Workgroup: Internet Area Working Group
Internet-Draft:
draft-piraux-intarea-quic-tunnel-session-00
Published: 2 November 2020
Intended Status: Experimental
Expires: 6 May 2021
Authors: M. Piraux O. Bonaventure A. Masputra
UCLouvain UCLouvain Apple Inc.

Session mode for multiple QUIC Tunnels

Abstract

This document specifies methods for grouping QUIC tunnel connections in a single session enabling the exchange of packets of Internet protocols over several QUIC connections.

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 6 May 2021.

Copyright Notice

Copyright (c) 2020 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 (https://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.

Table of Contents

1. Introduction

Mobile devices such as laptops, smartphones or tablets have different requirements than the traditional fixed devices. These mobile devices often change their network attachment. They are often attached to trusted networks, but sometimes they need to be connected to untrusted networks where their communications can be eavesdropped, filtered or modified. In these situations, the classical approach is to rely on VPN protocols such as DTLS or IPSec. These VPN protocols provide the encryption and authentication functions to protect those mobile clients from malicious behaviors in untrusted networks.

Today's mobile devices are often multihomed and many expect to be able to perform seamless handovers from one access network to another without breaking the established VPN sessions. In some situations it can also be beneficial to combine two or more access networks together to increase the available host bandwidth. A protocol such as Multipath TCP [RFC6824] supports those handovers and allows aggregating the bandwidth of different access links. It could be combined with single-path VPN protocols to support both seamless handovers and bandwidth aggregation above VPN tunnels.

Unfortunately, Multipath TCP is not yet deployed on most Internet servers and thus few applications would benefit from such a use case.

This document explores how QUIC could be used to enable multi-homed mobile devices to communicate securely in untrusted networks.

This document is organized as follows. Section 3 describes the reference environment. Then, we propose a new mode of operation, explained in Section 4, that use the recently proposed datagram extension ([I-D.pauly-quic-datagram]) for QUIC to transport plain IP packets over a QUIC connection. Section 5 specifies how a connection is established in this document proposal. Section 6 details the format of the messages introduced by this document.

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. Reference environment

The reference environment is illustrated in Figure 1, in which a client-initiated flow is tunneled through the concentrator.

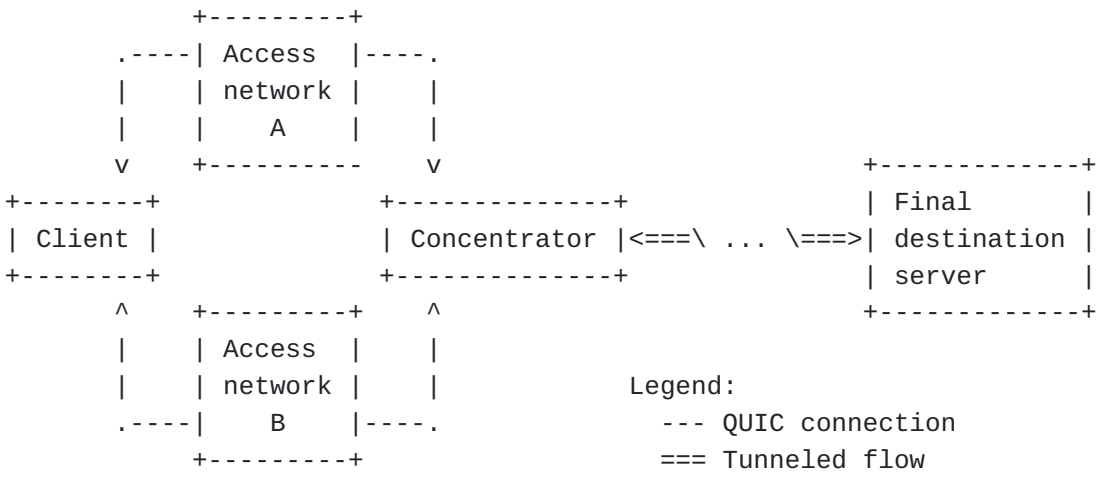

Figure 1: Example environment

Such a multihomed client would like to benefit from the different access networks available to reach the concentrator. These access networks can be used for load-sharing, failover or other purposes. One possibility to efficiently use these two access networks is to rely on the proposed Multipath extensions to QUIC [I-D.deconinck-

quic-multipath]. Another approach is to create one QUIC connection using the single-path QUIC protocol [I-D.ietf-quic-transport] over each access network and glue these different connections together in a single session on the concentrator. Given the migration capabilities of QUIC, this approach could support failover with a single active QUIC connection at a time.

