QUIC-LB: Using Load Balancers to Generate QUIC Connection IDs
draft-duke-quic-load-balancers-00.txt

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on August 12, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in
Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

QUIC connection IDs allow continuation of connections across address/port 5-tuple changes, and can store routing information for stateless or low-state layer 4 load balancers. They are also meant to prevent linkability of connections across deliberate address migration through the use of protected communications between client and server. This creates issues for load-balancing intermediaries. This specification standardizes the communication between load balancers and servers to overcome these issues in a protocol called QUIC-LB.

Table of Contents

1. Introduction...................................................3
   1.1. Terminology...............................................3
2. Protocol Objectives............................................4
   2.1. Simplicity................................................4
   2.2. Security..................................................4
   2.3. Robustness to Middleboxes..............................4
3. Protocol Design................................................4
   3.1. Connection ID Generation................................4
   3.2. Message Exchange........................................5
   3.3. Load Balancer Trust.....................................6
   3.4. Servers with Zero Stock................................6
4. Message Format................................................6
   4.1. NEW_IDS message........................................6
   4.2. ID_STOCK message.......................................7
5. Chained Load Balancers........................................8
6. Security Considerations........................................8
7. IANA Considerations...........................................8
8. References.....................................................9
   8.1. Normative References.....................................9
   8.2. Informative References....................................9
9. Acknowledgments.................................................9
Appendix A. Alternate Approaches to server/load balancer coordination problems........................................10
   A.1. Terminating proxies......................................10
   A.2. Unified control of load balancer and server.............10
   A.3. Synchronized Algorithms..................................10
1. Introduction

Server-generated connection IDs create a potential need for out-of-band communication. QUIC packets usually contain a connection ID to allow endpoints to associate packets with different address/port/protocol 5-tuples to the same connection context. This feature makes connections robust in the event of NAT rebinding.

QUIC allows servers (or load balancers) to designate an initial connection ID to encode useful routing information for load balancers. It also encourages servers, in packets protected by cryptography, to provide additional connection IDs to the client. This allows clients that know they are going to change IP address or port to use a separate connection ID on the new path, thus reducing linkability as clients move through the world.

There is a tension between the requirements to provide routing information and mitigate linkability. Ultimately, because new connection IDs are in protected packets, they must be generated at the server if the load balancer does not have access to the connection keys. However, it is the load balancer that has the context necessary to generate a connection ID that encodes useful routing information. In the absence of any shared state between load balancer and server, the load balancer must maintain a relatively expensive table of server-generated connection IDs, and will not route packets correctly if they use a connection ID that was originally communicated in a protected NEW_CONNECTION_ID frame.

This specification provides a method of coordination between QUIC servers and layer 4 load balancers to support connection IDs that encode routing information. It describes desirable properties of a solution, and then specifies a protocol that provides those properties.

1.1. Terminology

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.

In this document, "client" and "server" refer to the endpoints of a QUIC connection unless otherwise indicated. A "load balancer" is an intermediary for that connection that does not possess QUIC.
connection keys, but it may rewrite IP addresses or conduct other IP or UDP processing. In most respects, the load balancer behaves as a client in a QUIC-LB connection, but is always referred to as a "load balancer" below to avoid confusion.

2. Protocol Objectives

2.1. Simplicity

QUIC is intended to provide unlinkability across connection migration, but servers are under no obligation to provide connection IDs to enable this. If the coordination scheme is too difficult to implement, servers behind load balancers using connection IDs for routing will use trivially linkable connection IDs. Clients will therefore be forced to choose between terminating the connection during migration or remaining linkable.

The solution should be both simple to implement and require little additional infrastructure for cryptographic keys, etc.

2.2. Security

The path between load balancer and server may not be free of observers from which the client wishes to avoid linkability. Similarly, malicious hosts could spoof a trusted load balancer to provide connection IDs that are linkable. Therefore, coordination messages must be encrypted, and there must be some way for servers to authenticate the load balancer’s messages.

2.3. Robustness to Middleboxes

The path between load balancer and server may transit multiple domains. It is therefore advantageous to make messages resemble QUIC traffic as much as possible, as any viable path must obviously admit QUIC traffic.

3. Protocol Design

3.1. Connection ID Generation

Load balancers MAY use connection IDs to encode routing information to the destination server. This encoding MAY be opaque to the destination server and SHOULD be opaque to all other hosts.

The encoding scheme MUST be able to generate enough connection IDs for each server to have at least two for every QUIC connection concurrently assigned to it.
The encoding SHOULD maximize the cryptographic distance between connection IDs intended for the same server.

The encoding SHOULD NOT vary with the number of active servers, as the connection ID remains routable even if other servers boot up or suffer an outage.

A representative encoding that meets these requirements might concatenate the server’s IPv4 address and a monotonically increasing sequence number, and then encrypt the result to obtain the connection ID. For any incoming QUIC packet, the load balancer would decrypt the connection ID to extract the server IP address. There would be different routing rules for (readily identifiable) Initial packets that contain an (essentially random) client-generated connection ID.

3.2. Message Exchange

No message in this protocol is sent with reliability assurances. For all messages the load balancer uses an ephemeral UDP port, and the server uses UDP port 443. All messages are sent as encrypted records in an established DTLS connection.

