Workgroup: MASQUE
Internet-Draft:
draft-kuehlewind-masque-connect-ip-01
Published: 12 July 2021
Intended Status: Standards Track
Expires: 13 January 2022
Authors: M. Kuehlewind Ericsson
M. Westerlund Ericsson
M. Ihlar Ericsson
Z. Sarker Ericsson

The CONNECT-IP HTTP method for proxying IP traffic

Abstract

This draft specifies a new HTTP method CONNECT-IP to proxy IP traffic. CONNECT-IP uses HTTP/3 Datagrams to use QUIC Datagrams for efficient transport of proxied IP packets, with the possibility to fallback to HTTP/3 over reliable QUIC streams, or even HTTP 1.x and 2.

CONNECT-IP supports two modes: a tunneling mode where IP packets are forwarded without modifications and flow forwarding mode which supports optimization for individual IP flows forwarded to the targeted peer. To request tunneling or flow forwarding, a client connects to a proxy server by initiating a HTTP/3 connection and sends a CONNECT-IP request which either indicates the address of the proxy or the target peer. The proxy then forwards payload received on that stream or in an HTTP datagram with a certain stream ID.

Discussion Venues

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

Discussion of this document takes place on the MASQUE Working Group mailing list (masque@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/masque/.

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

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 13 January 2022.

Copyright Notice

Table of Contents

1. Introduction

This document specifies the CONNECT-IP method for IPv4 [RFC0791] and IPv6 [RFC8200] tunneling and flow forwarding over HTTP/3.

CONNECT-IP supports two modes: a tunneling mode where IP packets are forwarded without modifications and flow forwarding mode which supports optimization for individual IP flows forwarded to the targeted peer.

1.1. Tunnel mode

In tunnel mode the client requests to tunnel IP packets to and from one or more servers via the proxy. The Connect-IP request to the proxy establishes such a tunnel and optionally indicates the IP address or IP address range that will be allowed to be used by and forwarded to the client.

The tunnel mode is indicated by the ":authority" pseudo-header field of the CONNECT-IP request contain the host and listing port of the proxy itself. In this mode the proxy just blindly forwards all payload on its external interface without any modification and also forwards all incoming traffic to registered clients as payload within the respective tunnel association. That means all incoming traffic, where the destination address matches an by the client indicated IP address or range of IP addresses, is forwarded to the client over the tunnel association, except a more specific flow forwarding association exists where both destination and source IP address as well as any additionally used identifier match (see section Section 5.5).

However, a proxy MUST offer this service only for known clients and clients MUST be authenticated during connection establishment. The proxy SHOULD inspect the source IP address of the IP packet in the tunnel payload and only forward if the IP address matches the set of client IP addresses. Optionally, a proxy also MAY offer this service only for a limited set of target addresses. In such a case the proxy SHOULD also inspect the destination IP address of the tunnel payload as well as the source address of incoming packets from target servers and reject packets with unknown addresses with an error.

1.2. Flow Forwarding mode

In flow forwarding mode the CONNECT-IP method establishes an outgoing IP flow, from the MASQUE server's external address to the target server's address specified by the client for a particular upper layer protocol. This mode also enables reception and relaying of the reverse IP flow from the target address to the MASQUE server to ensure that return traffic can be received by the client. However, it does not support flow establishment by an external peer. This specification supports forwarding of incoming traffic to one of the clients only if an active mapping has previously been created based on an IP-CONNECT request. Clients that need to support reception of flows established by external peer need to use tunnel mode.

This mode covers the point-to-point use case [I-D.ietf-masque-ip-proxy-reqs] and allows for flow-based optimizations and a larger effective maximum packet size through the tunnel. The target IP address is provided by the client as part of the CONNECT-IP request. The sources address is either independently selected by the proxy or can be requested to be either the same as used in a previous and currently active CONNECT-IP request or different from currently requests by the same client. The client also indicates the upper layer protocol, thus defining the three tuple used as primary selector for the flow.

