MASQUE
Internet-Draft
Intended status: Standards Track
Expires: 19 September 2026

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

18 March 2026

ECN and DSCP support for HTTPS's Connect-UDP
draft-westerlund-masque-connect-udp-ecn-dscp-02

Abstract

HTTP's Extended Connect's Connect-UDP protocol enables a client to proxy a UDP flow from the HTTP server towards a specified target IP address and UDP port. QUIC and Real-time transport protocol (RTP) are examples of transport protocols that use UDP and support Explicit Congestion Notification (ECN) and provide the necessary feedback. This document specifies how ECN and DSCP can be supported through an extension to the Connect-UDP protocol for HTTP without per-packet byte overhead, soley using Context IDs.

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://gloinul.github.io/masque-ecn/#go.draft-westerlund-masque-connect-udp-ecn.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-westerlund-masque-connect-udp-ecn-dscp/.

Discussion of this document takes place on the MASQUE 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/gloinul/masque-ecn.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 19 September 2026.

Copyright Notice

Table of Contents

1. Introduction

Connect-UDP, as currently defined, limits the Explicit Congestion Notification (ECN) [RFC3168] exchange between the HTTP server and the target. There is no support for carrying the ECN bits between the HTTP Connect-UDP client and the HTTP server proxying the UDP flow. Thus, it is not possible to establish the end-to-end ECN information flow necessary to support either classic ECN [RFC3168] or L4S [RFC9330], [RFC9331].

Diffserv [RFC2475] enables differential network treatment of packets. Connect-UDP, as currently defined, lacks support for carrying the DSCP field [RFC2474] through the tunnel.

This document specifies a Connect-UDP extensions that enable end-to-end ECN and DSCP for proxied connections: the zero-bytes extension adds no per-packet overhead by encoding the ECN and DSCP values directly into Context IDs. This document specifies negotiation to provide an intial set of Context IDs and capsules for dynamic updates.

An alternative to this extension is Connect-IP [RFC9484]; however, it carries a full IP header between the HTTP client and server, resulting in significantly more overhead than this extension.

This extension is defined such that they allow clients to optimistically start sending UDP packets in HTTP Datagrams, i.e. before receiving the response to its UDP proxying request, as described in Section 5 of [RFC9298].

2. Conventions

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. Zero-byte Combined ECN and DSCP Extension

For a zero-overhead encoding, the ECN and DSCP bits are indicated by using different Context IDs. An example use of three additional Context IDs to only encode the ECN bit used together with a DSCP of 0 is shown in Table 1.

Context ID Value	ECN bit	ECN Value	DSCP Value
0	0b00	Not-ECT	0
2	0b01	ECT(1)	0
4	0b10	ECT(0)	0
6	0b11	CE	0

Table 1: ECN-only Encoding Table

No new Context ID value is defined to represent Not-ECT, since using a Context ID without this extension would, by default, imply Not-ECT. Additionally, Context IDs are defined to represent the combination of an ECN value other than Not-ECT and the DSCP values. If an application uses more DCSP values than just zero, additional Context IDs must be defined.

This extension results in four times as many Context IDs within a single Connect-UDP stream for each DSCP values used. We expect that this is acceptable in most cases, as a total of 31 client initiated Context IDs can be encoded in a single byte, thus resulting in no packet expansion for applications that use upto 8 DSCP values.

An endpoint enabling this extension MUST define all three ECN values, even if the ECN-enabled application expects that only one ECT value (and CE) is used. This is because of transmission errors or erroneous remarking in the network, where the other ECT codepoint, as well as Not-ECT, may be observed.

Negotiation of the context ID values is defined using both HTTP headers and capsules in Section 4.

4. Negotiating Extensions Usage

This section defines capability negotiation and Context ID configuration for the zero-bytes combined ECN and DSCP extensions.

Note that Context Identifiers are defined as QUIC varints (see Section 16 of [RFC9000]) and support values up to 4,611,686,018,427,387,903 (2^62-1), which is larger than what a Structure Header Integer supports (limited to 999,999,999,999,999). We foresee no issues with this limitation, as Context Identifiers should primarily use the single-byte representation for efficiency, i.e., they should rarely exceed 63.

