QUIC M. Duke
Internet-Draft F5 Networks, Inc.
Intended status: Informational 29 July 2020
Expires: 30 January 2021

Network Address Translation Support for QUIC
draft-duke-quic-natsupp-03

Abstract

Network Address Translators (NATs) are widely deployed to share scarce public IPv4 addresses among multiple end hosts. They overwrite IP addresses and ports in IP packets to do so. QUIC is a protocol on top of UDP that provides transport-like services. QUIC is better-behaved in the presence of NATs than older protocols, and existing UDP NATs should operate without incident if unmodified. QUIC offers additional features that may tempt NAT implementers as potential optimizations. However, in practice, leveraging these features will lead to new connection failure modes and security vulnerabilities.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Network Address Translators (NATs) are a widely deployed means of multiplexing multiple private IP addresses over scarce IPv4 public address space by replacing those addresses and using ports to distinguish those connections. The new address can also guarantee that packets move through a proxy throughout the life of a connection, so that the connection can continue with the required state at that proxy.

This document uses the colloquial term NAT to mean NAPT (section 2.2 of [RFC3022]), which overloads several IP addresses to one IP address or to an IP address pool, as commonly deployed in carrier-grade NATs or residential NATs.

QUIC [QUIC-TRANSPORT] is a protocol, operating over UDP, that provides many transport-like services to the application layer. Among these services is the mapping of multiple endpoint IP addresses to a single connection through use of a Connection ID (CID). Connection IDs are opaque byte fields that are expressed consistently across all QUIC versions [QUIC-INVARIANTS]. This feature may appear to present opportunities to optimize NAT port usage and simplify the work of the QUIC server. In fact, NAT behavior that relies on CID may instead cause connection failure when endpoints change Connection ID, and disable important protocol security features.

The remainder of this document explains how QUIC supports NATs better than other connection-oriented protocols, why NAT use of Connection ID might appear attractive, and how NAT use of CID can create serious problems for the endpoints. The conclusion of this document is that NATs should retain their existing 4-tuple-based operation and refrain from parsing or otherwise using QUIC connection IDs.

[RFC4787] contains some guidance on building NATs to interact constructively with a wide range of applications. This document extends the discussion to QUIC.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. QUIC and NAT Rebinding

An explicit goal of QUIC is to be robust to NAT rebinding. When a connection is idle for a long time, the NAT may guess it has terminated and assign the client port to a new connection. As TCP defines a connection by its address and port 4-tuple, a TCP packet will not appear to belong to any existing connection at the receiver.

QUIC endpoints identify their connections using a CID that is encoded in every packet. If the client attempts to resume communication, the packet will be assigned a new source IP and/or port. Incoming packets from the server will be misrouted and dropped until the client sends a packet from its new address.

Therefore, QUIC connections can survive NAT rebindings as long as no routing function in the path is dependent on client IP address and port to deliver packets between server and NAT. Reducing the timeout on UDP NATs might be tempting in light of this property, but not all QUIC server deployments will be robust to rebinding.

4. The Lure of the Connection ID

There are a few reasons that CID-aware NATs could seemingly appear attractive.

4.1. Resource Conservation

NATs sometimes hit an operational limit where they exhaust available public IP addresses and ports, and must evict flows from their address/port mapping. CIDs offer a way to multiplex many connections over a single address and port.

However, QUIC endpoints may negotiate new connection IDs inside cryptographically protected packets, and begin using them at will. Imagine two clients behind a NAT that are sharing the same public IP address and port. The NAT is differentiating them using the incoming Connection ID. If one client secretly changes its connection ID, there will be no mapping for the NAT, and the connection will suddenly break.

While mid-connection failure in some cases may seem superior to rejecting QUIC outright, HTTP/3 over QUIC falls back to TCP. This is preferable to a connection suddenly black holing and timing out. Furthermore, wide deployment of NATs with this behavior would make it risky to change Connection IDs in the internet, which would thwart various important protocol properties.

It is possible, in principle, to encode the client’s identity in a connection ID using [QUIC-LB] and explicit coordination with the NAT. However, QUIC-LB makes assumptions about endpoint mobility and common configuration in server infrastructure that are almost never valid in client/NAT architectures. Deploying such a system would include the administrative overhead while not solving the problem described in this section if the client changes networks.

Note that using connection IDs in this manner would anyway violate the best common practice to avoid "port overloading" as described in [RFC4787].