4. The tunnel session mode

The "tunnel session" mode enables the client and the concentrator to exchange packets of several network protocols through the QUIC tunnel connection at the same time. It also leverages the QUIC datagram extension [I-D.pauly-quic-datagram].

This document specifies the following format for encoding packets in QUIC DATAGRAM frame. It allows encoding packets from several protocols by identifying the corresponding protocol of the packet in each QUIC DATAGRAM frame. Figure 2 describes this encoding.

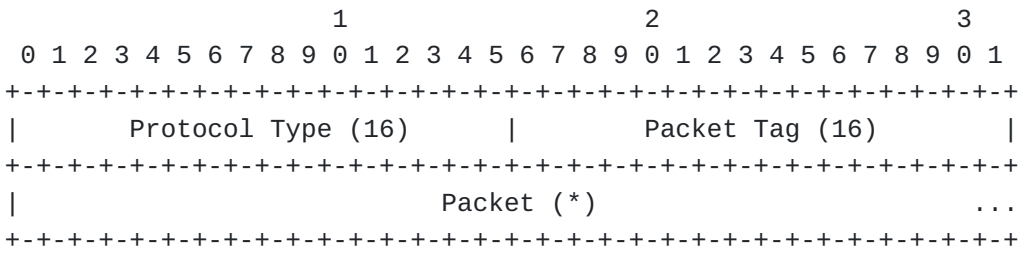

Figure 2: Encoding packets in QUIC DATAGRAM frame

This encoding defines three fields.

* Protocol Type: The Protocol Type field contains the protocol type of the payload packet. The values for the different protocols are defined as "ETHER TYPES" in [IANA-ETHER-TYPES]. A QUIC tunnel that receives a Protocol Type representing an unsupported protocol MAY drop the associated Packet. QUIC tunnel endpoints willing to exchange Ethernet frames can use the value 0x6558 for [Transparent-Ethernet-Bridging].

* Packet Tag: An opaque 16-bit value. The QUIC tunnel application is free to decide its semantic value. For instance, a QUIC tunnel endpoint MAY encode the sending order of packets in the Packet Tag, e.g. as a timestamp or a sequence number, to allow reordering on the receiver.

* Packet: The packet conveyed inside the QUIC tunnel connection.

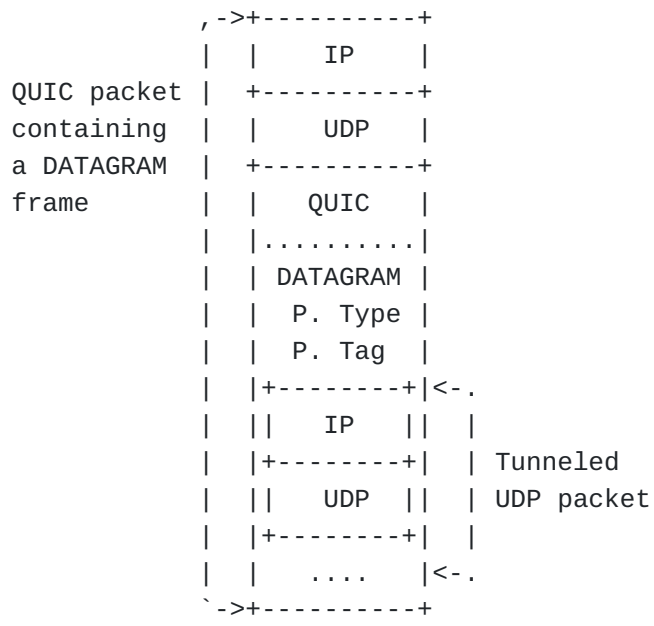

Figure 3: QUIC packet sent by the client when tunneling a UDP packet

Figure 3 illustrates how a UDP packet is tunneled using the tunnel session mode. The main advantage of the tunnel session mode is that it supports IP and any protocol above the network layer. Any IP packet can be transported using the datagram extension over a QUIC connection. However, this advantage comes with a large per-packet overhead since each packet contains both a network and a transport header. All these headers must be transmitted in addition with the IP/UDP/QUIC headers of the QUIC connection. For TCP connections for instance, the per-packet overhead can be large.