4.1. ECN DSCP Context Assignment Format

In this extension four Context IDs need to be configured for each DSCP value.

A configuration of ECN and DSCP signaling is represented by a five-tuple with the following format:

```
ECN_DSCP_CONTEXT_ASSIGNMENT {
  DSCP_VALUE (6),
  NOT_ECN_CONTEXT (i)
  ECT_1_CONTEXT (i),
  ECT_0_CONTEXT (i),
  CE_CONTEXT (i),
}
```

Figure 1: ECN DSCP CONTEXT ASSIGNMENT Format

DCSP_VALUE: The DSCP value in the IP valid for the following Context IDs.

NOT_ECN_CONTEXT: The Context ID used to indicate the payload was marked as Not-ECN-Capable.

ECT_1_CONTEXT: The Context ID used to indicate the payload was marked with ECN value ECT(1).

ECT_0_CONTEXT: The Context ID used to indicate the payload was marked with ECN value ECT(0).

CE_CONTEXT: The Context ID used to indicate the payload was marked with ECN value CE.

4.1.1. HTTP Structured field

ECN-DSCP-Context-ID is a Structured Header Field [RFC9651]. Its value is a List consisting of zero or more Inner Lists, where the Inner List contains five Integer Items. The Integer Items MUST be non-negative, as they are DSCP values and Context ID defined in [RFC9298]. The DSCP value and four ECN Context IDs are those defined in Figure 1, in that order.

When the header is included in an Extended Connect request, it indicates, first of all, support for this ECN extension. Secondly, it may define one or more 5-item inner lists of DSCP values and corresponding ECN Context IDs for the requestor-to-responder direction. If no 5-item inner lists of Context IDs are included, then this header only indicates support for the extension, and the Context IDs MAY be signaled using capsules.

When the request includes the ECN-DSCP-Context-ID header, the responder MAY include this header in the response. If included with one or more 5-item inner lists, it defines Context ID defined by the server, usable in either direction.

The following example indicates support for this extension and defines two sets of client initiated Context IDs: ID= 10, 12, 14, 16 (Not-ECN-Capable, ECT(1), ECT(0), CE) combined with DSCP 46 for Expedited Forwarding (EF): 46 ; and ID=0, 4, 6, 8 combined with the default DSCP value of 0. Note that the default context ID of 0 is (re-)used to indicate the default ECN value of Not-ECN Capable.

```
ECN-Context-ID: (46,8,10,12,14), (0,0,2,4,6 )
```

Figure 2: Example of ECN-DSCP-Context-ID header

4.1.2. ECN DSCP Context ID Assignment and ACK Capsules

The ECN_DSCP_CONTEXT_ASSIGN capsule is used to assign additional Context ID values after negotiation and initial assignment in the HTPP header.

```
ECN_DSCP_CONTEXT_ASSIGN Capsule {
  Type (i) = TBA_1
  Length (i),
  ECN_DSCP_CONTEXT_ASSIGNMENT (..) ...,
}
```

Figure 3: ECN_DSCP_CONTEXT_ASSIGN Capsule Format

Type and Length as defined by Section 3.2 of the HTTP Capsule specification [RFC9297]. The capsule value is the ECN_DSCP_CONTEXT_ASSIGNMENT defined above in Figure 1. Thus, the capsule value consists of zero or more ECN_DSCP_CONTEXT_ASSIGNMENT five-tuples.

The ECN_DSCP_CONTEXT_ACK capsule confirms the registration of a context IDs that were received via a ECN_DSCP_CONTEXT_ASSIGN capsule.

```
ECN_DSCP_CONTEXT_ACK Capsule {
  Type (i) = TBA_2
  Length (i),
  ECN_DSCP_CONTEXT_ASSIGNMENT (..) ...,
}
```

Figure 4: ECN_DSCP_CONTEXT_ACK Capsule Format

An endpoint only send a ECN_DSCP_CONTEXT_ACK capsule if it received a ECN_DSCP_CONTEXT_ASSIGN capsule with the same ECN_DSCP_CONTEXT_ASSIGNMENT. If an endpoint receives ECN_DSCP_CONTEXT_ACK capsule for a ECN_DSCP_CONTEXT_ASSIGNMENT it did not attempt to register, that capsule is considered malformed.

