Workgroup: MASQUE
Internet-Draft:
draft-schinazi-connect-udp-listen-02
Published: 2 March 2023
Intended Status: Standards Track
Expires: 3 September 2023
Authors: D. Schinazi
Google LLC
A. Singh
Google LLC

Proxying Listener UDP in HTTP

Abstract

The mechanism to proxy UDP in HTTP only allows each UDP Proxying request to transmit to a specific host and port. This is well suited for UDP client-server protocols such as HTTP/3, but is not sufficient for some UDP peer-to-peer protocols like WebRTC. This document proposes an extension to UDP Proxying in HTTP that enables such use-cases.

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://DavidSchinazi.github.io/draft-schinazi-connect-udp-listen/draft-schinazi-connect-udp-listen.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-schinazi-connect-udp-listen/.

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/DavidSchinazi/draft-schinazi-connect-udp-listen.

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 3 September 2023.

Table of Contents

1. Introduction
  1.1. Conventions and Definitions
2. Proxied UDP Listener Mechanism
3. HTTP Datagram Payload Format
4. The connect-udp-listen Header Field
5. Proxy behavior
6. Security Considerations
7. IANA Considerations
8. References
  8.1. Normative References
  8.2. Informative References
Appendix A. Example
Appendix B. Comparison with CONNECT-IP
Acknowledgments
Authors' Addresses

1. Introduction

The mechanism to proxy UDP in HTTP [CONNECT-UDP] allows creating tunnels for communicating UDP payloads [UDP] to a fixed host and port. Combined with the HTTP CONNECT method (see Section 9.3.6 of [HTTP]), it allows proxying the majority of a Web Browser's HTTP traffic. However WebRTC [WebRTC] relies on ICE [ICE] to provide connectivity between two Web browsers, and ICE relies on the ability to send and receive UDP packets to multiple hosts. While in theory it might be possible to accomplish this using multiple UDP Proxying HTTP requests, HTTP semantics [HTTP] do not guarantee that distinct requests will be handled by the same server. This can lead to the

UDP packets being sent from distinct IP addresses, thereby preventing ICE from operating correctly. Consequently, UDP Proxying requests cannot enable WebRTC connectivity between peers.

This document describes an extension to UDP Proxying in HTTP that allows sending and receiving UDP payloads to multiple hosts within the scope of a single UDP Proxying HTTP request.

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.

This document uses terminology from [CONNECT-UDP] and notational conventions from [QUIC]. This document uses the terms Integer and List from Section 3 of [STRUCTURED-FIELDS] to specify syntax and parsing.

2. Proxied UDP Listener Mechanism

In unextended UDP Proxying requests, the target host is encoded in the HTTP request path or query. For Listener UDP Proxying, it is instead conveyed in each HTTP Datagram, see Section 3.

When performing URI Template Expansion of the UDP Proxying template (see Section 3 of [CONNECT-UDP]), the client sets both the target_host and the target_port variables to the '*' character (ASCII character 0x2A).

Before sending its UDP Proxying request to the proxy, the client allocates an even-numbered context ID, see Section 4 of [CONNECT-UDP]. The client then adds the "connect-udp-listen" header field to its UDP Proxying request, with its value set as the allocated context ID, see Section 4.

3. HTTP Datagram Payload Format

When HTTP Datagrams [HTTP-DGRAM] associated with this Listener UDP Proxying request contain the context ID in the connect-udp-listen header field, the format of their UDP Proxying Payload field (see Section 5 of [CONNECT-UDP]) is defined by Figure 1:

```
Listener UDP Proxying Payload {
  IP Version (8),
  IP Address (32..128),
  UDP Port (16),
  UDP Payload (..),
}
```

Figure 1: Listener UDP Proxying HTTP Datagram Format

IP Version: The IP Version of the following IP Address field. **MUST** be 4 or 6.

IP Address: The IP Address of this proxied UDP packet. When sent from client to proxy, this is the target host to which the proxy will send this UDP payload. When sent from proxy to client, this represents the source IP address of the UDP packet received by the proxy. This field has a length of 32 bits when the corresponding IP Version field value is 4, and 128 when the IP Version is 6.

UDP Port: The UDP Port of this proxied UDP packet in network byte order. When sent from client to proxy, this is the target port to which the proxy will send this UDP payload. When sent from proxy to client, this represents the source UDP port of the UDP packet received by the proxy.

UDP Payload: The unmodified UDP Payload of this proxied UDP packet (referred to as "data octets" in [UDP]).

4. The connect-udp-listen Header Field

The "connect-udp-listen" header field’s value is an Integer. It is set as the Context ID allocated for Listener UDP Proxying; see Section 2. Any other value type **MUST** be handled as if the field were not present by the recipients (for example, if this field is defined multiple times, its type becomes a List and therefore is to be ignored). This document does not define any parameters for the connect-udp-listen header field value, but future documents might define parameters. Receivers **MUST** ignore unknown parameters.