4.1. Joining a tunneling session

If the client is multihomed, it can use Multipath QUIC [I-D.deconinck-quic-multipath] to efficiently use its different access networks. This version of the document does not elaborate in details on this possibility. If the concentrator does not support Multipath QUIC, then the client creates several QUIC connections and joins them at the application layer. This works as illustrated in figure Figure 4. Each message is exchanged over a dedicated unidirectional QUIC stream. Their format is detailed in Section 6. When the client opens the first QUIC connection with the concentrator, (1) it can request a QUIC tunnel session identifier. (2) The concentrator replies with a variable-length opaque value that identifies the QUIC tunneling session. When opening a QUIC connection over another access network, (3) the client can send this identifier to join the QUIC tunneling session. The concentrator matches the session identifier with the existing session with the client. It can then use both sessions to reach the client and receive tunneled packets from the client.

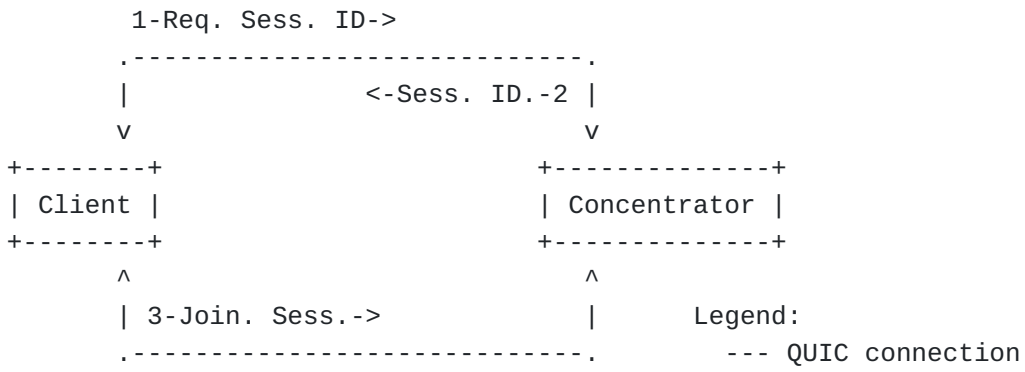

Figure 4: Creating sessions over different access networks

Joining a tunneling session allows grouping several QUIC connections to the concentrator. Each endpoint can then coordinate the use of the Packet Tag across the tunneling session as presented in Section 4.1.1.

Both QUIC tunnel endpoints open their first unidirectional stream (i.e. stream 2 and 3), hereafter named the QUIC tunnel control stream, to exchange these messages. A QUIC tunnel endpoint MUST NOT close its control stream and SHOULD provide enough flow control credit to its peer.

The messages format used for this purpose are described in Section 6. The client initiates the procedure and MAY either start a new session or join an existing session. This negotiation MUST NOT take place more than once per QUIC connection.

4.1.1. Coordinate use of the Packet Tag

When using the tunnel session mode, each packet is associated with a 16-bit value called Packet Tag. This document leaves defining the meaning of this value to implementations. This section provides some examples on how it can be used to implement packet reordering across several QUIC tunnel connections grouped in a tunneling session.

A first simple example of use is to encode the timestamp at which the datagram was sent. Using a millisecond precision and encoding the 16 lower bits of the timestamp makes the value wrapping around in a bit more than 65 seconds.

Another example of use is to maintain a value counting the datagrams sent over all QUIC tunnel connections of the tunneling session. The 16-bit value allows distinguishing at most 32768 packets in flight.

The QUIC tunnel receiver can then distinguish, buffer and reorder packets based on this value. Mechanisms for managing the datagram

buffer and negotiating the use of the Packet Tag are out of scope of this document.

5. Connection establishment

During connection establishment, the tunnel session mode support is indicated by setting the ALPN token "qt-session" in the TLS handshake. Draft-version implementations MAY specify a particular draft version by suffixing the token, e.g. "qt-session-00" refers to the first version of this document.

6. Messages format

In the following sections, we specify the format of each message introduced in this document. They are encoded as TLVs, following the format defined in Section 7 of [I-D.piraux-intarea-quic-tunnel].