In this mode the payload between the client and proxy does not contain the IP header in order to reduce overhead. Any additional information (other than the source and destination IP addresses and ports as well as the upper layer protocol identifier) that is needed to construct the IP header or to inform the client about information from received IP packets can be signalled as part of the CONNECT-IP request or using HTTP/3 Datagram [I-D.ietf-masque-h3-datagram] later.

In flow forwarding mode, usually one upper-layer end-to-end connection is associated to one CONNECT-IP forwarding association. While it would be possible for a client to use the same forwarding association for multiple end-to-end connections to the same target

server, as long as they all require the same Protocol (IPv4) / Next Header (IPv6) value, this would lead to the use of the same flow ID for all connections. As such, this is not recommended for connection-oriented transmissions. In order to enable multiple flow forwarding associations to the same server, the flow forwarding mode supports a way to specify some additional upper layer protocol selectors, e.g. TCP source and destination port, to enable multiple CONNECT-IP request for the same three tuple, see CONN-ID header Section 3.5.

The default model for address handling in this specification is that the proxy (Masque Server) will have a pool of one or more IP addresses that it can lend to the MASQUE client and routable over its external interface. Other potential use cases and address handling are possible, potentially requiring further extensions.

This proposal is based on the analysis provided in [I-D.westerlund-masque-transport-issues] indicating that most information in the IP header is either IP flow related or can or even should be provided by the proxy as the IP communication endpoint without the need for input from the client. The most crucial information identified that requires client interaction is ECN [RFC3168] and ICMP [RFC0792] [RFC4443] handling.

This document defines the following IP header field treatment.

Required to be determined in Connect-IP request and response:

- *IP version
- *IP Source Address
- *IP Destination Address (target address)
- *Upper Layer Protocol (IPv4 Protocol field / IPv6 Next Header field)

Can be chosen by Proxy on transmission:

- *IPv6 Flow label (per Connect-IP flow mode request)
- *IPv4 Time to live / IPv6 Hop Limit (proxy configured)
- *Diffserv Codepoint, default is set to 0 (Best Effort)

May optionally be provided on a per packet basis

- *Explicit Congestion Notification in both directions.

The consequence of this is certain limitations that future extension can address. For packets that are sent from the target server to the client, the client will not get any information on the actual value of TTL/Hop Count, DSCP, or flow label when received by the proxy. Instead these field are set and consumed by the proxy only.

Signalling of other dedicated values may be desired in certain deployments, e.g for DCSP [RFC2474]. However, DSCP is in any case a challenge due to local domain dependency of the used DSCP values and the forwarding behavior and traffic treatment they represent. Future use cases for DSCP, as well as new IPv6 extension headers or destination header options [RFC8200] may require additional signaling. Therefore, it is important that the signaling is extensible.

1.2.1. Motivation of IP flow model for flow forwarding

The chosen IP flow model is selected due to several advantages:

* Minimized per packet overhead: The per packet overhead is reduced to basic framing of the IP payload for each IP packet and flow identifiers. This enables a larger effective Maximum Transmission Unit (MTU) than tunnel mode.

* Shared functionality with CONNECT-UDP: The UDP flow proxying functionality of CONNECT-UDP will need to establish, store and process the same IP header related fields and state. So this can be accomplished by simply removing the UDP specific processing of packets.

* CONNECT-IP can establish a new IP flow in 0-RTT: No network related latencies in establishing new flow.

Disadvantages of this model are the following:

* Client to server focused solution: Accepting non-solicited peer-initiated traffic is not supported.

1.3. Definitions

* Proxy: This document uses proxy as synonym for the MASQUE Server or an HTTP proxy, depending on context.

* Client: The endpoint initiating a MASQUE tunnel and IP relaying with the proxy.

* Target host: A remote endpoint the client wishes to establish bi-directional communication with via tunnelling over the proxy.

*IP proxying: A proxy forwarding IP payloads to a target for an IP flow. Data is decapsulate at the proxy and amended by a IP header before forwarding to the target. Packet boundaries need to be preserved or signalled between the client and proxy.