The best practice for servers is to always provide at least one connection ID to clients beyond the one it is currently using. Load balancers SHOULD monitor the usage of these connection IDs and the number of active connections for each server. A "used" connection ID is one that has been used in the Connection ID field of a QUIC header, as opposed to the QUIC NEW_CONNECTION_ID frame. When the stock of unused connection IDs is low, load balancers SHOULD send a NEW_IDS message to that server.

Servers SHOULD periodically send an ID_STOCK message to the load balancer to synchronize the load balancer’s view of its current unused connection IDs. This allows the shared state to recover from lost NEW_CONN_ID messages.

Servers MAY increase the rate at which they send ID_STOCK messages as their stocks shrink, relative to the usage rate of connection IDs, to accelerate delivery of new IDs and overcome packet losses.

Note that the Connection IDs provided by the load balancer can be used by any connection terminated at the server. There is no need for the load balancer to designate specific QUIC connections for each ID. As a result, load balancers cannot necessarily associate packets before and after an IP address migration to the same connection.
3.3. Load Balancer Trust

Message authentication and encryption is achieved using DTLS 1.2 or 1.3 ([RFC6347] or [I-D.ietf-tls-dtls13]). Load balancers MUST initiate the handshake as the client, as some firewalls may block outbound connections from the server. Servers MUST request a Client Certificate to verify that the Load Balancer meets the trust requirements to potentially introduce linkable Connection IDs into the system.

Servers MUST NOT accept DTLS connections from load balancers for which they do not have configured trust relationships.

3.4. Servers with Zero Stock

If the server has an active DTLS connection with a lower balancer, but has zero stock, the server SHOULD use the connection ID provided in the Initial packet and SHOULD NOT generate QUIC NEW_CONNECTION_ID frames. Therefore, clients that knowingly change IP address or port are forced to choose between terminating the connection and traceably changing IP address.

Servers with no such trust relationship MUST behave in accordance with the QUIC transport spec [ID-ietf-quic-transport], generating new connection IDs at will.

4. Message Format

All messages below are encapsulated in DTLS Records.

The type field is not strictly necessary to resolve ambiguity, as each message type is only sent by one entity in the connection. However, the type byte allows future extension of the protocol.

4.1. NEW_IDS message

Load Balancers MUST ignore NEW_IDS messages.
4.2. ID_STOCK message

Servers MUST ignore received ID_STOCK messages.

This message simply reports the number of unused connection IDs in a 32-bit integer in Network order. Load Balancers MUST update their estimate of server stock based on this message, as some connection IDs may have been lost in transit.
5. Chained Load Balancers

In some deployments, there may be multiple tiers of trusted load balancers in the path between client and server. All load balancers using connection ID to encode routing information MUST agree on how to decode connection IDs as routing instructions. Due to QUIC packet authentication, connection IDs of established QUIC connections cannot be rewritten in flight without access to the QUIC connection keys.

A server configured to trust multiple load balancers MAY accept DTLS connections from all of them and use provided Connection IDs interchangeably. It SHOULD report its entire stock of connection IDs to all trusted load balancers, rather than the number of IDs issued from each source.

6. Security Considerations

QUIC-LB is intended to preserve routability and prevent linkability, so attacks on the protocol would compromise at least one of these objectives.

Injection of connection IDs could either break routability (by diverting flows to a server with no QUIC connection context) or allow linkability (by allowing observers to determine that two connection IDs originate from the same server, and that one begins at roughly the same time that the other disappears). Use of DTLS authentication mechanisms, at both load balancer and server, are meant to mitigate this risk.

Cleartext connection IDs would also allow observers to map connection IDs to a specific server, potentially allowing linkability. QUIC-LB utilizes DTLS-based encryption to avoid this.

QUIC-LB depends on DTLS, and therefore on Public Key Infrastructure. Any compromise of the PKI would allow untrusted middleboxes to successfully execute either of the attacks above.

7. IANA Considerations

There are no IANA requirements.
8. References

8.1. Normative References


8.2. Informative References

None.

9. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.
Appendix A. Alternate Approaches to server/load balancer coordination problems.

A.1. Terminating proxies

A load balancer that terminates the QUIC connection at the application layer, with separate connections to client and server, can generate new connection IDs on its own and has enough state to not require connection ID to route packets. However, this approach cannot work if the load balancer is not a trusted intermediary with the TLS certificates required to terminate a QUIC connection. Moreover, the load balancer may not have sufficient resources to keep the state for every connection that traverses it.

A.2. Unified control of load balancer and server

Obviously, a vendor or network operator with access to the source code of both the load balancer and the server can develop a proprietary method to share the necessary state. This is difficult to implement for network operators that purchase their servers and load balancers from different vendors.

A.3. Synchronized Algorithms

If server understands how the load balancer encodes routing information via a standard algorithm, it can generate new connection IDs that preserve routability. However, if the algorithm is not cryptographically secure this erodes protection against linkability. Ultimately, this approach requires a key rotation scheme to remain secure.
Authors’ Addresses

Martin Duke
F5 Networks
401 Elliott Ave W.
Seattle, WA, USA 98119

Email: m.duke@f5.com