httpbis
Internet-Draft
Intended status: Standards Track
Expires: 30 March 2026

奥 一穂 (K. Oku)
Fastly

26 September 2025

Protocol for Transposed Transactions over HTTP
draft-kazuho-ptth-ptth-00

Abstract

This document specifies the Protocol for Transposed Transactions over HTTP (PTTH), an HTTP extension that allows a backend server to establish an HTTP connection to a reverse proxy and transpose HTTP request flow. The reverse proxy then forwards incoming requests to the backend server over the transposed connection. This extension lets backend servers behind restrictive firewalls accept HTTP traffic through reverse proxies without changing firewall settings and with virtually zero overhead.

Discussion Venues

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

Discussion of this document takes place on the Protocol for Transposed Transactions over HTTP Working Group mailing list (ptth@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/ptth/.

Source for this draft and an issue tracker can be found at https://github.com/kazuho/draft-unknown-ptth-ptth.

Status of This Memo

Table of Contents

1. Introduction

In scalable HTTP deployments— such as those using CDNs and dynamic backend server pools— clients send requests to reverse proxies, which then forward them to backend servers. Backend servers frequently reside behind firewalls that block inbound TCP or QUIC connections, requiring special network or firewall configuration to permit proxy-initiated traffic. To overcome these restrictions, some organizations use VPNs, but VPNs introduce operational complexity, hamper scalability, and impose performance overhead.

PTTH enables a backend server to establish an HTTP connection to a reverse proxy and transpose the flow of HTTP requests so that the proxy sends incoming requests back over the same connection. An HTTP request is used for authenticating the backend server and for negotiating the scope of requests forwarded to the transposed connection, providing flexibility to deployments.

Because PTTH transposes the direction of communication rather than encapsulating traffic, it incurs virtually zero overhead and delivers high efficiency.

TODO: expand

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.

3. The Protocol

To set up a transposed connection, the backend server connects to the reverse proxy and sends an HTTP request including a URI specifying the transposed endpoint and credentials that authenticate the backend server.

The exact form of the URI specifying the transposed endpoint is unspecified; it is up to each reverse proxy deployment.

Similarly, the authentication scheme is unspecified. Deployments can use either a TLS- or an HTTP-based authentication scheme, or something else.

The method being used to establish the transposed connection is different between HTTP versions. However, the HTTP header fields are version-independent, and therefore the parameters for negotiating PTTH can be defined in a version-neutral manner.

PTTH cannot originate over HTTP/2. To establish a transposed HTTP/2 channel, HTTP/1.1 upgrade is used, with the ALPN specifying HTTP/2.

3.1. HTTP/1.1 and HTTP/2

To establish a transposed HTTP/1.1 or HTTP/2 channel, the backend server connects to the reverse proxy using HTTP/1.1 ([HTTP1]), and uses the HTTP upgrade mechanism ([HTTP-SEMANTICS] Section 7.8) to negotiate the transposition.

The method of the upgrade request SHALL be "GET", accompanied by an "Upgrade: ptth" header field.

The request MUST also include the ALPN header field ([ALPN-HEADER]) specifying the HTTP versions that the backend server is willing to use on the transposed connection.

Once the transposed connection is established successfully, the reverse proxy responds with a 101 (Switching Protocols) response, alongside an ALPN response header specifying the HTTP version being chosen. After a 101 response is sent, HTTP requests are sent in the direction from the reverse proxy to the backend server.

Figure 1 shows an exchange of HTTP/1.1 upgrade request and response establishing the transposed connection. In this example, the Basic HTTP Authentication Scheme [BASIC-AUTH] is used to authenticate the backend server. As for the application protocol to be used on the transposed connection, the backend server is offering both HTTP/2 and HTTP/1.1, and the reverse proxy selects HTTP/2.

```
GET /reverse-endpoint HTTP/1.1
Host: example.com
Connection: upgrade
Upgrade: ptth
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
ALPN: h2, http%2F1.1

HTTP/1.1 101 Switching Protocols
Connection: upgrade
Upgrade: ptth
ALPN: h2
```

Figure 1: Establishing a transposed connection over HTTP/1.1

As the parameters for the transposed connection are exchanged using the upgrade request, they cannot be changed once the transposed connection is established. To change those parameters, a new HTTP/1.1 connection should be established and transposed.