*IP flow: A flow of IP packets between two hosts as identified by their IP addresses, and where all the packets share some properties. These properties include source/destination address, protocol / next header field, flow label (IPv6 only), and DSCP per direction.

Address = IP address

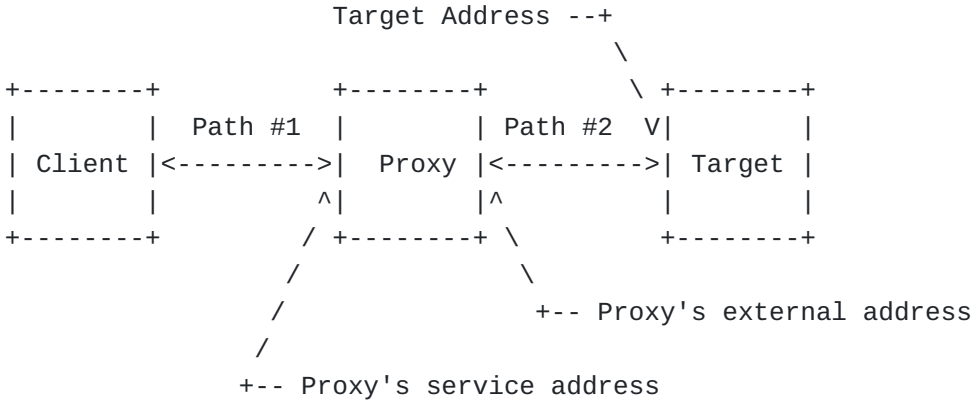

Figure 1: The nodes and their addresses

Figure 1 provides an overview figure of the involved nodes, i.e. client, proxy, and target host. There are also two network paths. Path #1 is the client to proxy path, where IP proxying is provided over an HTTP/3 session, usually over QUIC, to tunnel IP flow(s). Path #2 is the path between the proxy and the target.

The client will use the proxy's service address to establish a transport connection on which to request IP proxying using HTTP/3 CONNECT-IP. The proxy will then relay the client's IP flows to the target host. The IP header from the proxy to the target carries the proxy's external address as source address and the target's address as destination address.

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. The CONNECT-IP method

This document defines a new HTTP [I-D.ietf-httpbis-semantics] method CONNECT-IP to convert streams into tunnels or initialize HTTP datagram flows [I-D.ietf-masque-h3-datagram] to a forwarding proxy. Each stream can be used separately to establish forwarding to potentially different remote hosts. Unlike the HTTP CONNECT method, CONNECT-IP does not request the proxy to establish a TCP connection to the remote target host. Instead the tunnel payload will be forwarded as individual IP packets (tunnel mode) or right on top of the IP layer (flow forwarding), meaning the proxy has to identify messages boundaries to each message before forwarding (see section Section 5).

This document specifies CONNECT-IP for HTTP following the same semantics as the CONNECT method. As such a CONNECT-IP request MUST be constructed as follows:

- The ":method" pseudo-header field is set to "CONNECT-IP"
- The ":scheme" and ":path" pseudo-header fields are omitted
- The ":authority" pseudo-header field contains either the host address to connect to (equivalent to the authority-form of the request-target of CONNECT-UDP [I-D.ietf-masque-connect-udp] or CONNECT requests; see Section 3.2.3 of [I-D.ietf-httpbis-messaging]) or the host and port of the proxy if tunnel mode is requested

A CONNECT request that does not conform to these restrictions is malformed; see Section 4.1.3 of [I-D.ietf-quic-http].

Unlike the CONNECT method, CONNECT-IP does not sequentially trigger a connection establishment process from the proxy to the target host. Therefore, the client does not need to wait for an HTTP response in order to send forwarding data, unless in tunnel mode and requesting assignment of an external IP address. However, the client, especially on tunnel mode, SHOULD limit the amount of traffic sent to the proxy before a 2xx (Successful) response is received.