5. Tunnels and DSCP and ECN marking interactions

5.1. Tunnel Endpoint Marking

The Tunnel Endpoint, when receiving an IP/UDP packet belonging to a Connect-UDP request with the ECN DSCP extension enabled, the DSCP value two ECN bits in the incoming IP/UDP packet are used to select the appropriate Context ID. If a non-yet-know DSCP value is received the endpoint can register a new Context ID assignments using the ECN_DSCP_CONTEXT_ASSIGN capsule and optimistically start us them.

The Tunnel Endpoint on egress sets the DSCP that belongs to the received Context ID and corresponding ECN values in the IP packet it creates for this UDP Proxying payload.

A Tunnel endpoint which is unable to read or set the ECN Field SHALL NOT enable the ECN extension.

5.2. DSCP Remarking Considerations

The tunnel may interconnect two different administrative domains that use DSCP values differently. Thus, the endpoints likely need to perform remarking of DSCP field values, similar to what an inter-domain router would. Depending on use cases and deployment, the HTTP client can be in different network domains with different DSCP usages. An HTTP server that, based on user identification, connects the HTTP client to different network domains behind it may also need to support multiple external domains.

The above complications in handling DSCP make it impossible to provide a standardized remarking instruction. Instead, the deployment will have to define whether remarking is handled by the HTTP server, the HTTP client, or both, considering the tunnel a specific network domain in itself.

5.3. Tunnel Transport Connection ECN Interactions and Congestion Control

The primary goal of the ECN DSCP extension is to enable ECN usage between the proxy and the target and to have the end-to-end transport react to that ECN. However, different potential models exist for providing ECN interactions for the tunnel, i.e., between the HTTP client and server. The choice depends on how the tunnel is configured and what additional support has been implemented for the Connect-UDP protocol.

The default deployment would be to use congestion controlled transport protocols between the HTTP endpoints or proxies for the tunneled ECN enabled packets. This include all HTTP versions before HTTP/3 [RFC9114], as well as HTTP/3 sending packets over reliable streams as well as over congestion controlled datagrams. In this deployment on the ingress to each congestion controlled transport an Active Queue Management (AQM) is recommend that can mark the tunneled packet's ECN field (or drop them) in case there is sufficient queue build up.

For tunnels using HTTP/3 with datagrams, where the QUIC connection disables congestion control on packets containing HTTP datagrams as discussed in Section 6 of [RFC9298], the ECN marking on tunneled packets can be propagated between the IP packet of the transport connection and the end-to-end packet. This represents a specific implementation of IP-in-IP tunnels with tightly coupled shim headers as discussed in [RFC9601]. It is implemented as Feed-Forward-and-Up as discussed in [RFC9599], and MUST use the normal mode on tunnel ingress and follow the specified default behavior on egress as defined in [RFC6040].

5.4. Tunnel Transport Connection DSCP Interactions

For HTTP tunnels not using HTTP/3 [RFC9114], HTTP/3 using reliable streams, or HTTP/3 with datagrams but without disabling congestion control, the tunnel will consist of one or possibly several chained congestion-controlled transport connections. These transport connections can use only a single DSCP code point to avoid inconsistent network treatment that might confuse the congestion controller and retransmission mechanism. Thus, even if the tunneled packets use different DSCP values, the transport connection must settle on a single, suitable DSCP value. However, if the QUIC multipath extension [I-D.ietf-quic-multipath] is used, each path can have a different DSCP value. In this latter case, packets with different DSCP values can be mapped to different paths with the appropriate network treatment as indicated by their DSCP values.

For tunnels using HTTP/3 with datagrams and where the QUIC connection disables congestion control on packets containing HTTP datagrams, as discussed in Section 6 of [RFC9298], the QUIC packets can be marked using the most suitable DSCP value based on the encapsulated packet. In cases where the tunnel connection is sent into a different network domain than the one on which the tunneled packet was received, a suitable remapping must occur for the domain to which the tunnel packet will be sent. The HTTP tunnel MUST NOT coalesce different tunneled payloads that are not mapped to the same DSCP in a single QUIC packet.