3.2. HTTP/3

In HTTP/3 ([HTTP3]), the OPTIONS method (Section 9.3.7 of [HTTP-SEMANTICS]) is used to transpose HTTP request flow on the HTTP/3 connection. As the flow of the existing connection is transposed, neither the :protocol pseudo-header field nor the ALPN header field is used.

The target of the OPTIONS request is the endpoint that tranposes the connection; therefore, the asterisk ("*") request is never used for establishing PTTH.

Once the reverse proxy responds with a 2xx response, it starts forwarding HTTP requests on the server-initiated, bidirectional QUIC streams. Note that, due to packet reordering, backend servers might receive these requests before receiving a 200 response for the OPTIONS request.

Similarly to when HTTP/1.1 is used, establishment of a new HTTP/3 connection is required if a transposed HTTP/3 connection with different set of parameters is needed.

Once the connection is transposed, the reverse proxy MAY reset incoming requests that it receives using a H3_REQUEST_REJECTED error (Section 8.1 of [HTTP3]).

TODO: Discuss the downsides of transposing an HTTP/3 connection; notes:

* No issues with SETTINGS; none of the HTTP/3 settings are specific to clients or servers.

* No issues with QPACK; one set of QPACK streams can handle requests flying in both directions.

* We need to consider how to handle quarter stream IDs of HTTP/3 datagrams; but that issue not orthogonal to sending HTTP requests in both directions. The issue arises for any design that establishes the QUIC connection in the reverse direction. Maybe the answer here is to use stream_id / 4 + (2 << 60) as the quarter stream IDs for datagrams belonging to the transposed requests.

* Otherwise, the design does not interfere with WebTransport over HTTP/3; for both client- and server-initiated bidirectional streams, WebTransport streams can be identified by their signal vallues (0x41), and if they are associated to client- or server-initiated requests can be determined by their Session ID (i.e., the stream ID of the CONNECT stream).

* Rather than using OPTIONS, do we want to use an extended CONNECT? While the use of OPTIONS might be fine, HTTP requests without a special pseudo-header is end-to-end per definition. Using an extended CONNECT is a straightforward way to constrain the setup of a transposed _connection_ to hop-by-hop.

4. Establishing Authority

In HTTP, only the URI’s authority may process or delegate the request (Section 17.1 of [HTTP-SEMANTICS]).

This authority model of HTTP remains unchanged under PTTH:

* When the backend server connects to the reverse proxy and requests the transposition of the connection, the backend identifies the reverse proxy using a target URI whose authority component identifies the reverse proxy.

* When the reverse proxy forwards requests to the backend over a transposed connection, it is merely exercising its rights as the authoritative server. This behavior is identical to forwarding requests over connections the reverse proxy initiated, using whatever authentication scheme it chooses. PTTH differs only in how the backend connections are established.

5. Security Considerations

TODO

6. IANA Considerations

Once approved, this document will request IANA to register the following entry to the "HTTP Upgrade Tokens" registry maintained at https://www.iana.org/assignments/http-upgrade-tokens (https://www.iana.org/assignments/http-upgrade-tokens):

Value: ptth

Description: Establishes a transposed HTTP/1.1 connection.

Expected Version Tokens: None

Reference: this document

7. References

7.1. Normative References

[ALPN-HEADER]
Hutton, A., Uberti, J., and M. Thomson, "The ALPN HTTP Header Field", RFC 7639, DOI 10.17487/RFC7639, August 2015, <https://www.rfc-editor.org/rfc/rfc7639>.

[HTTP-SEMANTICS]
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>.

[HTTP1] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/rfc/rfc9112>.

[HTTP3] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/rfc/rfc9114>.

[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>.

7.2. Informative References

[BASIC-AUTH]
Reschke, J., "The ’Basic’ HTTP Authentication Scheme", RFC 7617, DOI 10.17487/RFC7617, September 2015, <https://www.rfc-editor.org/rfc/rfc7617>.

Acknowledgments

TODO.

Author’s Address

Kazuho Oku
Fastly
Email: kazuhooku@gmail.com

Additional contact information:

奥 一穂
Fastly