5. Proxy behavior

After accepting the Connect-UDP Listener proxying request, the proxy uses a UDP port to transmit UDP payloads received from the client to the target IP Address and UDP Port specified in each Listener Datagram Payload received from the client. The proxy uses the same port to listen for UDP packets from any authorized target and encapsulates the packets in the Listener Datagram Payload format, specifying the IP and port of the target and forwards it to the client.

6. Security Considerations

The security considerations described in Section 7 of [CONNECT-UDP] also apply here. Since TURN can be run over this mechanism, implementors should review the security considerations in Section 21 of [TURN].

Since unextended UDP Proxying requests carry the target as part of the request, the proxy can protect unauthorized targets by rejecting requests before creating the tunnel, and communicate the rejection reason in response header fields. Listener UDP Proxying requests do not have this ability. Therefore, proxies **MUST** validate the target on every datagram and **MUST NOT** forward individual datagrams with unauthorized targets. Proxies can either silently discard such datagrams or abort the corresponding request stream.

7. IANA Considerations

This document will request IANA to register the following entry in the "HTTP Field Name" registry maintained at <https://www.iana.org/assignments/http-fields>:

Field Name: connect-udp-listen
Template: None
Status: provisional (permanent if this document is approved)
Reference: This document
Comments: None

8. References

8.1. Normative References

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

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[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>.

[QUIC] 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>.

[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>.

[STRUCTURED-FIELDS] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/rfc/rfc8941>.

[UDP] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/rfc/rfc768>.

8.2. Informative References

[CONNECT-IP] Pauly, T., Schinazi, D., Chernyakhovsky, A., Kühlewind, M., and M. Westerlund, "Proxying IP in HTTP", Work in Progress, Internet-Draft, draft-ietf-masque-connect-ip-08, 1 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-connect-ip-08>.

[ICE] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/rfc/rfc8445>.

[TURN] Reddy, T., Ed., Johnston, A., Ed., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 8656, DOI 10.17487/RFC8656, February 2020, <https://www.rfc-editor.org/rfc/rfc8656>.

[WebRTC] "WebRTC", W3C Recommendation, 26 January 2021, <https://www.w3.org/TR/webrtc/>.

Appendix A. Example

In the example below, the client is configured with URI Template "https://example.org/.well-known/masque/udp/{target_host}/{target_port}/" and wishes to use WebRTC with another browser over a listener UDP Proxying tunnel. It contacts a STUN server at 192.0.2.42. The STUN server, in response, sends the proxy's IP address to the other browser at 203.0.113.33. Using this

information, the other browser sends a UDP packet to the proxy, which is proxied over HTTP back to the client.

```
 Client                                                Server

 STREAM(44): HEADERS            -------->
   :method = CONNECT
   :protocol = connect-udp
   :scheme = https
   :path = /.well-known/masque/udp/*/*/
   :authority = proxy.example.org
   connect-udp-listen = 2
   capsule-protocol = ?1

 DATAGRAM                       -------->
   Quarter Stream ID = 11
   Context ID = 2
   IP Version = 4
   IP Address = 192.0.2.42
   UDP Port = 1234
   UDP Payload = Encapsulated UDP Payload

            <--------  STREAM(44): HEADERS
                         :status = 200
                         capsule-protocol = ?1

/* Wait for STUN server to respond to UDP packet. */

            <--------  DATAGRAM
                         Quarter Stream ID = 11
                         Context ID = 2
                         IP Version = 4
                         IP Address = 192.0.2.42
                         UDP Port = 1234
                         UDP Payload = Encapsulated UDP Payload

/* Wait for the STUN server to send the proxy's IP and */
/* port to the other browser and for the other browser */
/* to send a UDP packet to the proxy. */

            <--------  DATAGRAM
                         Quarter Stream ID = 11
                         Context ID = 2
                         IP Version = 4
                         IP Address = 203.0.113.33
                         UDP Port = 4321
                         UDP Payload = Encapsulated UDP Payload
```

Appendix B. Comparison with CONNECT-IP

While the use-cases described in Section 1 could be supported using IP Proxying in HTTP [CONNECT-IP], it would require that every HTTP Datagram carries a complete IP header. This would lead to both inefficiencies in the wire encoding and reduction in available Maximum Transmission Unit (MTU). Furthermore, Web browsers would need to support IPv4 and IPv6 header generation, parsing, validation and error handling.

Acknowledgments

This proposal is the result of many conversations with MASQUE working group participants.

Authors' Addresses

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

Email: dschinazi.ietf@gmail.com

Abhi Singh
Google LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America

Email: abhisinghietf@gmail.com