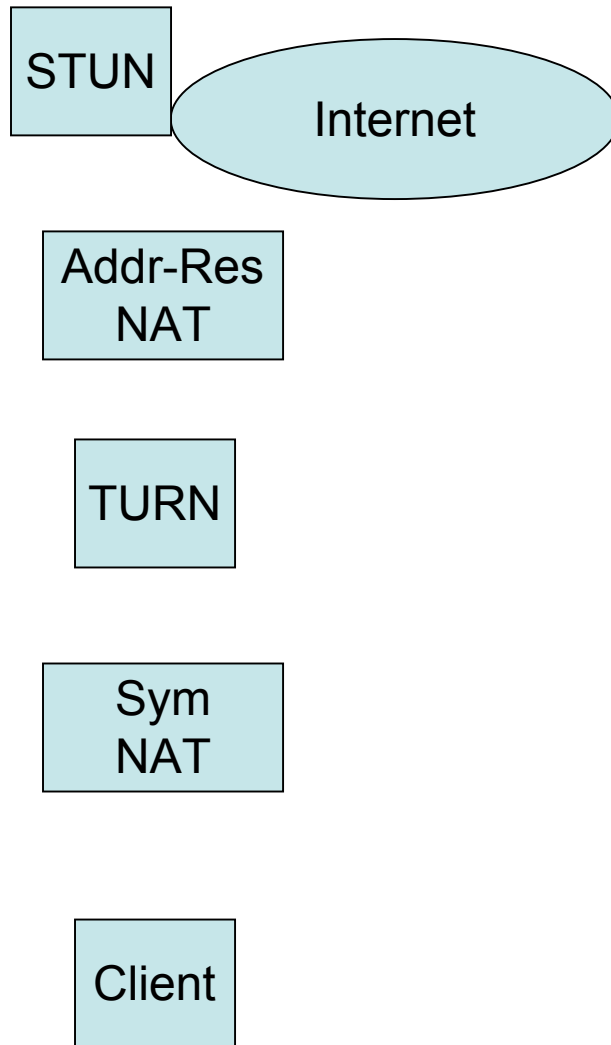
Open Issue #4

- Document talks about answer in places where the answer is the one in an unreliable 18x
- Rohan argues this is an early answer or courtesy answer, not real answer, change terminology
- I Propose to keep as is
 - Draft is *about* offer/answer usage!

Open Issue #5: Delay Checks

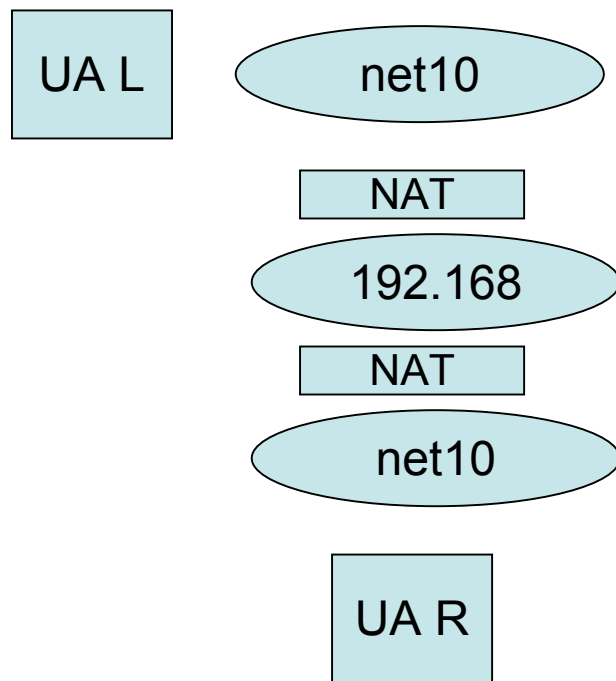
- ICE-09 (was also in several prior versions) handles the following use case
 - A offers to B
 - B sends answer, sends first check
 - Check beats answer back to A
 - A responds to STUN request, and when answer comes, does rest of ICE processing
- Somewhat of a hack, admittedly
- Proposal on list to, instead, have B delay sending check for 100ms
- I don't support this
 - Race may still happen with any timer value – just a question of probabilities
 - Delay would increase call setup times
 - Current mechanism not broken

Open Issue #6: STUN from TURN



- Address learned from client directly from STUN server not useful
- If client connects to TURN server, and from TURN server sends STUN query to STUN server, that produces useful address
- Do we want to support this?
- Proposal: No

Open Issue #7: Twice Ugly



- Checks from R to L create peer reflexive candidates
- What if outer NAT happens to allocate IP/port which match local address that UA R sent from?
- ICE fails in this case!
- Proposal:
 - Don't do any work to support, but hold that thought

Open Issue #8: Too Complicated

- I received several pieces of feedback that ICE has just gotten too complicated
- Addition of overview helps but mechanism is many pages of hard to understand text
- Concerns that it will hamper interop and limit deployment
- I have an idea on a simplification, I believe will reduce 20-30% of the normative text in the document without loss of function
- Looking for rough interest for me to try it out

Sources of Complexity

- Where is ICE complexity?
 - Lots of terminology
 - Candidate pairs and transport address pairs
 - Peer derived candidates
 - Complex logic in state machine for matching and missing

Big Idea

- Completely eliminate idea of peer derived candidates and all associated terminology
- Instead, when a connectivity check is received, and source IP differs, modify the remote address of the existing candidate pair
- Avoids the need to create new candidate, passwords, priorities, etc.
- Avoids the need to do miss/match processing
- No loss of functionality since this is what effectively happens
- Approximate elimination of 12 pages of 67 pages of normative text
- HOWEVER, would make it impossible to pair a peer derived address with other addresses, per issue #2
 - Would have to use reinvite

Other ideas

- Eliminate candidate pair wrapper
 - Rename existing transport address pair to candidate pair
 - Benefits
 - Should simplify presentation
 - Candidate line actually contains a candidate
 - USERNAME is just left:right not left:1:right:1
 - Can separately prioritize components
 - As a side effect fixes issue #7
- Unify algorithm with ICE-TCP

Drawbacks

- Will take some time to get a new draft out
- Will not be backwards compatible with ice-09
- So, question for group is: is there agreement that complexity reduction is sufficiently important to merit delay?