Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 5, 2021

S. Chen
Y. Zhang
H. Wang
Z. Li
Huawei Technologies

June 3, 2021

BGP Over QUIC
draft-chen-idr-bgp-over-quic-00

Abstract

Border Gateway Protocol (BGP) is an autonomous system routing protocol. The main function of BGP is to exchange routing and reachability information between autonomous systems(AS) on the Internet. BGP uses TCP to implement reliable and orderly transmission of information. Similar to TCP, QUIC is a UDP-based, byte-stream-based reliable data transmission service. In addition, by integrating with TLS 1.3, QUIC also supports functions such as establishing connections with minimum latency and providing confidentiality and integrity protection for the transmitted data, and multi-stream multiplexing. Taking use of QUIC for BGP can achieve the possible advantages. This document defines the mechanism of BGP over QUIC to and corresponding procedures.

Requirements Language

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 [RFC2119].

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 December 5, 2021.

Table of Contents

1. Introduction

BGP is used to distribute IP routes between autonomous systems and it is one of the most important Internet protocols. BGP Multiprotocol Extensions (MP-BGP) [RFC4760] enables BGP to distribute routes of various address families, such as VPN-IPv4 routes[RFC4364], VPN-IPv6 routes [RFC4659], and EVPN routes [RFC7432].

BGP is a routing protocol that requires long-term session persistence. BGP requires that transport protocol provide reliable and secure data transmission services. In [RFC4271] , TCP is defined as the transport protocol for BGP. However, TCP does not protect the confidentiality of the transmitted data.

Currently, BGP uses MD5, TCP-AO and TCP Over TLS to provide integrity protection. However, MD5 has been considered an insecure encryption algorithm ([RFC6151]). TCP-AO ([RFC5925][RFC5926]) is unable to encrypt the payload. TLS ([RFC5246][RFC8446]) can be added between BGP and TCP to provide identity authentication, confidentiality, and integrity protection for BGP. However, the way is inefficient since when establishing a BGP session, a three-way handshake is adopted to establish a TCP connection, and then TLS handshake authentication is also performed.

QUIC [RFC9000] [RFC9001] is a UDP-based transport protocol that provides the following functions:

1. Reliable data transmission service based on byte streams similar to TCP.

2. Support low-latency connection establishment.

3. Authentication of the server is provided during connection establishment.

4. Authentication of the client is provided during connection establishment. (Optional)

5. QUIC provides confidentiality and integrity protection for transport data and key fields in QUIC headers. QUIC also supports periodic key updates.

6. Supports stream multiplexing, including unidirectional and bidirectional streams.

7. Supports connection migration

Comparing with BGP over TCP, BGP over Quic (BoQ) provides the following benefits:

1. The BGP session establishment delay can be reduced since the handshake times can be reduced comparing with BGP over TCP/TLS.

2. The head-of-line block between BGP address families can be eliminated by adopting stream-multiplexing for BGP.

3. Endogenous transport-layer security is provided, and no additional TLS is required.

This document defines the mechanism of BGP over QUIC to and corresponding procedures.

2. Terminology

Client: QUIC client, the active part of QUIC connection.

Server: QUIC server, the passive part of QUIC connection.

BGP over TCP: BGP using TCP as the transport layer, as [RFC4271].

BoQ: BGP over QUIC, i.e., BGP using QUIC as the transport layer.

ALPN: Application-Layer Protocol Negotiation

3. Design Consideration

3.1. BGP Specification Compatibility

BoQ replaces only the transport layer of BGP over TCP, requiring that the BGP protocol specification remain backward compatible.

Note that during the establishment of a BGP session, the BGP session state machine needs to receive transport-layer event. The BoQ also needs to receive and process QUIC-related events.

3.2. Design for Minimum Latency

QUIC provides minimal connection setup delay. The BGP session setup delay is shortened from TLS 1.3(1 RTT) + TCP(3 RTT) to QUIC(1 RTT). If a BGP session is not established for the first time, the RTT can be set to 0 to shorten the BGP session setup delay.

As the core routing protocol of the Internet, a large-scale BGP session needs to be established over a long distance. Reducing the BGP session setup delay helps improve the overall network performance.