The forwarding stays active as long as the respective stream is open. A Forwarded IP packet can be either an encapsulated HTTP datagram on the same HTTP stream as the CONNECT-IP request, or as a HTTP datagram sent over QUIC datagram.

2.1. Data encapsulation

Once the CONNECT-IP method has completed, only CAPSULE [I-D.ietf-masque-h3-datagram] frames are permitted to be sent on that stream.

Extension frames MAY be used if specifically permitted by the definition of the extension. Receipt of any other known frame type MUST be treated as a connection error of type H3_FRAME_UNEXPECTED.

Each HTTP Datagram frame contains one of the below specified data formats (Section 2.2) depending on request forwarding mode and given headers and parameters.

Stream based forwarding provides in-order and reliable delivery but may introduce Head of Line (HoL) Blocking if independent messages are send over the same CONNECT-IP association. On streams payload data is encapsulated in the CAPSULE Frame using the DATAGRAM capsule (type=0x02) [I-D.ietf-masque-h3-datagram].

The client can, in addition to stream-based forwarding, request use of HTTP/3 datagrams [I-D.ietf-masque-h3-datagram].

To request datagram support the client sends H3_DATAGRAM SETTINGS parameter with a value of 1 [I-D.ietf-masque-h3-datagram]. Datagram support MUST only be requested when the QUIC datagram extension [I-D.ietf-quic-datagram] was successfully negotiated during the QUIC handshake.

Datagrams provide un-ordered and unreliable delivery. In theory both, stream- as well as datagram-based forwarding, can be used in parallel, however, for most transmissions it is expected to only use one.

While IP packets sent over streams only have to respect the end-to-end MTU between the client and the target server, packets sent in datagrams are further restricted by the QUIC packet size of the QUIC tunnel and any overhead within the QUIC tunnel packet. Ideally, the proxy can provide MTU and overhead information to the client. The client MUST take the estimated overhead into account when indicating the MTU to the application (see section Section 6.3).

2.2. Datagram Formats

This section defines the different datagram formats used by Connect-IP. Even if only one format is currently used it is expected that for some usages future extension may require the flexibility to use multiple different formats for a given CONNECT-IP request.

2.2.1. Tunnel Mode IPv4 Format

The Datagram contains one full IPv4 Packet per [RFC0791]. Used in tunnel mode and when the IP Version is 4 per the IP-Version header or explicit given target address.

2.2.2. Tunnel Mode IPv6 Format

The Datagram contains one full IPv6 Packet per [RFC8200]. Used in tunnel mode and when the IP Version is 6 per the IP-Version header or explicit given target address.

2.2.3. Flow Forwarding Format

The Datagram contains only the IP payload. This is defined as the payload following the IPv4 header and any options for IPv4, and for IPv6 as the payload following the IPv6 header and any extension header. Used for Flow Forwarding mode.

2.2.4. ICMP Message Format

This datagram contains a summary message of the ICMP message received and validated for the respective IP flow. The message format carries the ICMP packet for ICMPv4 [RFC0792] or ICMPv6 [RFC4443]. This format is chosen for forward compatibility. From an implementation perspective the client don't need to verify the checksum or validate the header fields because that is done by the server. However, some type codes, like IMCPv4 type 2, (Packet Too Big) carries an MTU field that the implementation want to read beyond understanding the meaning of the type and code combination.

3. HTTP Headers

Note: This section should be improved by clarifying if headers are in request, response or both.

3.1. IP-Protocol Header for CONNECT-IP

In order to construct the IP header the the proxy needs to fill the "Protocol" field in the IPv4 header or "Next header" field in the IPv6 header. As the IP payload is otherwise mostly opaque to the proxy, this information has to be provided by the client for each CONNECT-IP request for flow forwarding.

