Multiplexed Application Substrate over QUIC Encryption
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

M. K端hlewind
M. Ihlar
M. Westerlund
Ericsson

20 October 2025

Extension to Connect-IP for static IP header compression
draft-kuehlewind-masque-ip-compression-00

Abstract

This document specifies an extension for IP header compression when using Connect-IP proxying.

About This Document

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

The latest revision of this draft can be found at https://mirjak.github.io/draft-masque-ip-compression/draft-kuehlewind-masque-ip-compression.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-kuehlewind-masque-ip-compression/.

Discussion of this document takes place on the Multiplexed Application Substrate over QUIC Encryption Working Group mailing list (mailto:masque@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/masque/. Subscribe at https://www.ietf.org/mailman/listinfo/masque/.

Source for this draft and an issue tracker can be found at https://github.com/mirjak/draft-masque-ip-compression.

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 23 April 2026.

Table of Contents

1. Introduction

This document specifies an extension for IP header compression when using Connect-IP [CONNECT-IP] proxying. It specifies a way to indicate which IP header fields can be omitted and provides reference values that are then used to reconstruct omitted fields when using the indicated context ID.

This document defines four new capsules to assign new context IDs and provide static information in IPv4 and IPv6 headers, acknowlegde the use of such a context ID, or cancel its use. The capsules MUST only be used with Connect-IP [CONNECT-IP].

Extending this work to include compression of transport layer fields like the port numbers in TCP or UDP is left for consideration for further revisions.

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.

2.1. IP4_COMPRESSION_ASSIGN and IP6_COMPRESSION_ASSIGN capsules

The IPv4 Compression Assign capsule (capsule type TBD) is used to register a Context ID and provides reference values of the IP header fields that will be omitted from the HTTP Datagram Payload when using this Context ID. It has the following format:

```
IP4_COMPRESSION_ASSIGN Capsule {
  Type (i) = TBD,
  Length (i),
  Context ID (i),
  IPv4 Flags (16),
  IPv4 Header (20)
}
```

Figure 1: IPv4 Compression Assign Capsule Format

It contains the following fields:

IPv4 Flags: A 16-bit field containing flags that indicate which byte-aligned IP header fields can be omitted. The format is defined in Figure 2.

IPv4 Header: This field contains the first 20 bytes of the IPv4 header. Fields that are not indicated as omitted will be included in the compressed packet and therefore their reference value is not relevant.

The IPv4 Flags field has the following format:

```
IPv4 Flags {
  Version-IHL (1),
  DSCP-ECN (1),
  Total Length (1),
  Identification (1),
  Flags-FragOffset (1),
  TTL (1),
  Protocol (1),
  Checksum (1),
  Source Address (1),
  Destination Address (1),
  Reserved (6),
}
```

Figure 2: IPv4 Flags Format

The flags have the following meanings:

Version-IHL: This flag indicates if byte 0 of the IPv4 header (Version and IHL fields) can be omitted.

DSCP-ECN: This flag indicates if byte 1 of the IPv4 header (DSCP and ECN fields) can be omitted.

Total Length: This flag indicates if bytes 2-3 of the IPv4 header (Total Length field) can be omitted (receiver will compute from payload size).

Identification: This flag indicates if bytes 4-5 of the IPv4 header (Identification field) can be omitted.

Flags-FragOffset: This flag indicates if bytes 6-7 of the IPv4 header (Flags and Fragment Offset fields) can be omitted.

TTL: This flag indicates if byte 8 of the IPv4 header (TTL field) can be omitted.

Protocol: This flag indicates if byte 9 of the IPv4 header (Protocol field) can be omitted.

Checksum: This flag indicates if bytes 10-11 of the IPv4 header (Header Checksum field) can be omitted (receiver will recompute).

Source Address: This flag indicates if bytes 12-15 of the IPv4 header (Source Address field) can be omitted.

Destination Address: This flag indicates if bytes 16-19 of the IPv4 header (Destination Address field) can be omitted.

Reserved: These bits are reserved for future use and MUST be set to zero.

```
IP6_COMPRESSION_ASSIGN Capsule {
  Type (i) = TBD,
  Length (i),
  Context ID (i),
  IPv6 Flags (16),
  IPv6 Header (40)
}
```

Figure 3: IPv6 Compression Assign Capsule Format

It contains the following fields:

IPv6 Flags: A 16-bit field containing flags that indicate which byte-aligned IP header fields can be omitted. The format is defined in Figure 4.

IPv6 Header: This field contains the first 40 bytes of the IPv6 header. Fields that are not indicated as omitted will be included in the compressed packet and therefore their reference value is not relevant.

The IPv6 Flags field has the following format:

```
IPv6 Flags {
  Version-FL (1),
  Traffic Class (1),
  Payload Length (1),
  Next Header (1),
  Hop Limit (1),
  Source Address (1),
  Destination Address (1),
  Reserved (9),
}
```

Figure 4: IPv6 Flags Format

The flags have the following meanings:

Version-FL: This flag indicates if the Version field (upper 4 bits of byte 0) and Flow Label field (lower 4 bits of byte 1, and all of bytes 2-3) can be omitted. When this flag is set but the Traffic Class flag is not set, the Traffic Class field (lower 4 bits of byte 0 and upper 4 bits of byte 1) MUST be preserved as a single byte in the compressed packet.

Traffic Class: This flag indicates if the Traffic Class field (lower 4 bits of byte 0 and upper 4 bits of byte 1) can be omitted. This flag can be set independently of the Version-FL flag to allow omitting Version and Flow Label while preserving Traffic Class.