3.3. Eliminate Head-of-line Block

Data transmitted in a BGP session can be classified into multiple objects: address family, VRF, and route prefix. Different objects can be mapped to different QUIC streams as required to isolate these objects, thereby eliminating the head-of-line blocking problem.

4. Specification

4.1. BoQ Protocol Stack

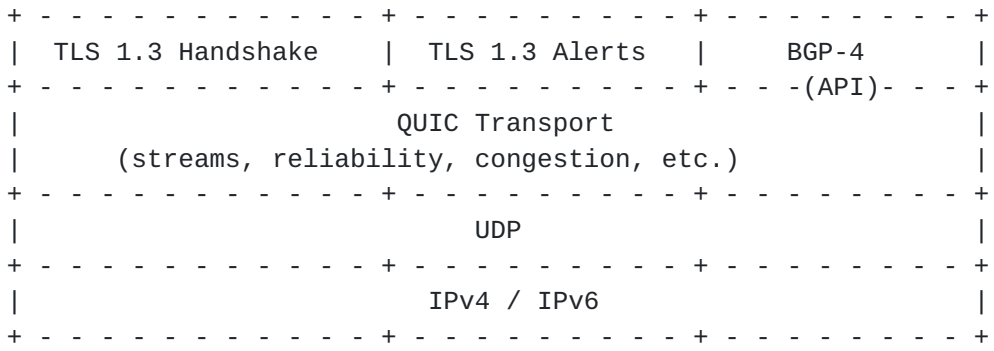

Figure 1. QUIC Protocol Stack

QUIC provides reliable data transmission services based on byte streams. In addition, QUIC uses TLS 1.3 [RFC8446] to protect confidentiality and integrity of transmitted data.

4.2. Port number and ALPN

According to Figure 1, QUIC is an application protocol that runs on top of UDP. However, QUIC is designed as a generic (application-layer) transport protocol and does not define a standalone UDP port [I-D.ietf-quic-applicability]. The QUIC uses the port of the application protocol to identify a specific application protocol. In addition, in some cases, multiple protocol versions can run simultaneously on a port number. For example, multiple protocol versions of HTTPS use TCP/UDP port 443, such as HTTP/2 [RFC7540] and HTTP/3 [I-D.ietf-quic-http].

BGP uses TCP/UDP port 179, and currently only a unique valid protocol version BGP-4 exists. Therefore, the QUIC may use only UDP port 179 to uniquely identify the BoQ (that is, BGP-4 over QUIC). The ALPN does not need to be specified.

4.3. Stream mapping

In a QUIC connection, up to 2 ^ 62 - 1 QUIC streams can be created and the QUIC stream can be unidirectional or bidirectional. In one connection, data of multiple streams may be transmitted at the same time. QUIC strictly ensures a transmission sequence of data of the same stream, but does not guarantee a transmission sequence of data of different streams. QUIC also supports stream-level flow control. This function is called stream multiplexing.

BGP can take use of the stream multiplexing to solve the head-of-line issues. For example, for MP-BGP [RFC4760], multiple address families can be deployed in a BGP session. Each address family can be configured with multiple VRFs, and each VRF can contain multiple route prefixes. To isolate objects at different layers and eliminate queue head blocking between these objects, the following QUIC stream mapping mode can be selected:

Option 1, Mapping streams based on address families: One or more address family can be mapped to one stream.

Option 2, Mapping streams based on VRFs: One or more VRFs can be mapped to one stream.

Option 3, Mapping streams based on prefix: it can be combinations of prefixes.

Note that regardless of which mapping mode is selected, data of the same object MUST be received and transmitted using the same QUIC stream.

4.4. BGP Session Establishment

4.4.1. BGP FSM

Before distributing routes, BGP needs to establish a point-to-point connection, called a BGP session. [RFC4271] defines BGP FSM to describe the BGP session establishment process.

A BGP session can be established in two phases:

1. Establish a transport layer connection. For BGP over TCP, a TCP connection is established through a three-way handshake. For BoQ, a 1-RTT handshake or 0-RTT handshake is used to establish a QUIC connection.

2. Establish a BGP session. After a transport-layer connection is established, BGP peers exchange BGP Open and BGP Keepalive messages. If the BGP peers reach the Established state, the BGP session has been established.

