Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Introductory Talk for the IDP

C++-based MASQUE-Proxying for Lower OSI-Layer Protocol Traffic

Name:	Christoph **Rotte**
Advisor:	Lion Steger, Richard von Seck
Supervisor:	Prof. Dr.-Ing. Georg Carle
Begin:	12/2022
End:	08/2023

Topic and Background

Today, nearly all default proxying protocols like SOCKS, *Performance Enhancing Proxies* (PEPs), or HTTP CONNECT are based on reliable traffic enabled by TCP. One challenge that occurs as a result of using TCP for proxy applications, however, are multiplexed channels on concurrent streams. While HTTP/2 CONNECT, for example, supports creating several tunnels within a single connection, TCP logically reduces all tunnels to a single stream. This may lead to head-of-line blocking, in which independent streams slow each other down [1].

QUIC is a new multiplexed transport protocol based on UDP, and was introduced in 2021 by the IETF. In contrast to TCP, each stream's packets are independently secured and transmitted. This allows for efficient parallel data streams without the risk of head-of-line blocking. Since HTTP/3, which strictly uses QUIC instead of TCP, supports HTTP CONNECT, it is also possible to proxy TCP-based traffic via QUIC. [2]

In order to efficiently proxy QUIC-based traffic itself via QUIC, however, the protocol had to be developed further, which became the task of the recently formed *Multiplexed Application Substrate over QUIC Encryption* (MASQUE) Working Group. MASQUE introduced a new HTTP request, called CONNECT-UDP [3], that uses an extension for QUIC [4] to enable efficient proxying of UDP-based traffic. CONNECT-UDP can then be used to transmit encapsulated QUIC packets.
Because of this, MASQUE-based proxying helps to, for example, minimize the amount of data which can be accessed by the proxy. Consequently, the protocol easily enables nested MASQUE proxying with multiple proxy servers, which may be used to increase the users' privacy. [5]

MASQUE then extended CONNECT-UDP in the form of CONNECT-IP [6] which supports proxying of arbitriary IP-Layer traffic via HTTP. CONNECT-IP can therefore be used to communicate using proxied IP packets without having to, for example, differentiate between CONNECT for TCP or CONNECT-UDP for UDP.

In the scope of this Interdisciplinary Project, we are going to use CONNECT-IP to implement MASQUE proxying on the IP-Layer.

Related Work

The goal of this IDP is to extend a preexisting open source library/framework to support CONNECT-IP-proxied traffic as well as to subsequently analyze its behaviour and performance. Additionally, if possible, we are going to compare our implementation to other, external libraries that also support CONNECT-IP.

Up until now, however, only few implementations[1] exist for MASQUE-Proxying, like in Apple's closed source *Private Relay*[2] or Google's *quiche* library[3]. While the latter is one of the only open source projects to support CONNECT-UDP, it only recently started implementing CONNECT-IP[4].
Other QUIC and HTTP libraries, for example, facebook's *mvfst*[5] (QUIC library) and *proxygen*[6] (HTTP library), currently only have limited support for CONNECT-UDP and do not offer CONNECT-IP functionality at all.

The approach of this IDP is similiar to the paper "Evaluation of QUIC-based MASQUE Proxying" [7] which also implemented and evaluated MASQUE proxying in *aioquic*[7], a QUIC library for Python. For evaluation, it employed a proxied `client` $\leftrightarrow$ `server` structure which was then analyzed within a docker network with fixed parameters such as bandwidth limit and packet delay. The analysis confirmed the theoretical performance decline when transmitting data between client and proxy reliably. Additionally, the reliable transmission lead to higher round-trip delays and resource consumption of the proxy as sending reliably tends to hide congestion signals from the server. However, the implementation is solely based on the CONNECT-UDP protocol, which we want to further develop using CONNECT-IP.
As mentioned above, Google's quiche library currently seems to be the only open-source project which is starting to work on CONNECT-IP. The extension of an external library to use IP-Layer MASQUE Proxying, as well as its evaluation, therefore will not only benefit this library's current users but may also provide a second point of view on the CONNECT-IP protocol and MASQUE Proxying in general.

Research Questions

RQ 1 What are the implications of using CONNECT-IP in the context of overhead caused by encapsulation?

RQ 2 How does CONNECT-IP behave in the context of transmission performance compared to CONNECT-UDP?

RQ 3 What areas may cause library-specific differences and which challenges may occur because of this?

Approach

In order to implement the CONNECT-IP protocol, we are going to make use of the existing QUIC/HTTP functionality of mvfst and proxygen, respectively, for the following two reasons:

1. The two libraries (which are designed to work together), especially proxygen, seem to be much more popular than Google's quiche, when comparing the number of repository stars. Additionally, the code bases offer friendlier and well-documented interfaces.
2. By extending facebook's libraries, we can test and evaluate our implementation against Google's quiche, which may give us more insight in the overall behaviour of the CONNECT-IP protocol.[8]