Payload Length: This flag indicates if bytes 4-5 of the IPv6 header (Payload Length field) can be omitted (receiver will compute from payload size).

Next Header: This flag indicates if byte 6 of the IPv6 header (Next Header field) can be omitted.

Hop Limit: This flag indicates if byte 7 of the IPv6 header (Hop Limit field) can be omitted.

Source Address: This flag indicates if bytes 8-23 of the IPv6 header (Source Address field) can be omitted.

Destination Address: This flag indicates if bytes 24-39 of the IPv6 header (Destination Address field) can be omitted.

Reserved: These bits are reserved for future use and MUST be set to zero.

When the indicated Context ID is used, all IP header fields that are flagged as omitted as well as the version field MUST be omitted from the HTTP datagram payload. The receiver of the datagram with the indicated Context ID MUST reconstruct the complete IP header using the reference values from the COMPRESSION_ASSIGN capsule and/or by computing derived values (such as checksums and length fields) before forwarding the payload.

When an endpoint receives a IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule, it MUST either accept or reject the corresponding registration:

* if it accepts the registration, the receiver MUST return a COMPRESSION_ACK capsule with the Context ID set to the one from the received IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule back to its peer

* if it rejects the registration, the receiver MUST respond by sending a COMPRESSION_CLOSE capsule with the Context ID set to the one from the received IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule.

As mandated in Section 4 of [CONNECT-UDP], clients can only allocate even Context IDs, while proxies can only allocate odd ones. This makes the registration capsules from this document unambiguous. For example, if a client receives a COMPRESSION_ASSIGN capsule with an even Context ID, that has to be an echo of a capsule that the client initially sent, indicating that the proxy accepted the registration. Since the value 0 was reserved by unextended connect-udp, the Context ID value of COMPRESSION_ASSIGN can never be zero.

Endpoints MUST NOT send two COMPRESSION_ASSIGN capsules with the same Context ID. If a recipient detects a repeated Context ID, it MUST treat the capsule as malformed. Receipt of a malformed capsule MUST be treated as an error processing the Capsule Protocol, as defined in Section 3.3 of [HTTP-DGRAM].

Endpoints MAY pre-emptively use Context IDs not yet acknowledged by the peer via COMPRESSION_ACK, knowing that those HTTP Datagrams can be dropped if they arrive before the corresponding IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule, or if the peer rejects the registration.

2.2. IP_COMPRESSION_ACK capsule

The IP Compression Acknowledgment capsule (capsule type TBD) confirms registration of a context ID that was received in an IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule.

```
IP_COMPRESSION_ACK Capsule {
  Type (i) = TBD,
  Length (i),
  Context ID (i),
}
```

Figure 5: IP Compression Acknowledgment Capsule Format

An endpoint can only send a COMPRESSION_ACK capsule if it received a IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule with the same Context ID. If an endpoint receives COMPRESSION_ACK capsule for a context ID it did not attempt to register in an IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule, that capsule is considered malformed.

2.3. IP_COMPRESSION_CLOSE capsule

The IP Compression Close capsule (capsule type TDB) serves two purposes. It can be sent as a direct response to a received IP4_COMPRESSION_ASSIGN or IP6_COMPRESSION_ASSIGN capsule capsule, to indicate that the registration was rejected. It can also be sent later to indicate the closure of a previously assigned registration.

```
IP_COMPRESSION_CLOSE Capsule {
  Type (i) = TBD,
  Length (i),
  Context ID (i),
}
```

Figure 6: IP Compression Close Capsule Format

Once an endpoint has either sent or received a IP_COMPRESSION_CLOSE for a given Context ID, it MUST NOT send any further datagrams with that Context ID. Since the value 0 was reserved by unextended connect-udp, the Context ID value of IP_COMPRESSION_CLOSE can never be zero.

3. Security Considerations

TODO Security

4. IANA Considerations

5. IANA Considerations

This document will request IANA to register the following new items to the "HTTP Capsule Types" registry maintained at <https://www.iana.org/assignments/masque>:

Value	Capsule Type
TBD	IP4_COMPRESSION_ASSIGN
TBD	IP6_COMPRESSION_ASSIGN
TBD	IP_COMPRESSION_ACK
TBD	IP_COMPRESSION_CLOSE

Table 1: New Capsules

All of these new entries use the following values for these fields:

Status: provisional (permanent if this document is approved)
Reference: This document
Change Controller: IETF
Contact: MASQUE Working Group masque@ietf.org
(mailto:masque@ietf.org)
Notes: None

6. Normative References

[CONNECT-IP]
Pauly, T., Ed., Schinazi, D., Chernyakhovsky, A., K端hlewind, M., and M. Westerlund, "Proxying IP in HTTP", RFC 9484, DOI 10.17487/RFC9484, October 2023, <https://www.rfc-editor.org/rfc/rfc9484>.

[CONNECT-UDP]
Schinazi, D., "Proxying UDP in HTTP", RFC 9298, DOI 10.17487/RFC9298, August 2022, <https://www.rfc-editor.org/rfc/rfc9298>.

[HTTP-DGRAM]
Schinazi, D. and L. Pardue, "HTTP Datagrams and the Capsule Protocol", RFC 9297, DOI 10.17487/RFC9297, August 2022, <https://www.rfc-editor.org/rfc/rfc9297>.

[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>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Mirja K端hlewind
Ericsson
Email: mirja.kuehlewind@ericsson.com

Marcus Ihlar
Ericsson
Email: marcus.ihlar@ericsson.com

Magnus Westerlund
Ericsson
Email: magnus.westerlund@ericsson.com