Similar to TCP, QUIC distinguishes the client (active party) from the server (passive party). Therefore, the connection conflict detection and resolution methods described in [RFC4271] are also applicable to BoQ FSM.

In this document, BGP FSM is referred to as BoQ FSM. In the 1-RTT handshake, the BoQ FSM inherits the BGP FSM defined in [RFC4271].

TCP-related session attributes should be replaced with BoQ FSM-specific session attributes and FSM events defined in this document. In addition, the processing of BoQ FSM in the 0-RTT handshake is added.

The session attributes and FSM events specific to BoQ FSM are defined as follows:

1.Session Attribute

(1) Optional Session Attributes: PassiveQuicEstablishment

Description: This option indicates that the BGP FSM will passively wait for the remote BGP peer to establish the BGP QUIC connection.

Value: TRUE or FALSE

(2) Optional Session Attributes: TrackQuicState

Description: The BGP FSM normally tracks the end result of a QUIC connection attempt rather than individual QUIC messages. Optionally, the BGP FSM can support additional interaction with the QUIC connection negotiation. .

Value: TRUE or FALSE

2. FSM Event

QUIC directly encapsulates the handshake process of TLS 1.3 [RFC8446]. In addition, QUIC requires that all packets must be explicitly acknowledged. Therefore, QUIC defines the end state of two connection establishment [RFC9001]

(1) Handshake Complete: TLS 1.3 has successfully completed the handshake.

(2) Handshake Confirmed: The QUIC has successfully completed the handshake.

On the client, the state is Handshake Complete and then Handshake Confirmed. On the server, the two states are reached at the same time.

The transport layer events for BoQ FSM are defined as follows :

Event 29: ManualStart_with_PassiveQuicEstablishment

Definition: Local system administrator manually starts the peer connection, but has PassiveQuicEstablishment enabled.

Status: Optional, depending on local system

Optional Attribute Status:

1) The PassiveTcpEstablishment attribute SHOULD be set to TRUE if this event occurs.

2) The DampPeerOscillations attribute SHOULD be set to FALSE when this event occurs.

Corresponding TCP events: Event 4

Event 30: AutomaticStart_with_PassiveQuicEstablishment

Definition: Local system automatically starts the BGP connection with the PassiveQuicEstablishment enabled.

Status: Optional, depending on local system

Optional Attribute Status:

1) The AllowAutomaticStart attribute SHOULD be set to TRUE.

2) The PassiveTcpEstablishment attribute SHOULD be set to TRUE.

3) If the DampPeerOscillations attribute is supported, the DampPeerOscillations SHOULD be set to FALSE.

Corresponding TCP events: Event 5

Event 31: AutomaticStart_with_DampPeerOscillations_and_PassiveQuicEstablishment

Definition: Local system automatically starts the BGP peer connection with peer oscillation damping enabled and PassiveQuicEstablishment enabled. The exact method of damping persistent peer oscillations is determined by the implementation and is outside the scope of this document.

Status: Optional, depending on local system

Optional Attribute Status:

1) The AllowAutomaticStart attribute SHOULD be set to TRUE.

2) The DampPeerOscillations attribute SHOULD be set to TRUE.

3) The PassiveTcpEstablishment attribute SHOULD be set to FALSE.

Corresponding TCP events: Event 7

Event 32: QuicConnection_Valid

Definition: This parameter is applicable only to the QUIC server. It indicates that the Handshake Confirmed state is reached.

Status: Optional

Optional Attribute Status: 1) The TrackTcpState attribute SHOULD be set to TRUE if this event occurs.

Corresponding TCP events: Event 14

Event 33: Quic_CR_Invalid

Definition: This parameter applies only to the QUIC server and indicates that an invalid QUIC connection request is received. Initial packets with invalid source addresses or port numbers,invalid destination addresses or port numbers or version negotiation or address validation fails.

Status: Optional

Optional Attribute Status: 1) The TrackTcpState attribute should be set to TRUE if this event occurs.

Corresponding TCP events: Event 15

Event 34: Quic_CR_Acked

Definition: This parameter applies only to the QUIC client. It indicates that an Initial ACK message is received from the QUIC server and an Initial/Handshake message is sent to the QUIC server. Note: When this event is received, the QUIC client has reached the Handshake Complete state.