IP-Protocol is a Item Structured Header [RFC8941]. Its value MUST be an Integer. Its ABNF is:

```
IP-Protocol = sf-integer
```

3.2. IP-Version header for CONNECT-IP

IP-Version is a Item Structured Header [RFC8941]. Its value MUST be an Integer and either 4 or 6. This information is used by the proxy to check if the requested IP version is supported by the network that the proxy is connected to, as well as to check the destination or source IP address for compliance.

```
IP-Version = sf-integer
```

3.3. IP-Address header for CONNECT-IP

IP-Address is an Item Structured Header [RFC8941]. Its value MUST be an String contain an IP address or IP address range of the same IP version as indicated in the IP-Version header. The address must be specified in the format specified by TBD.

This header is used to request the use of a certain IP address or IP address range by the client to be used as source IP address in tunnel mode. If the IP-Address header is not presented, the proxy is implicitly requested to assign an IP address or IP address range and provide this information to the client with the HTTP response.

If the the client does not provide an IP address or IP address range is has to wait for the proxy response before any payload data can be sent in tunnel mode. If the request is denied by the proxy, any sent payload data will be discarded and a new CONNECT-IP request has to be sent.

The header is also used as a response header from the proxy to the client to indicate the actual IP address or IP address range that should be used by the client in tunnel mode or will be used by the proxy in flow forwarding mode.

```
IP-Address = sf-string
```

3.4. IP-Address-Handling Header for CONNECT-IP

This header can be used to request the use of a stable address for multiple active flow forwarding associations. The first association will be established with an IP selected by the proxy unless also the IP-Address header (Section 3.3) is provided and accepted by proxy. However, additional forwarding association can be requested by the client to use the same IP address as a previous request by specifying the stream ID as value in this header. This header can also be used to ensure that a "new", not yet for this client used address is selected by setting a value that is larger than the maximum stream ID.

IP-Address-Handling is a Item Structured Header [RFC8941]. Its value MUST be an Integer and indicates the stream ID of the corresponding active flow forwarding association. Its ABNF is:

```
IP-Address-Handling = sf-integer
```

3.5. Conn-ID Header for CONNECT-IP

This document further defines a new header field to be used with CONNECT-IP "Conn-ID". The Conn-ID HTTP header field indicates the value, offset, and length of a field in the IP payload that can be used by the proxy as a connection identifier in addition to the IP address and protocol tuple when multiple connections are proxied to the same target server for incoming traffic on the service address.

Conn-ID is a Item Structured Header [RFC8941]. Its value MUST be a Byte Sequence. Its ABNF is:

```
Conn-ID = sf-binary
```

The following parameters are defined:

* A parameter whose name is "offset", and whose value is an Integer indicating the offset of the identifier field starting from the beginning of a datagram or HTTP frame on the forwarding stream.

* A parameter whose name is "length", and whose value is an Integer indicating the length of the identifier field starting from the offset.

Both parameters MUST be present and the header MUST be ignored if these parameter are not present.

This function can be used to e.g. indicate the source port field in the IP payload when containing a TCP packet.

4. Client Connect-IP Request

4.1. Requesting flow forwarding

To request flow forwarding, the client sends a CONNECT-IP request to the forwarding proxy indicating the target host and port in the ":authority" pseudo-header field. The host portion is either an IP literal encapsulated within square brackets, an IPv4 address in dotted-decimal form, or a registered name. Further the CONNECT-IP request MUST contain the IP-Protocol header (Section 3.1) and MAY contain the IP-Address-Handling (Section 3.4) or the Conn-ID (Section 3.5) header.

4.2. Requesting tunnel mode

In tunnel mode, the CONNECT-IP request MUST contain the IP-Version header to indicate if IPv4 or IPv6 is used for the IP packet in the tunnel payload. Further, the request MAY contain an IP-Address header to request use of an IP address or IP address range.

5. MASQUE server behavior

Upon the establishment of a HTTP Connection with the proxy on its service addresses. HTTP level capabilities will be exchanged in the HTTP SETTINGS frame. This will determine if support of datagrams is indicated. If indicated by the client, the MASQUE server SHALL send a H3_DATAGRAM SETTINGS parameter with a value of 1 to indicates is support.

A MASQUE server that receives an IP-CONNECT request examines the target URL to determine if this request is for tunnel or flow forwarding mode. Based on the mode it determines if the required headers are present and which of the optional headers that are included.

