ICIC Express Letters
Volume **16**, Number **2**, February **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **169–176**

A SURVEY ON TRANSPORT LAYER IETF STANDARDIZATION WORKS FOR RESOLVING LIMITATIONS OF TCP

Pyung Soo Kim

Department of Electronic Engineering
Korea Polytechnic University
237, Sangidaehak-ro, Siheung-si, Gyeonggi-do 15073, Korea
pskim@kpu.ac.kr

Received June 2021; accepted August 2021

Abstract. *To address issues that were not considered when TCP was developed 40 years ago, the IETF, a world-renowned Internet standardization organization, put a lot of effort into it. In this paper, various standardization works are surveyed, classified, compared, and analyzed to provide Internet architects and engineers with insight and future prospects on the transport layer IETF standardization trend to improve the limitations of traditional TCP. Firstly, standardization works with small changes to TCP are described and then ossification problems of them are discussed. Secondly, standardization works with drastic changes using UDP to overcome TCP's inherent limitations and challenges of them are described. Finally, these two kinds of standardization works are compared and analyzed from a variety of views.*

Keywords: IETF standardization, Transport layer protocol, TCP, UDP, QUIC

1. **Introduction.** Internet, a remarkably complex system architecture, has revolutionized communications and methods of commerce by allowing various computer networks around the world to interconnect [1,2]. The Transmission Control Protocol (TCP) standardized by the Internet Engineer Task Force (IETF) in RFC (Request for Comments) 793 at 1981, is the core engine of the Internet and thus used by the vast majority of applications to transport their data reliably across the Internet. Due to its ubiquitous use in video streaming, web browsing, file transfers, communications, and other application types, TCP typically represents 85%∼90% of fixed access Internet traffic. In addition, the global mobile traffic is growing exponentially, and currently more than 90% of the mobile Internet traffic depends on TCP for reliable transmission [5,6]. However, for a number of reasons, it is well known that TCP performs quite poorly over unreliable wireless networks, while its dynamic TCP flow control is sensitive to congestion events and tends to underutilize the available network capacity. Far from a minor inconvenience, this reality costs network operators enormous sums due to inefficient use of expensive resources, poor subscriber quality of experience, and other factors.

To resolve limitations of the traditional TCP, there were several standardization works such as Stream Control Transmission Protocol (SCTP) [7,8], MultiPath TCP (MPTCP) [9,10], and TCP Fast Open (TFO) [11-13], that handle small changes to TCP, i.e., minor extensions to TCP algorithms and protocols. However, small changes to TCP suffer from difficulties in deploying them on the Internet due to protocol ossification problems. In addition, they may not overcome TCP's inherent limitations. Therefore, most recently, the transport QUIC with drastic changes using User Datagram Protocol (UDP) has been standardized by IETF [14-21]. Although its name was initially proposed as the acronym for "Quick UDP Internet Connections", IETF's use of the word QUIC is not an acronym. In addition, HTTP/3 will soon be standardized by IETF as the new version of HTTP

DOI: 10.24507/icicel.16.02.169

that runs on QUIC [18,19]. As QUIC and HTTP/3 standardization are complete, the WebTransport over UDP-based HTTP/3 protocol has been recently launched by IETF to address a replacement for WebSocket over TCP-based HTTP protocols [17,20]. Moreover, Multiplexed Application Substrate over QUIC Encryption (MASQUE) has been started to address challenges due to QUIC's mandatory encryption [17,21].

To provide Internet architects and engineers with insight and future prospects on the transport layer IETF standardization trend to improve the limitations of traditional TCP, this paper surveys, classifies, compares and analyzes various standardization works. Firstly, standardization works with small changes to TCP are described and then ossification problems of them are discussed. Secondly, standardization works with drastic changes using UDP and challenges of them are described. Finally, these standardization works are compared and analyzed from a variety of views such as working group, standard number, key characteristics, advantage, and limitation.

This paper is organized as follows. In Section 2, standardization works with small changes to TCP are surveyed. In Section 3, standardization works with drastic change using UDP are surveyed. In Section 4, comparative analysis from various perspectives is performed. Finally, conclusions are presented in Section 5.

2. Resolving Limitations with Small Changes to TCP.

2.1. Stream Control Transmission Protocol (SCTP).

HTTP/2, standardized by the IETF in RFC7540, is a major revision of the HTTP protocol. With HTTP/2, typical browsers do tens or hundreds of parallel transfers over a single TCP connection. If a single packet is lost in the network somewhere between two endpoints with HTTP/2, it means the entire TCP connection is down while the lost packet is re-transmitted and tries to find a way to their destination. That is, if one link is suddenly missing, everything that would come after the lost link needs to wait. It becomes a TCP-based Head of Line Block (HOLB). As the packet loss rate increases, HTTP/2 performs less and less well. Stream Control Transmission Protocol (SCTP) was adopted to resolve the HOLB problem in HTTP/2. SCTP is a transport layer protocol standardized by the Transport Area Working Group (TSV WG) of IETF in RFC4960 with several of the desired characteristics [7,8]. SCTP ensures reliable, in-sequence transport of data. SCTP provides multihoming to support multiple IP paths to its peer endpoint. This enables transparent failover between redundant network paths.