Status: Mandatory

Corresponding TCP events: Event 16

Event 35: QuicConnectionConfirmed

Definition: This parameter applies to both QUIC client and QUIC server, indicating that the Handshake Confirmed state has been reached.

Status: Mandatory

Corresponding TCP events: Event 17

Event 36: QuicConnectionFails

Definition: This parameter applies to both the QUIC client and the QUIC server. It indicates that an error occurs in the QUIC handshake before the system enters the Handshake Confirmed state.

Status: Mandatory

Corresponding TCP events: Event 18

4.4.2. 1-RTT Handshake

Normally, a BoQ should use the QUIC 1-RTT handshake to establish a BGP session because it does not require any preconditions. In particular, 1-RTT MUST be used when a BGP session is established for the first time.

When the 1-RTT handshake is used, the BoQ FSM only needs to replace the TCP event in the BGP FSM [RFC4271] with the QUIC event according to the mapping in section 4.4.1.

The QUIC has complete security only when it reaches the Handshake Confirmed state. Therefore, the BoQ FSM should allow the BGP Open message to be sent only after receiving the QuicConnectionConfirmed event.

Although to reduce the connection setup delay, QUIC allows application data to be sent before the Handshake Confirmed state is reached. However,the BGP FSM status needs to be changed for security reasons and the same issues as 0-RTT. Therefore, it is not recommended that the BGP Open message be sent before the Handshake Confirmed state is reached.

If the DelayOpen or PassiveQuicEstablished function is configured on the local system, 1-RTT is also required.

4.4.3. 0-RTT Handshake

When the 0-RTT handshake is used, the QUIC client sends a connection establishment request (Initial packet) with a BGP Open Data message. (Referred to as early data and sent using 0-RTT packets) , which means:

(1) After sending an Initial packet, the client enters the BGP OpenSent state.

(2) After receiving the Client Initial packet and sending the Server Initial/Handshake packet, the server may send BGP Open and change the status to BGP OpenConfirmed. At this time, the Server has not reached the Handshake Complete state.

(3) When receiving the Server Initial/Handshake/BGP Open message, the client also reaches the BGP OpenConfirmed state. In this case, the client is not in the Handshake Complete state either.

Therefore, in the 0-RTT handshake, the BoQ FSM can directly skip the BGP Connect and Active states. This minimizes the BGP session setup delay. After a BGP peer is disconnected from the Established state, 0-RTT can be used for re-establishment.

It should be noted that when the BoQ reaches the BGP OpenConfirmed state, because neither the client nor the server reaches the Handshake Complete state, the handshake may fail. Therefore, during the 0-RTT handshake, before the Handshake Confirmed state is reached, that is, before the BoQ FSM receives the QuicConnectionConfirmed event, the BGP KEEPALIVE/UPDATE/ROUTE-REFRESH message MUST NOT be sent but the NOTIFICATION message MAY be sent.

When the 0-RTT handshake is used to establish a BGP session, delete the Connect and Active states and replace Section 8.2.2 of [RFC4271] with the following content.

For details about the event indexes, refer to [RFC4271].

Idle state:

Initially, the BGP peer FSM is in the Idle state. Hereafter, the BGP peer FSM will be shortened to BGP FSM.

In this state, BGP FSM refuses all incoming BGP connections for this peer. No resources are allocated to the peer. In response to a ManualStart event (Event 1) or an AutomaticStart event (Event 3), the local system:

- initializes all BGP resources for the peer connection,

- sets ConnectRetryCounter to zero,

- starts the ConnectRetryTimer with the initial value,

- initiates a QUIC connection to the other BGP peer,In addition, the Initial packet carries the BGP Open packet.

- listens for a connection that may be initiated by the remote BGP peer, and

- changes its state to OpenSent.

For details about how to handle other events in this state, refer to [RFC4271].

OpenSent:

In this state, the BGP OPEN packet has been sent to the neighbor along with the Initial packet.

The start events (Events 1, 3,6,29-31) are ignored in the OpenSent state.