6.1. QUIC tunnel control TLVs

This document specifies additional QUIC tunnel control TLVs:

Type	Size	Sender	Mode	Name
0x01	Variable	Client	tunnel session	New Session TLV
0x02	Variable	Concentrator	tunnel session	Session ID TLV
0x03	Variable	Client	tunnel session	Join Session TLV

Figure 5: QUIC tunnel control TLVs

The New Session TLV is used by the client to initiate a new tunneling session. The Session ID TLV is used by the concentrator to communicate to the client the Session ID identifying this tunneling session. The Join Session TLV is used to join a given tunneling session, identified by a Session ID. All QUIC these tunnel control TLVs MUST NOT be sent on other streams than the QUIC tunnel control streams.

When the tunnel session mode is in use, the Access Report TLV defined in Section 7.1.1 of [I-D.piraux-intarea-quic-tunnel] MUST be sent on other streams than the QUIC tunnel control stream.

6.1.1. New Session TLV

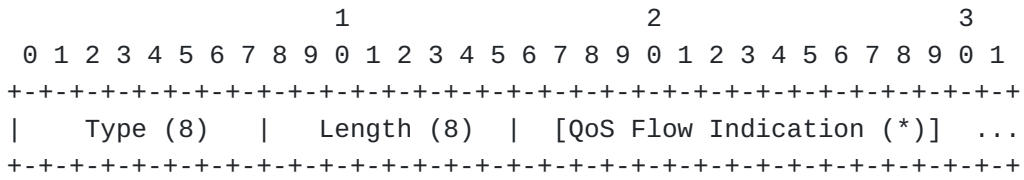

Figure 6: New Session ID TLV

The New Session TLV contains an optional value. It initiates a new tunneling session at the concentrator, and can contain an opaque value giving an indication on the type of traffic conveyed over this session. The concentrator can use this indication for QoS purposes for instance.

The concentrator MUST send a Session ID TLV in response, with the Session ID corresponding to the tunneling session created. After sending a New Session TLV, the client MUST close the QUIC tunnel control stream.

The concentrator MUST NOT send New Session TLVs.

6.1.2. Session ID TLV

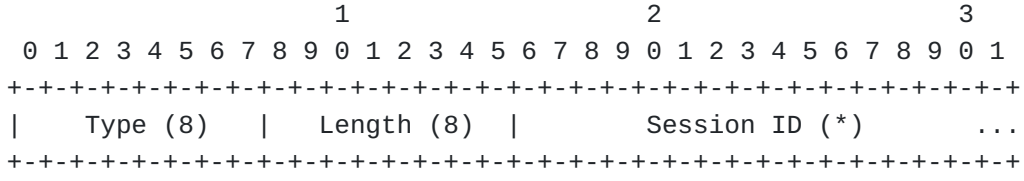

Figure 7: Session ID TLV

The Session ID TLV contains an opaque value that identifies the current tunneling session. It can be used by the client in subsequent QUIC connections to join them to this tunneling session. The concentrator MUST send a Session ID TLV in response of a New Session TLV, with the Session ID corresponding to the tunneling session created.

The client MUST NOT send a Session ID TLV. The concentrator MUST close the QUIC tunnel control stream after sending a Session ID TLV.

6.1.3. Join Session TLV

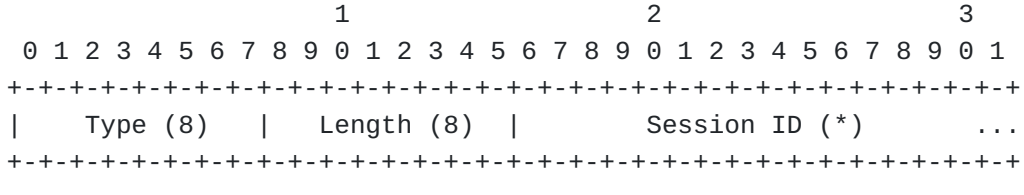

Figure 8: Join Session TLV

The Join Session TLV contains an opaque value that identifies a tunneling session to join. The client can send a Join Session TLV to join the QUIC connection to a particular tunneling session. The tunneling session is identified by the Session ID. After sending a Join Session TLV, the client MUST close the QUIC tunnel control stream.

