

Workgroup: Network Working Group  
Internet-Draft:  
draft-schinazi-masque-connect-udp-ecn-00  
Published: 12 December 2020  
Intended Status: Standards Track  
Expires: 15 June 2021  
Authors: D. Schinazi  
Google LLC

## **An ECN Extension to CONNECT-UDP**

### **Abstract**

The CONNECT-UDP method allows proxying UDP packets over HTTP. This document describes an extension to CONNECT-UDP that allows conveying ECN information on proxied UDP packets.

Discussion of this work is encouraged to happen on the MASQUE IETF mailing list ([masque@ietf.org](mailto:masque@ietf.org)) or on the GitHub repository which contains the draft: <https://github.com/DavidSchinazi/draft-connect-udp-ecn>.

### **Discussion Venues**

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

Source for this draft and an issue tracker can be found at <https://github.com/DavidSchinazi/draft-connect-udp-ecn>.

### **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 15 June 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](#)
  - [1.1. Conventions and Definitions](#)
- [2. ECN Header Definition](#)
- [3. Datagram Encoding of Proxied UDP Packets](#)
- [4. Stream Encoding of Proxied UDP Packets](#)
- [5. HTTP Intermediaries](#)
- [6. Security Considerations](#)
- [7. IANA Considerations](#)
  - [7.1. HTTP Header](#)
  - [7.2. Stream Chunk Type Registration](#)
- [8. Normative References](#)
- [Acknowledgments](#)
- [Author's Address](#)

## 1. Introduction

The CONNECT-UDP [[CONNECT-UDP](#)] method allows proxying UDP packets over HTTP. This document describes an extension to CONNECT-UDP that allows conveying ECN [[ECN](#)] information on proxied UDP packets.

Discussion of this work is encouraged to happen on the MASQUE IETF mailing list ([masque@ietf.org](mailto:masque@ietf.org)) or on the GitHub repository which contains the draft: <https://github.com/DavidSchinazi/draft-connect-udp-ecn>.

### 1.1. 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. ECN Header Definition

"ECN" is a Item Structured Header [[STRUCT-HDR](#)]. Its value MUST be a Boolean. Its ABNF is:

ECN = sf-boolean

The "ECN" header indicates whether the sender supports this extension. A value of 1 indicates support whereas a value of 0 (or the absence of the header) indicates lack of support.

Clients MUST NOT indicate support for this extension unless they know that the protocol running over UDP that is being proxied supports ECN, and will react appropriately to Congestion Experienced (CE) markings.

Proxies MUST NOT indicate support for this extension unless they know they have the ability to read and write the IP ECN bits on its target-bound UDP sockets.

This extension is said to have been negotiated when both client and proxy indicated support for it in their CONNECT-UDP request and response.

### **3. Datagram Encoding of Proxied UDP Packets**

If a client supports this extension and HTTP/3 datagrams [[H3DGRAM](#)], it can attempt to use datagrams for ECN information. This is done by allocating four datagram flow identifiers (as opposed to one in traditional CONNECT-UDP) and communicating them to the proxy using parameters on the "Datagram-Flow-Id" header. These parameters are "ect0", "ect1" and "ce". For example:

Datagram-Flow-Id = 100; ect0=102; ect1=104; ce=106

If the proxy wishes to support datagram encoding of this extension, it echoes those parameters in its CONNECT-UDP response. The main flow identifier now represents Not-ECT, whereas the one in "ect0" represents ECT(0), "ect1" represents ECT(1) and "ce" represents CE; see Section 5 of [[ECN](#)] for the definition of these IP header fields.

When the proxy receives a datagram from the given flow identifier, it sets the IP packet's ECN bits accordingly on the UDP packet it sends to the target. Similarly, in the other direction the flow identifier represents which ECN bits were seen on the UDP packets received from the target.

### **4. Stream Encoding of Proxied UDP Packets**

If HTTP/3 datagrams are not supported, the stream is used to convey UDP payloads, and the CONNECT-UDP Stream Chunk Type is used to indicate the values of the ECN bits, as defined below:

Value	Type	ECN Field
0x00	UDP_PACKET	Not-ECT
0x31	UDP_PACKET_ECT0	ECT(0)
0x32	UDP_PACKET_ECT1	ECT(1)
0x33	UDP_PACKET_CE	CE

The proxy then uses the the CONNECT-UDP Stream Chunk Type on received UDP payloads to set the ECN bits on the IP packets it sends to the target, and in the reverse direction to indicate which ECN bits received from the target.

## 5. HTTP Intermediaries

HTTP/3 DATAGRAM flow identifiers are specific to a given HTTP/3 connection. However, in some cases, an HTTP request may travel across multiple HTTP connections if there are HTTP intermediaries involved; see Section 2.3 of [[RFC7230](#)].

Intermediaries that support this extension and HTTP/3 datagrams MUST negotiate flow identifiers separately on the client-facing and server-facing connections. This is accomplished by having the intermediary parse the "Datagram-Flow-Id" header on all CONNECT-UDP requests it receives, and sending the same value in the "Datagram-Flow-Id" header on the response. The intermediary will perform this individually for all the parameters defined by this extension as well, in addition to the rules in the "HTTP Intermediaries" section of [[CONNECT-UDP](#)].

## 6. Security Considerations

This document does not have additional security considerations beyond those defined in [[CONNECT-UDP](#)].

## 7. IANA Considerations

### 7.1. HTTP Header

This document will request IANA to register the "ECN" header in the "Permanent Message Header Field Names" registry maintained at <https://www.iana.org/assignments/message-headers>.

Header Field Name	Protocol	Status	Reference
ECN	http	std	This document

## 7.2. Stream Chunk Type Registration

This document will request IANA to register the following entry in the "CONNECT-UDP Stream Chunk Type" registry [[CONNECT-UDP](#)]:

Value	Type	Description	Reference
0x31	UDP_PACKET_ECT0	UDP payload with ECT(0)	This document
0x32	UDP_PACKET_ECT1	UDP payload with ECT(1)	This document
0x33	UDP_PACKET_CE	UDP payload with CE	This document

## 8. Normative References

- [[CONNECT-UDP](#)] Schinazi, D., "The CONNECT-UDP HTTP Method", Work in Progress, Internet-Draft, draft-ietf-masque-connect-udp-01, 28 August 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-masque-connect-udp-01.txt>>.
- [[ECN](#)] 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/info/rfc3168>>.
- [[H3DGRAM](#)] Schinazi, D., "Using QUIC Datagrams with HTTP/3", Work in Progress, Internet-Draft, draft-schinazi-quick-h3-datagram-05, 12 October 2020, <<http://www.ietf.org/internet-drafts/draft-schinazi-quick-h3-datagram-05.txt>>.
- [[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>>.
- [[RFC7230](#)] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and

Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014,  
<<https://www.rfc-editor.org/info/rfc7230>>.

[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>>.

[STRUCT-HDR] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", Work in Progress, Internet-Draft, draft-ietf-httpbis-header-structure-19, 3 June 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-httpbis-header-structure-19.txt>>.

## Acknowledgments

This proposal was inspired directly or indirectly by prior work from many people. The author would like to thank contributors the MASQUE working group.

## Author's Address

David Schinazi  
Google LLC  
1600 Amphitheatre Parkway  
Mountain View, California 94043,  
United States of America

Email: [dschinazi.ietf@gmail.com](mailto:dschinazi.ietf@gmail.com)