The proxy maintains a database with mappings between the HTTP connections and stream IDs and the IP level selectors and Conn-ID information. Using this database and the pool of available addresses and the requests IP-Address-Handling, Conn-ID, IP-Version, IP-Address headers (if included) to select a source IP address. This selection for flow forwarding mode is further discussed below in Section 5.2. For Tunnel Mode, the proxy determine if the proposed IP address per IP-Version and IP-Address headers is possible to use if included, else selects a otherwise unused address from its pool. For tunnel mode the IP selector for incoming traffic for this HTTP Connection and Stream ID is simply the IP destination address.

Once the mapping is successfully established, the proxy sends a HEADERS frame containing a 2xx series status code to the client. The response MUST contain an IP-Address header indicating the outgoing source IP address or IP address range selected by the proxy.

All Datagram capsules received on that stream as well as all HTTP/3 datagrams belonging to this CONNECT-IP association are processed for forwarding to the target server. For flow forwarding mode the Datagram is processed as specified in Section 5.3 to produce IP packets that can be forwarded. For tunnel-mode the complete IP packet are extracted from the Datagram and then forwarded as specified in Section 5.4.

IP packets received from the target server are mapped to an active forwarding connection and are respectively forwarded in an CAPSULE DATAGRAM frame or HTTP/3 datagram to the client (see section Section 5.5 below).

5.1. Error handling

TBD (e.g. out of IP address, conn-id collision)

5.2. IP address selection in flow forwarding mode

In flow forwarding mode the proxy constructs the IP header when sending the IP payload towards the target server and it selects an source IP address from its pool of IP addresses that are routed to the MASQUE server.

If no additional information about a payload field that can be used as an identifier based on Conn-ID header is provided by the client, the proxy uses the source/destination address and protocol ID 3-tuple in order to map an incoming IP packet to an active forwarding connection. The proxy MUST also consider if IP-Address-Handling header Section 3.4 is included and its value. If the IP-Address-Handling header is not included and the there has been prior request the proxy SHOULD give the client the same source Address as the first flow forwarding request. Given these constraints the MASQUE proxy MUST select a source IP address that leads to a unique tuple, and if that is not possible an error response is generated. The same IP address MAY be used for different clients when those client connect to different target servers. However, this also means that potentially multiple IP address are used for the same client when multiple connection to one target server are needed. This can be problematic if the source address is used by the target as an identifier. Therefore it is RECOMMENDED that clients are given unique addresses unless a large fraction of the pool has been exhausted.

If the Conn-ID header is provided, the proxy should use that field as an connection identifier together with protocol ID, source and destination address, as a 4-tuple. In this case it is recommended to use a stable IP address for each client, while the same IP address might still be used for multiple clients, if not indicated differently by the client in the configuration file. Note that if the same IP address is used for multiple clients, this can still lead to an identifier collision and the IP-CONNECT request MUST be reject if such a collision is detect.

Note: Are we allowing multiple client's to share the same 3-tuple when using Conn-ID? It might be good for privacy reasons however, it significantly increases the collision risk.

5.3. Constructing the IP header in flow forwarding mode

To retrieve the source and destination address the proxy looks up the mapping for the datagram flow ID or stream identifier. The IP version, flow label, DiffServ codepoint (DSCP), and hop limit/TTL is selected by the proxy. The IPv4 Protocol or IPv6 Next Header field is set based on the information provided by the IP-Protocol header in the CONNECT-IP request.

The proxy MUST set the Don't Fragment (DF) flag in the IPv4 header. Payload that does not fit into one IP packet MUST be dropped. A dropping indication should be provided to the client. Further the proxy should provide MTU information.

The ECN field is by default set to non-ECN capable transport (non-ECT). Further ECN handling is described in Section Section 6.1.

5.4. Decapsulation of tunnel mode IP Packets

On receiving an HTTP Datagram containing any of the tunnel mode formats for IPv4 or IPv6 the proxy extracts the full IP packet.

