

NICER - Nicer ICE based on RTT

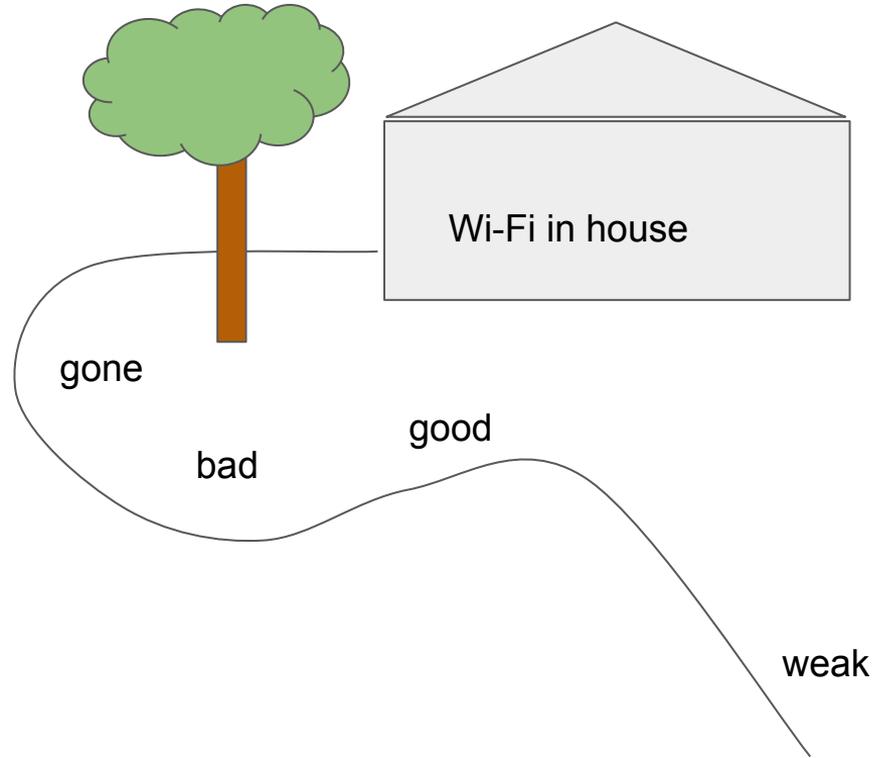
Jonas Orelund, Harald Alvestrand
Google

Problem Statement

ICE tries many paths.

It then selects one, and keeps it until ICE restart.

This is not always optimal.



Basic Idea of NICER

- Rather than throwing away non-best paths, keep them (or some of them)
- Keep probes running on them to give an idea of current quality
- When one path deteriorates, switch to another one
- If the bad path turns good again, switch back

Meshes well with Trickle-ICE: Making new candidates (and candidate pair sets) available whenever a new interface turns up.

Offers many opportunities for heuristic tuning and optimization

Requires a few changes in order to be interoperable

NICER: Required extensions to the ICE specification

- Permit the ICE controller to change the selected candidate pair
 - draft-thatcher-ice-renomination was one attempt at specifying this
- Be explicit that ICE candidate pairs with connectivity are not discarded
 - Those that never connect, or that are suspected of being duplicates, can be discarded
- Support Trickle ICE (RFC 8838)

We believe that with these extensions, any NICER ICE controller endpoint should be interoperable with any NICER ICE controlled endpoint.

NICER: Possible optimizations in the ICE protocol

There are a number of changes one could do in order to make NICER perform better.

NICER sends a large number of pings, so the size of the ping is important.

- Shorter checksums
- Omitting parameters whose value is unchanging after the first ping

NICER could also make good use of more information about the network on the ICE controlled side - however, this raises some privacy concerns.

NICER: ICE Controller behavior examples

NICER has to make two important decisions:

- When to ping
- When to switch

The ping decision may take into account phase of the call (ping a lot at first, more rarely later), stability of ping results (varying RTT may indicate a rapidly fluctuating network condition, requiring more pings for reliability), and other factors.

The switch decision may take into account RTT, ping failure rate, cost of interface (prefer free over \$-per-gigabyte), and other factors

NICER: Tricks and lessons

NICER will send many pings. One thing we learned is that on mobile devices, activating a radio costs a lot of power budget. Thus - rather than running all pairs on individual timers, bunch up all the pings that use the same radio, so that they are sent at the same time, activating only once.

NICER will switch earlier than “switch-after-break” schemes based on ICE restart, but switching has a cost too - a certain hysteresis needs to be applied so that the connection does not oscillate.

NICER will not probe with a full load, so when the channel switches, congestion may occur. Bandwidth management needs to integrate with NICER.

NICER: Open issues

NICER works best when it has maximum information available. But it has to run in the ICE Controller, since that's the one that initiates switches.

This works well for client to datacenter, since all the network complexity will be on the client side; it takes the controller role.

It works less well with peer-to-peer, unless the peers can exchange more information than is presently done.

NICER: Next steps

We seek the DISPATCH group's advice on one of the following:

- This is a bad idea because of <reason>. Do not speak of it again.
- This is something best done in private, but documented in public. Ask for an ind-sub Informational to tell others what you are doing.
- This is a small good idea. We should ask for an AD-sponsored PS.
- This is a big good idea. We should dispatch it to <Insert WG here>.
- This is a big good idea. We should reactivate or create a WG for it.

We do not have a preference at this time.