If a ManualStop event (Event 2) is issued in the OpenSent state,the local system:

- sets the ConnectRetryTimer to zero,

- releases all BGP resources,

- drops the QUIC connection,

- sets the ConnectRetryCounter to zero, and

- changes its state to Idle.

In response to the ConnectRetryTimer_Expires event (Event 9), the local system:

- drops the QUIC connection,

- restarts the ConnectRetryTimer,

- initiates a QUIC connection to the other BGP peer. In addition, the Initial packet carries the BGP Open packet.

- continues to listen for a connection that may be initiated by the remote BGP peer, and

- stays in the OpenSent state.

If the local system receives a DelayOpenTimer_Expires event (Event12), the local system:

- sends the NOTIFICATION with the Error Code Finite State Machine Error,

- sets the ConnectRetryTimer to zero,

- stops and clears the DelayOpenTimer (set to zero),

- drops the QUIC connection, - increments the ConnectRetryCounter by 1,

- optionally performs peer oscillation damping if the DampPeerOscillations attribute is set to TRUE, and

- changes its state to IDLE.

If the BGP FSM receives a QuicConnection_Valid event (Event 32),the QUIC connection is processed, and the connection remains in the OpenSent state.

If the BGP FSM receives a Quic_CR_Invalid event (Event 15), the local system rejects the QUIC connection, and the connection remains in the Opensent state.

If the QUIC connection succeeds (Event 34 or Event 35), the local system has sent the OPEN packet to the neighbor,the local system:

- stops the ConnectRetryTimer (if running) and sets the ConnectRetryTimer to zero,

- sets the HoldTimer to a large value, and

- stays in the OpenSent state.

If the TCP connection fails (Event 36), the local system:

- restarts the ConnectRetryTimer (with the initial value),

- releases all BGP resources,

- increments the ConnectRetryCounter by 1,

- optionally performs peer oscillation damping if the DampPeerOscillations attribute is set to TRUE,

- continues to listen for a connection that may be initiated by the remote BGP peer, and

- changes its state to Idle.

If the BGP message header checking (Event 21) or OPEN message checking detects an error (Event 22)(refer to Section 6.2 of [RFC4271]), the local system:

- sends a NOTIFICATION message with the appropriate error code,

- sets the ConnectRetryTimer to zero,

- releases all BGP resources,

- drops the QUIC connection,

- increments the ConnectRetryCounter by 1,

- (optionally) performs peer oscillation damping if the DampPeerOscillations attribute is TRUE, and

- changes its state to Idle.

If a NOTIFICATION message is received with a version error (Event 24), the local system:

- sets the ConnectRetryTimer to zero,

- releases all BGP resources,

- drops the QUIC connection,

- increments the ConnectRetryCounter by 1,

- (optionally) performs peer oscillation damping if the DampPeerOscillations attribute is TRUE, and

- changes its state to Idle.

When an OPEN message is received, all fields are checked for correctness. If there are no errors in the OPEN message (Event 19), the local system:

- sets the BGP ConnectRetryTimer to zero,

- sends a KEEPALIVE message, and

- sets a KeepaliveTimer (via the text below)

- sets the HoldTimer according to the negotiated value (refer to Section 4.2 of [RFC4271]),

- changes its state to OpenConfirm.

In response to any other event (Events 8,10-11,20,23,25-28),the local system:

- sends the NOTIFICATION with the Error Code Finite State Machine Error,

- sets the ConnectRetryTimer to zero,

- releases all BGP resources,

- drops the QUIC connection,

- increments the ConnectRetryCounter by 1,

- (optionally) performs peer oscillation damping if the DampPeerOscillations attribute is set to TRUE, and

- changes its state to Idle.

OpenConfirm:

For details, refer to [RFC4271] . The only modification is to replace TCP Event with QUIC Event. For details, refer to section 4.4.1.

Established:

For details, refer to [RFC4271]. The only modification is to replace TCP Event with QUIC Event. For details, refer to section 4.4.1.

4.5. BGP session management

4.5.1. Error Handling

As shown in section 4.4.1, BoQ error handling involves the following three types of errors:

(1) QUIC error: Includes stream error and connection error [RFC9001]. In some cases, a stream error may cause a connection error. For

example, if an operation error occurs on all streams, the connection error should be triggered to close the connection.