5.5. QUIC Aware Forwarding

An HTTP endpoint that supports this extension and QUIC Aware Forwarding [I-D.ietf-masque-quic-proxy] MUST preserve ECN markings on forwarded packets in both directions to ensure end-to-end ECN functionality. Using this extension in combination with QUIC Aware Forwarding, rather than relying solely on the latter, also ensures that ECN black holes do not occur, for example, on long-header packets or packets sent before the QUIC Aware Forwarding path is established for short-header packets. Thus, supporting both provides a consistent ECN experience.

6. IANA Considerations

6.1. HTTP Field Names

IANA is requested to register one new permanent Field name in the Hypertext Transfer Protocol (HTTP) Field Name Registry (At time of writing residing at: https://www.iana.org/assignments/http-fields/http-fields.xhtml).

6.1.1. DSCP-ECN-Context-ID

Field Name: DSCP-ECN-Context-ID

Status: Permanent

Structured Type: List

Reference: RFC-TO-BE

6.2. HTTP Capsule Type

IANA is requested ot register two new HTTP Capsule Types in the permanent range (0x00-0x3f).

6.2.1. ECN_DSCP_CONTEXT_ASSIGN

Value: TBA_1

Capsule Type: ECN_CONTEXT_ASSIGN

Status: permanent

Reference: RFC-TO-BE

Change Controller: IETF

Contact: MASQUE Working Group masque@ietf.org

Notes: None

6.2.2. DSCP_ECN_CONTEXT_ACK

Value: TBA_2

Capsule Type: DSCP_ECN_CONTEXT_ACK

Status: permanent

Reference: RFC-TO-BE

Change Controller: IETF

Contact: MASQUE Working Group masque@ietf.org

Notes: None

7. References

7.1. Normative References

[I-D.ietf-masque-quic-proxy]
Pauly, T., Rosenberg, E., and D. Schinazi, "QUIC-Aware Proxying Using HTTP", Work in Progress, Internet-Draft, draft-ietf-masque-quic-proxy-08, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-quic-proxy-08>.

[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>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/rfc/rfc2474>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/rfc/rfc3168>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/rfc/rfc6040>.

[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>.

[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/rfc/rfc9000>.

[RFC9114] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/rfc/rfc9114>.

[RFC9297] 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>.

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

[RFC9331] De Schepper, K. and B. Briscoe, Ed., "The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)", RFC 9331, DOI 10.17487/RFC9331, January 2023, <https://www.rfc-editor.org/rfc/rfc9331>.

[RFC9599] Briscoe, B. and J. Kaippallimalil, "Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP", BCP 89, RFC 9599, DOI 10.17487/RFC9599, August 2024, <https://www.rfc-editor.org/rfc/rfc9599>.

[RFC9601] Briscoe, B., "Propagating Explicit Congestion Notification across IP Tunnel Headers Separated by a Shim", RFC 9601, DOI 10.17487/RFC9601, August 2024, <https://www.rfc-editor.org/rfc/rfc9601>.

[RFC9651] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 9651, DOI 10.17487/RFC9651, September 2024, <https://www.rfc-editor.org/rfc/rfc9651>.

7.2. Informative References

[I-D.ietf-quic-multipath] Liu, Y., Ma, Y., De Coninck, Q., Bonaventure, O., Huitema, C., and M. K端hlewind, "Managing multiple paths for a QUIC connection", Work in Progress, Internet-Draft, draft-ietf-quic-multipath-21, 17 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-multipath-21>.

[I-D.schinazi-masque-connect-udp-ecn] Schinazi, D., "An ECN Extension to CONNECT-UDP", Work in Progress, Internet-Draft, draft-schinazi-masque-connect-udp-ecn-02, 28 March 2022, <https://datatracker.ietf.org/doc/html/draft-schinazi-masque-connect-udp-ecn-02>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/rfc/rfc2475>.

[RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <https://www.rfc-editor.org/rfc/rfc8311>.

[RFC9330] Briscoe, B., Ed., De Schepper, K., Bagnulo, M., and G. White, "Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture", RFC 9330, DOI 10.17487/RFC9330, January 2023, <https://www.rfc-editor.org/rfc/rfc9330>.

[RFC9484] 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>.

Authors' Addresses

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

Marten Seemann
Smallstep
Email: martenseemann@gmail.com

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

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