4.2. "Helping" with routing infrastructure issues

One problem in QUIC deployment is router and switch server infrastructures that direct traffic based on address-port 4-tuple rather than connection ID. The use of source IP address means that a NAT rebinding or address migration will deliver packets to the wrong server. For the reasons described above, routers and switches will not have access to negotiated CIDs. This is a particular problem for low-state load balancers, and a QUIC extension exists [QUIC-LB] to allow some server-load balancer coordination for routable CIDs.

A NAT at the front of this infrastructure might save the effort of converting all these devices by decoding routable connection IDs and rewriting the packet IP addresses to allow consistent routing by legacy devices.

Unfortunately, the change of IP address or port is an important signal to QUIC endpoints. It requires a review of path-dependent variables like congestion control parameters. It can also signify various attacks that mislead one endpoint about the best peer address for the connection (see section 9 of [QUIC-TRANSPORT]). The QUIC PATH_CHALLENGE and PATH_RESPONSE frames are intended to detect and mitigate these attacks and verify connectivity to the new address. This mechanism cannot work if the NAT is bleaching peer address changes.

For example, an attacker might copy a legitimate QUIC packet and change the source address to match its own. In the absence of a bleaching NAT, the receiving endpoint would interpret this as a potential NAT rebinding and use a PATH_CHALLENGE frame to prove that the peer endpoint is not truly at the new address, thus thwarting the attack. A bleaching NAT has no means of sending an encrypted PATH_CHALLENGE frame, so it might start redirecting all QUIC traffic to the attacker address and thus allow an observer to break the connection.

5. Filtering behavior

[RFC4787] describes possible packet filtering behaviors that relate to NATs. Though thes guidance there holds, a particularly unwise behavior is to admit a handful of UDP packets and then make a decision as to whether or not to filter it. QUIC applications are encouraged to fail over to TCP if early packets do not arrive at their destination. Admitting a few packets allows the QUIC endpoint to determine that the path accepts QUIC. Sudden drops afterwards will result in slow and costly timeouts before abandoning the connection.

6. QUIC Detection

Beyond the above difficulties, merely identifying that a UDP packet is part of a QUIC connection is not straightforward. Due to address migration, NATs cannot assume that QUIC version 1 application traffic is preceeded by a handshake on the path. The short header prepended to version 1 application traffic has few consistent codepoints that reliably identify it as QUIC. Moreover, the protocol is designed to be extensible. [QUIC-INVARIANTS] describes the small set of QUIC protocol properties that will remain stable across versions.

For these reasons, applying generalized UDP policies will prevent accidental breakage of QUIC features and mishandled non-QUIC UDP packets.

7. Security Considerations

This document proposes no change in behavior in the internet, so there are no new security implications. However, ignoring the recommendations here could prevent existing security mechanisms in QUIC from working properly.

8. IANA Considerations

There are no IANA requirements.

9. Informative References

[QUIC-INVARIANTS]
Thomson, M., "Version-Independent Properties of QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-invariants-latest, <https://tools.ietf.org/html/draft-ietf-quic-invariants-latest>.

[QUIC-LB] Duke, M. and N. Banks, "QUIC-LB: Generating Routable QUIC Connection IDs", Work in Progress, Internet-Draft, draft-duke-quic-load-balancers-latest, <https://tools.ietf.org/html/draft-duke-quic-load-balancers-latest>.

[QUIC-TRANSPORT]
Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-latest, <https://tools.ietf.org/html/draft-ietf-quic-transport-latest>.

[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/info/rfc2119>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <https://www.rfc-editor.org/info/rfc3022>.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <https://www.rfc-editor.org/info/rfc4787>.

Appendix A. Acknowledgments

Thanks to Dmitri Tikhonov, who first recognized that certain NAT behaviors could create problems for QUIC.

Appendix B. Change Log

> *RFC Editor’s Note:* Please remove this section prior to$ publication of a final version of this document.$

B.1. since draft-duke-quic-natsupp-02

- Added discussion of QUIC identification

B.2. since draft-duke-quic-natsupp-01

- Added brief discussion of impact of filtering.

- Added references to RFC 4787.

- Corrected normative reference to be informative.

B.3. since draft-duke-quic-natsupp-00

- Tightened NAT terminology

- Added additional clarfying examples

- Added warning against using QUIC-LB for NATs that front clients.

Author’s Address

Martin Duke
F5 Networks, Inc.

Email: martin.h.duke@gmail.com