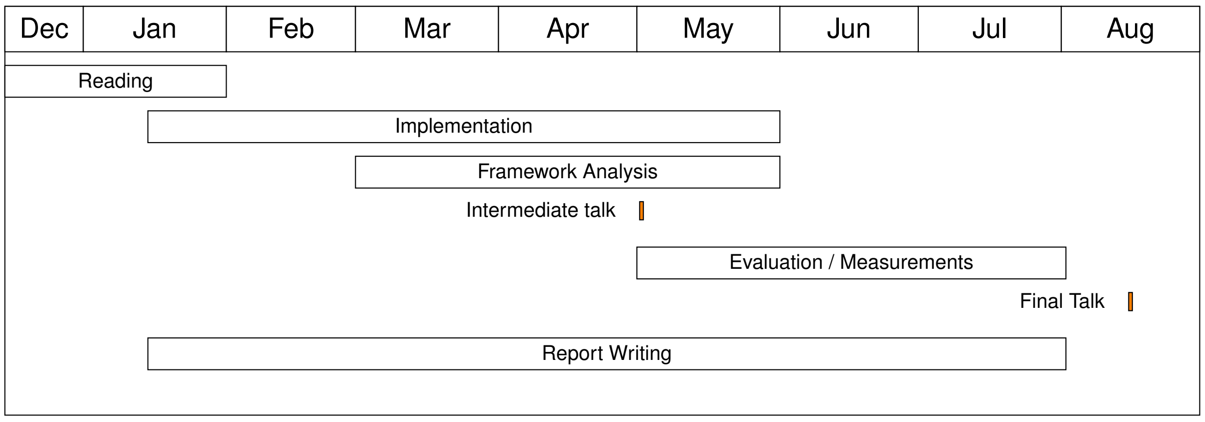
[1] `https://github.com/ietf-wg-masque/draft-ietf-masque-connect-udp/wiki/Implementations`
[2] `https://apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf`
[3] `https://github.com/google/quiche`
[4] `https://github.com/google/quiche/commit/425f03b`
[5] `https://github.com/facebookincubator/mvfst`
[6] `https://github.com/facebook/proxygen`
[7] `https://github.com/aiortc/aioquic`
[8] We believe that their implementation will be finished in time for our comparision.

While mvfst contains the functionality for the previously mentioned QUIC extension in the form of unreliable QUIC datagrams, proxygen supports communication on the HTTP/3 layer. In particular, we can use its preexisting implementation of HTTP/3 datagrams[9] to built CONNECT-IP upon.

For the evaluation, we plan on testing our implementation with a simulated typical use case, probably in the context of HTTP/3 applications. For this, we are going to implement a `client` $\leftrightarrow$ `proxy` $\leftrightarrow$ `server` structure similar to the docker-based approach with by Kühlewind et al. [7] and analyze the traffic according to the formulated research questions. Due to the similarities between CONNECT-UDP and CONNECT-IP, we assume that our outcomes may match certain results of the paper, for example, the observation that solely datagram-based connections scale better with the number of connections than stream-based connections in the context of performance and packet losses.

Since MASQUE proxying allows for nested tunneling, it should also be possible to further build a setup in which our IP traffic is routed through multiple proxies employing CONNECT-IP.
Additionally, we plan to analyze our implementation for interoperability with Google's quiche library by switching our client and/or proxy server with one built using quiche.

Planned Schedule



Interdisciplinary Context

The Department of Electrical and Computer Engineering offers the course *Applied Cryptology*[10] (EI04003), which teaches the functionality, analysis, and implementation of basic cryptographic procedures. Since QUIC usually employs TLS 1.3 to protect the contents of the individual packets[11], it depends on the secure application of several cryptographic algorithms. Furthermore, as proxies may create a bottleneck for the traffic, it is especially important to understand the inner workings of cryptography implementations to rule out unnecessarily insecure and/or inefficient data protection.

TLS 1.3 makes use of several symmetric and asymmetric encryption mechanisms to guarantee privacy, cryptographic hash functions for message authentication codes, and digital signatures to authenticate both client

[9] `https://www.rfc-editor.org/rfc/rfc9297.html`
[10] `https://www.ce.cit.tum.de/en/eisec/courses/?id=5250`
[11] `https://www.rfc-editor.org/rfc/rfc9001.html`

and server, which are all part of *Applied Cryptology*. In addition, the lecture examines different cryptographic work modes and explains when and how to correctly apply them in different scenarios.

References

[1] R. Marx, "Head-of-line blocking in quic and http/3: The details," Dec 2020, last accessed: 2022-11-30. [Online]. Available: https://calendar.perfplanet.com/2020/head-of-line-blocking-in-quic-and-http-3-the-details/

[2] L. Pardue and C. Wood, "A primer on proxies," Mar 2022, last accessed: 2022-11-29. [Online]. Available: https://blog.cloudflare.com/a-primer-on-proxies/

[3] D. S. and, "Proxying udp in http," Internet Requests for Comments, RFC Editor, RFC 9298, August 2022, https://www.rfc-editor.org/info/rfc9298. [Online]. Available: https://www.rfc-editor.org/info/rfc9298

[4] T. Pauly, E. Kinnear, and D. Schinazi, "An unreliable datagram extension to quic," Internet Requests for Comments, RFC Editor, RFC 9221, March 2022, https://www.rfc-editor.org/info/rfc9221. [Online]. Available: https://www.rfc-editor.org/info/rfc9221

[5] L. Pardue and C. Wood, "Unlocking quic's proxying potential with masque," Mar 2022, last accessed: 2022-11-29. [Online]. Available: https://blog.cloudflare.com/unlocking-quic-proxying-potential/

[6] T. Pauly, D. Schinazi, A. Chernyakhovsky, M. Kühlewind, and M. Westerlund, "IP Proxying Support for HTTP," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-masque-connect-ip-03, September 2022, https://www.ietf.org/archive/id/draft-ietf-masque-connect-ip-03.txt. [Online]. Available: https://www.ietf.org/archive/id/draft-ietf-masque-connect-ip-03.txt

[7] M. Kühlewind, M. Carlander-Reuterfelt, M. Ihlar, and M. Westerlund, "Evaluation of quic-based masque proxying," in *Proceedings of the 2021 Workshop on Evolution, Performance and Interoperability of QUIC*, ser. EPIQ '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 29–34. [Online]. Available: https://doi.org/10.1145/3488660.3493806