The proxy MUST verify that the extracted IP packet's source IP address matches any address associated with this CONNECTION-IP request, i.e. the assigned address or IP range. This is to prevent source address spoofing in tunnel mode.

Further the proxy should verify that the IP header length field correspond to the extracted packets length.

5.5. Receiving an IP packet

When the proxy receives an incoming IP packet on the external interface(s), it checks the packet selectors to find the mappings that match the given packet.

If a client has a tunnel as well as multiple flow forwarding associations, the proxy need to check the mappings for the flow forwarding associations first, and only send it over the the tunnel association if no active flow forwarding is found.

If one or more mappings exists, it further checks if this mapping contains additional identifier information as provided by the Conn-ID Header of the CONNECT-IP request. If this field maps as well, the IP payload is forwarded to the client. If no active mapping is found, the IP packet is discarded.

The above is achieve by using the selector with the most number of fields that match the packet.

If both datagram and stream based forwarding is supported, it is recommended for the proxy to use the same encapsulation as most recently used by the client or datagrams as default. Further considerations might be needed here.

6. Additional signalling

Context ID as defined by [I-D.ietf-masque-h3-datagram] can be used to provide additional per association or per-payload signals. As [I-

D.ietf-masque-h3-datagram] is still work in progress, registration and use of Context IDs is left for future work at this point.

6.1. ECN

ECN requires coordination with the end-to-end communication points as it should only be used if the endpoints are also capable and willing to signal congestion notifications to the other end and react accordingly if a congestion notification is received.

The probing and verification in the upper layer protocol of end-to-end ECN requires per packet control over what value is set on IP packet transmission as well as which of all values are received by the proxy. The QUIC specification is providing one such example in Section 13.4 of [RFC9000]. Thus in flow forwarding mode the proxy needs to be able to set and read the ECN values in sent and received IP packets respectively. This may motivate that this functionality is optional to implement, even if supporting CONNECT-IP implementations in general will need to handle IP packets and their fields with fine grained control. If optional some negotiation mechanism is needed.

Possible realizations are:

a) always have two bits before payload in flow forwarding model, e.g. by including the whole Type of Service (TOS) byte, which would also enable DSCP setting and reading.

b) use 4 different context IDs depending on what ECN field value was received or should be set.

This is work in process and will be further specified in a future version of this document.

6.2. ICMP handling

ICMP messages are directly forwarded in tunneling mode. In flow forwarding mode a ICMP datagram format (Section 2.2.4) is used to send the information from some ICMP message to the client.

The proxy upon receiving an ICMP message with a destination of an IP address it performs flow forwarding on it needs to process the ICMP message. First it should validate that the ICMP message and find if it matches any of its IP flow selectors (including Conn-ID). In case there are multiple matching use the IP selector with the most number of field that matches fully.

Some messages may be applicable both to the proxy and the client. For example an verified ICMPv6 Packet Too Big is applicable both to the proxy and the client. Others like ICMPv6 Destination Unreachable

(Type=1), Code=3 (Address unreachable) and Code=4 (Port unreachable) is only possible to act on by the client.

QUESTION: Which ICMP messages should be suppressed by the proxy?

If a matching IP selector was chosen, then lookup the mapping for the HTTP connection and Stream ID which this message should be sent to. Encapsulate the received ICMP message in the ICMP datagram format and send it to the client.

6.3. MTU considerations

The use of QUIC as a encapsulation between the client and proxy introduces additional overhead. If datagrams are used to encapsulate packets between the proxy and client, the end-to-end packets must fit within one datagram but the size of the datagrams is limited by the tunneling encapsulation overhead.

In forwarding mode the client is usually also the tunnel endpoint that knows about the tunnel overhead and can therefore restrict the size of the packets on the end-to-end connection accordingly. However, the target endpoint is usually not aware of the tunnel overhead. Additional signalling on the end-to-end connection from the client to the target endpoint might be needed to restrict the packet size. If QUIC is also used as end-to-end protocol, this could be realized by the transport parameter. In additional, signal from the proxy to the client could be provided as an extension to indicate the tunnel overhand more accurately and flexibly over time. Such signalling might the realized on the HTTP layer in order to take any additional limitations by HTTP intermediates into account.

