Workgroup: dispatch
Internet-Draft:
draft-oreland-dispatch-ice-nicer-00
Published: 6 July 2021
Intended Status: Informational
Expires: 7 January 2022
Authors: J. Oreland H. Alvestrand
Google Google

NICER - a better usage profile on ICE

Abstract

NICER presents an usage profile of ICE that permits more dynamic adaptation to network conditions over the time of a call.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the mailing list (dispatch@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dispatch/.

Source for this draft and an issue tracker can be found at https://github.com/alvestrand/nicer-spec.

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 https://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 7 January 2022.

Table of Contents

1. Introduction
2. Conventions and Definitions
3. Problem statement
4. Traditional ICE
5. NICER - the idea
6. Standardization requirements
7. Possible optimizations
8. Implementation issues
9. Security Considerations
10. IANA Considerations
11. Normative References
Acknowledgments
Authors' Addresses

1. Introduction

TODO Introduction

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Problem statement

Consider the case of someone walking home while on an Internet-based audio call. 4G coverage in his area is good, but for cost reasons, he prefers to use wifi when available.

As he nears his house, the phone picks up a weak wifi signal - high packet loss, low bandwidth, but definitely present. Next to his front door, there is a big tree. As he walks behind the tree, the signal disappears. As he walks in the front door, the wifi signal is strong and stable.

What media should the call use while it is progressing through these scenarios?

4. Traditional ICE

The traditional ([RFC8863]) version of ICE can be briefly described as:

- *Generate a large number of candidate pairs
- *Try them in order until one of them works
- *Start using the working one (ICE controller decides)
- *Throw away all the other ones
- *When the chosen pair breaks, do an ICE restart and start over

There are extensions specified to this model; one of particular interest is Trickle ICE (RFC 8838), which allows adding more candidates after initialization, forming new sets of candidate pairs without an ICE restart.

5. NICER - the idea

The idea behind NICER is that rather than keeping a single candidate pair up, the ICE controller will form a list of candidate pairs it considers "potentially viable". The ICE controller will perform STUN Pings (bind requests) on these pairs to keep them alive and get some metrics on quality (RTT, packet loss).

When the ICE controller decides that one of these pairs is doing better than the currently active candidate pair, it will switch the active pair to this pair, and relegate the old pair to the "alternate" pool, informing the other party through a BIND request with the "nominated" flag set.

The ICE controller will still discard candidate pairs that never started working, and candidate pairs that have a high likelihood of being duplicates of other candidate pairs in the pool.

When new network interfaces come up, the ICE controller will use trickle-ICE to communicate with the other party and form new sets of candidate pairs; when interfaces go down, the ICE controller will switch to a still-working interface at once; ICE restart will only happen once all previously usable connections have failed.

It follows from the description above that NICER may need to add candidates at any time; the simplest approach compatible with standard ICE is to never send end-of-candidates, but more subtle approaches should be possible.

6. Standardization requirements

Most of the adaptations needed for NICER are within the ICE controller, and don't need to be standardized. However, there are a few parts that affect messages on the wire, and these call for some standardization effort - either by pushing through existing proposals that cover the need, or by standardizing new features.

These are:

*Trickle ICE [RFC8838]

*Continuous renomination

Continuous renomination means that for some subset of candidate pairs not selected, rather than discarding them as mandated by [RFC8863] section 8.3, they will be retained and be made available for selection by sending a check with the USE-CANDIDATE attribute on that candidate pair. One writeup for an extension that would permit this was draft-thatcher-ice-renomination (expired I-D).

With these features in place, NICER should be deployable against any system that impelments these features.

7. Possible optimizations

In Google's experimentation with NICER, we have experiemented with reducing the size of the ping - omitting known parameters and using shorter message-integrity functions. These optimizations may be interesting to standardize, but not essential for making NICER work.

8. Implementation issues

These are things for which standardization is not needed, but where implementors who want to use NICER need to be aware of them.

The definition of "better" is quite fluid and complex. The data available from a connection that is only occasionally pinged is also limited; for instance, bandwidth limitations can't be probed with just occasional probe packets (although guesses can be made). In particular, the ICE concept of "priority" (used to rank candidate pairs consistently at the ICE controller and controlled entities) is not useful for ranking the preferability of candidates for switching.

Managing power budgets on mobile devices can be challenging. In particular, pinging interfaces keeps radios alive and therefore consume power; when an interface has no reachable connections, one should avoid pinging it.

Given the dynamic nature of RF environments, occasional pinging runs the risk of decisions being taken on stale data, while frequent pings

use battery and bandwidth; tuning these tradeoffs requires some attention while implementing.

9. Security Considerations

Keeping additional paths open increases the attack area for MITM attacks, naturally. So just as in the case of other TURN usages, it is important that the traffic sent over these TURN connections be authenticated and encrypted as appropriate.

Shortening the checksum will weaken the barrier to impersonation. This may not matter if the shortened checksum is only used on subsequent pings, not initial handshakes.

10. IANA Considerations

This document has no IANA actions.

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8838] Ivov, E., Uberti, J., and P. Saint-Andre, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", RFC 8838, DOI 10.17487/RFC8838, January 2021, <https://www.rfc-editor.org/rfc/rfc8838>.

[RFC8863] Holmberg, C. and J. Uberti, "Interactive Connectivity Establishment Patiently Awaiting Connectivity (ICE PAC)", RFC 8863, DOI 10.17487/RFC8863, January 2021, <https://www.rfc-editor.org/rfc/rfc8863>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Jonas Oreland
Google

Email: jonaso@google.com

Harald Alvestrand

Google

Email: hta@google.com