The concentrator MUST NOT send Join Session TLVs. After receiving a Join Session TLV, the concentrator MUST use the Session ID to join this QUIC connection to the tunneling session. Joining the tunneling session implies merging the state of this QUIC tunnel connection to the session. A successful joining of connection is indicated by the closure of the QUIC tunnel control stream of the concentrator.

In cases of failure when joining a tunneling session, the concentrator MUST send a RESET_STREAM with an application error code discerning the cause of the failure. The possible codes are listed below:

*UNKNOWN_ERROR (0x0): An unknown error occurred when joining the tunneling session. QUIC tunnel endpoints SHOULD use more specific error codes when applicable.

*UNKNOWN_SESSION_ID (0x1): The Session ID used in the Join Session TLV is not a valid ID. It was not issued in a Session ID TLV or refers to an expired tunneling session.

*CONFLICTING_STATE (0x2): The current state of the QUIC tunnel connection could not be merged with the tunneling session.

7. Security Considerations

The security considerations of [I-D.piraux-intarea-quic-tunnel] are also applicable to this document.

8. IANA Considerations

8.1. Registration of QUIC tunnel Identification String

This document creates a new registration for the identification of the QUIC tunnel protocol in the "Application Layer Protocol Negotiation (ALPN) Protocol IDs" registry established in [RFC7301].

The "qt-session" string identifies the QUIC tunnel protocol tunnel session mode.

Protocol: QUIC Tunnel session mode

Identification Sequence:
0x71 0x74 0x2d 0x73 0x65 0x73 0x73 0x69 0x6f 0x6e ("qt-session")

Specification: This document

8.2. QUIC tunnel control TLVs

The following subsections detail new registries within "QUIC tunnel control Parameters" registry.

8.2.1. QUIC tunnel control TLVs Types

This document creates three new registrations to identify the QUIC tunnel control TLVs defined in this document in the "QUIC tunnel control TLVs Types" sub-registry defined in [I-D.piraux-intarea-quic-tunnel].

The values to be added in the registry are as follows:

Code	Name	Reference
1	New Session TLV	[This-Doc]
2	Session ID TLV	[This-Doc]
3	Join Session TLV	[This-Doc]

8.3. QUIC tunnel control Error Codes

This document establishes a registry for QUIC tunnel control stream error codes. The "QUIC tunnel control Error Code" registry manages a 62-bit space. New values are assigned via IETF Review (Section 4.8 of [RFC8126]).

The initial values to be assigned at the creation of the registry are as follows:

Code	Name	Reference
0	UNKNOWN_ERROR	[This-Doc]
1	UNKNOWN_SESSION_ID	[This-Doc]
2	CONFLICTING_STATE	[This-Doc]

9. References

9.1. 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/info/rfc2119>.

[I-D.piraux-intarea-quic-tunnel]
Piraux, M., Bonaventure, O., and A. Masputra, "Tunneling Internet protocols inside QUIC", Work in Progress, Internet-Draft, draft-piraux-intarea-quic-tunnel-00, 2 November 2020, <http://www.ietf.org/internet-drafts/draft-piraux-intarea-quic-tunnel-00.txt>.

[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/info/rfc8174>.

9.2. Informative References

[I-D.pauly-quic-datagram] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", Work in Progress, Internet-Draft, draft-pauly-quic-datagram-05, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-pauly-quic-datagram-05.txt>.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-32, 20 October 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-32.txt>.

[I-D.deconinck-quic-multipath]
Coninck, Q. and O. Bonaventure, "Multipath Extensions for QUIC (MP-QUIC)", Work in Progress, Internet-Draft, draft-deconinck-quic-multipath-05, 20 August 2020, <http://www.ietf.org/internet-drafts/draft-deconinck-quic-multipath-05.txt>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC6824]
Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <https://www.rfc-editor.org/info/rfc6824>.

[IANA-ETHER-TYPES] "IANA ETHER TYPES", https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.txt , 2019.

[Transparent-Ethernet-Bridging]
Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, DOI 10.17487/RFC1701, October 1994, <https://www.rfc-editor.org/info/rfc1701>.

Appendix A. Change Log

Acknowledgments

Authors' Addresses

Maxime Piraux
UCLouvain

Email: maxime.piraux@uclouvain.be

Olivier Bonaventure
UCLouvain

Email: olivier.bonaventure@uclouvain.be

Adi Masputra
Apple Inc.

Email: adi@apple.com