If the proxy receives an incoming packet from a target endpoint that is too big to fit within a datagram on the tunnel connection, the proxy MAY either forward the packet encapsulated in the CAPSULE frames on the respective stream or, if IPv4 with DF bit set or IPv6 is used, the proxy MAY reject the packet and send an ICMPv4 Packet type 3 code 4, or ICMPv6 Too Big (PTB) message.

7. Examples

TBD

8. Security considerations

This document does currently not discuss risks that are generic to the MASQUE approach.

Any CONNECT-IP specific risks need further consideration in future, especially when the handling of IP functions is defined in more detail.

9. IANA considerations

9.1. HTTP Method

This document (if published as RFC) registers "CONNECT-IP" in the HTTP Method Registry maintained at <https://www.iana.org/assignments/http-methods>.

Method Name	Safe	Idempotent	Reference
CONNECT-IP	no	no	This document

9.2. HTTP Header

This document (if published as RFC) registers the following header in the "Permanent Message Header Field Names" registry maintained at <https://www.iana.org/assignments/message-headers>.

Header Field Name	Protocol	Status	Reference
Conn-ID	http	exp	This document
IP-Protocol	http	exp	This document
IP-Address	http	exp	This document
IP-Address-Handling	http	exp	This document
IP-Verison	http	exp	This document

Acknowledgments

References

Normative References

[I-D.ietf-httpbis-messaging] Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP/1.1", Work in Progress, Internet-Draft, draft-ietf-httpbis-messaging-16, 27 May 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-messaging-16>.

[I-D.ietf-httpbis-semantics] Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP Semantics", Work in Progress, Internet-Draft, draft-ietf-httpbis-semantics-16, 27 May 2021,

<https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-semantics-16>.

[I-D.ietf-masque-connect-udp] Schinazi, D., "The CONNECT-UDP HTTP Method", Work in Progress, Internet-Draft, draft-ietf-masque-connect-udp-03, 5 January 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-connect-udp-03>.

[I-D.ietf-masque-h3-datagram] Schinazi, D. and L. Pardue, "Using QUIC Datagrams with HTTP/3", Work in Progress, Internet-Draft, draft-ietf-masque-h3-datagram-02, 26 May 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-h3-datagram-02>.

[I-D.ietf-quic-datagram] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-datagram-02, 16 February 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-datagram-02>.

[I-D.ietf-quic-http] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-34, 2 February 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/rfc/rfc791>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/rfc/rfc792>.

[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>.

[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>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/rfc/rfc4443>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/rfc/rfc8200>.

[RFC8941] 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>.

[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>.

Informative References

[I-D.ietf-masque-ip-proxy-reqs] Chernyakhovsky, A., McCall, D., and D. Schinazi, "Requirements for a MASQUE Protocol to Proxy IP Traffic", Work in Progress, Internet-Draft, draft-ietf-masque-ip-proxy-reqs-02, 30 April 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-ip-proxy-reqs-02>.

[I-D.westerlund-masque-transport-issues] Westerlund, M., Ihlar, M., Sarker, Z., and M. Kuehlewind, "Transport Considerations for IP and UDP Proxying in MASQUE", Work in Progress, Internet-Draft, draft-westerlund-masque-transport-issues-02, 12 July 2021, <https://datatracker.ietf.org/doc/html/draft-westerlund-masque-transport-issues-02>.

[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>.

Authors' Addresses

Mirja Kuehlewind
Ericsson

Email: mirja.kuehlewind@ericsson.com

Magnus Westerlund
Ericsson

Email: magnus.westerlund@ericsson.com

Marcus Ihlar
Ericsson

Email: marcus.ihlar@ericsson.com

Zaheduzzaman Sarker
Ericsson

Email: zaheduzzaman.sarker@ericsson.com