(2) TLS alert: In [RFC9001],a QUIC endpoint MUST treat any alert from TLS as if it were at the "fatal" level. For TLS alerts, this includes replacing any alert with a generic alert, such as handshake_failure (0x128 in QUIC).

(3) BGP error: If an error occurs in BGP processing [RFC4271], it can be mapped to the following BoQ Error Codes[RFC9000].

This document defines some of the following BoQ Error Codes:

(1) BOQ_NO_ERROR (0x00): No error. This is used when the connection or stream needs to be closed, but there is no error to signal.

(2) BOQ_INTERNAL_ERROR (0x01): The BoQ implementation encountered an internal error and is incapable of continuing the stream or the connection.

4.5.2. Session closure

QUIC provides three ways to close a connection (refer to Section 10 of[RFC9000]):

(1) Idle timeout

(2) Immediate Close

(3) Stateless Reset

When the idle timer expires, the connection is closed immediately. Idle timeout can be calculated using the following formula:

idle_timeout = MAX(min_idle_timeout, 3*PTO)

The PTO is a time that the sender should wait for an acknowledgment of a sent packet. For a calculation method, refer to Section 6.2.1 of [RFC9002] .

When establishing a QUIC connection, the transmission parameter max_idle_timeout is used. Endpoints advertise local idle_timeout to each other. If no max_idle_timeout advertisement is received from the remote end, the remote idle_timeout is set to a value of 0. Based on the values of local idle_timeout and remote idle_timeout, there are three possible scenarios:

(1) If both the values are 0, disable the idle timeout function.

(2) If there is only one value 0, set min_idle_timeout to a non-zero value in between.

(3) If neither value is 0, set min_idle_timeout to the smaller value.

Two options are available for the idle timer during BGP session establishment. Option 1 is recommended by default.

Option 1: Set this parameter to 0, indicating that idle timeout is disabled.

Option 2: The value must be greater than the value of BGP HoldTimer. It is recommended that the value be greater than five times the value of BGP HoldTimer.

5. Security Considerations

This document replaces the transport protocol layer of BGP from TCP to QUIC. It does not modify the basic protocol specifications of BGP, and therefore does not introduce new security risks to the basic BGP protocol.

BoQ enhances transport-layer security for BGP sessions, refer to[RFC7475] :

(1) Supports server identity authentication.

(2) (Optional) Supports client identity authentication.

(3) Confidentiality protection of BGP messages is supported. All BGP messages are encrypted for transmission.

(4) Supports integrity protection for BGP messages.

As described in Section 8 of [RFC8446] and Section 9.2 of [RFC9001], the 0-RTT handshake may cause replay attacks. To avoid Replay attacks, the following methods are recommended:

(1) By default, the 0-RTT handshake is not used to establish a BGP session.

(2) If the 0-RTT handshake is used to establish a BGP session, it is recommended that the receiver directly discard the replayed packets in case of replay attacks. This does not affect the BGP session establishment. RFC 8470 also provides detailed analysis and mitigation measures for the risk of replay attacks caused by the use of early TLS data.

6. Contributors

TBD

7. Acknowledgments

TBD.

8. References

[I-D.ietf-quic-applicability]
Kuehlewind, M. and B. Trammell, "Applicability of the QUIC Transport Protocol", draft-ietf-quic-applicability-11 (work in progress), April 2021.

[I-D.ietf-quic-http]
Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", draft-ietf-quic-http-34 (work in progress), February 2021.

[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>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN", RFC 4659, DOI 10.17487/RFC4659, September 2006, <https://www.rfc-editor.org/info/rfc4659>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC7475] Dawkins, S., "Increasing the Number of Area Directors in an IETF Area", BCP 9, RFC 7475, DOI 10.17487/RFC7475, March 2015, <https://www.rfc-editor.org/info/rfc7475>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/info/rfc9001>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/info/rfc9002>.

Authors' Addresses

Shuanglong Chen
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: chenshuanglong@huawei.com

Yongkang Zhang
Huawei Technologies
101 Yuhuatai Software Avenue
Nanjing
China

Email: zhangyongkang@huawei.com

Haibo Wang
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: rainsword.wang@huawei.com

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com