2.2. MultiPath TCP (MPTCP).

When a TCP connection is established, the connection is bound to the IP addresses of the two communicating endpoints. For whatever reason, the connection will fail if one of these addresses changes. Today's networks are multipath, that is, mobile devices have multiple heterogeneous wireless interfaces, datacenters have many redundant paths between servers, and multihoming has become the norm for big server farms. In fact, a TCP connection over more than one path within the network can cause packet reordering and thus TCP misinterprets this reordering as congestion and slows down. This mismatch between today's multipath networks and TCP's single-path design creates tangible problems. MultiPath TCP (MPTCP) is a modification to TCP that allows multiple paths to be used simultaneously over multiple interfaces by a single transport connection [9,10]. MPTCP is a transport layer protocol standardized by the MPTCP WG of IETF in RFC6824, which was later replaced by RFC8684. MPTCP enables endpoints to send the data corresponding to any TCP connection over different paths. With MPTCP, a mobile device can simultaneously and efficiently use both its cellular and Wi-Fi interfaces.

2.3. **TCP Fast Open (TFO).** A traditional TCP handshake is a three-step process and thus called the TCP 3-way handshake. Once this process is complete, the sender and receiver can both start exchanging data. However, performing these three steps in turn increases network latency time which therefore decreases overall page load speed. TCP Fast Open (TFO) standardized by the TCP Maintenance and Minor Extensions (TCPM) WG of IETF in RFC7413 is an optional algorithm within TCP that lets endpoints that have established a full TCP connection in the past eliminate a round-trip of the handshake and send data right away [11-13]. That is, TFO is an extension to speed up the opening of successive TCP connections between two endpoints. This speeds things up for endpoints that are going to keep talking to each other in the future and is especially beneficial on high-latency networks where Time to First Byte (TTFB) is critical. TTFB refers to the time between the browser requesting a page and when it receives the first byte of information from the server. TFO specification describes how applications can pass data to the server to be delivered already in the first TCP SYN packet. Thus, the biggest reason to use TFO is to get that first chunk of data faster.

2.4. **Ossification problems.** New transport layer protocols with small changes to TCP have suffered from protocol ossification problems. Many middle-boxes such as firewalls, NATs, routers that only allow TCP or UDP between a client and the remote server will spot unknown new TCP options and block such connections since they do not know what the options are. If allowed to detect protocol details, systems learn how protocols typically behave and over time it becomes impossible to change them. That is, introducing another transport layer protocol makes some of the connections fail because they are being blocked by middle-boxes that see it not being UDP or TCP and thus evil or wrong somehow. Additionally, changing things in the transport protocol layer of the network stack typically means protocols implemented by operating system kernels. Updating and deploying new operating system kernels is a slow process that requires significant effort. The only truly effective way to resolve ossification problem is to encrypt as much of the communication as possible to prevent middle-boxes from seeing much of the protocol passing through.

3. Resolving Limitations with Drastic Changes Using UDP.

3.1. **Google's QUIC.** QUIC is a transport layer protocol designed with the intention of reducing connection and transport latency as well as providing bandwidth estimation in each direction to avoid congestion [14-16]. The initial QUIC protocol was designed by Google and initially implemented in 2012, announced publicly to the world in 2013 when Google's experimentation broadened. QUIC runs over UDP rather than TCP. UDP is much faster than TCP but is generally less reliable as it does not have the same error checking and loss prevention as TCP does. UDP is commonly used in applications that do not require packets to be in the exact right order, but care about latency. Google implemented the protocol and subsequently deployed it both in their widely used Chrome browser and in their widely used server-side services such as Google search, Gmail, and YouTube. They iterated protocol versions fairly quickly and over time they proved the concept to work reliably for a vast portion of users. The initial QUIC implementation garnered attention from the web performance community when Google shared performance results indicating that QUIC reduced latency of desktop Google Search responses by 8.0% and rebuffering rates of YouTube playbacks by 18.0%. In 2017, numbers quoted by QUIC engineers at Google mentioned that around 7% of all Internet traffic were already using this protocol.

3.2. **Transport QUIC.** The first Internet draft for QUIC was sent to the IETF for standardization in 2015, but it took until late 2016 for a QUIC WG to get approved and started [17]. However, then it took off immediately with a high degree of interest from many parties. The QUIC WG quickly decided that the QUIC protocol should be able to transfer other protocols compared with only HTTP. Google-QUIC only ever transported HTTP. It was also stated that IETF version QUIC, called the IETF-QUIC, should base its encryption and security on TLS 1.3 instead of the custom approach used by Google version QUIC, called the Google-QUIC. The IETF-QUIC protocol architecture was split in two separate layers: the transport QUIC and the HTTP over QUIC layer. While the work on IETF-QUIC has progressed, the Google team has incorporated details from the IETF version and has started to slowly progress their version of the protocol towards what the IETF version might become. Google has continued using their version of QUIC in their browser and services. Most new implementations under development have decided to focus on the IETF version and are not compatible with the Google version. The IETF just officially formalized and published transport QUIC as RFC9000, supported by RFC9001, RFC9002, and RFC8999 in May 2021. This news is a big deal, both for the IETF and for the Internet ecosystem. Transport QUIC has been one of the IETF's most high-profile activities in recent years. Starting as an experiment at Google, transport QUIC was developed through a collaborative and iterative standardization process at the IETF. Transport QUIC has finally become a new latency-reducing, reliable, and secure Internet transport protocol that is slated to replace TCP, the most commonly used transport today.

3.3. **HTTP/3.** HTTP/3, the version of HTTP that runs on transport QUIC, is following closely behind and should be published soon as the IETF standard [18,19]. The performance of HTTP is an important factor when it comes to loading web pages quickly and efficiently. HTTP is a well-established protocol that has several versions, with each adding features that improve performance over the older one. HTTP/1.1 and HTTP/2 are widely deployed on the Internet today and rely on TCP and optionally TLS. HTTP/3 is in the final stages of standardization in the IETF QUIC WG. The original proposal was named 'HTTP/2 Semantics Using the QUIC Transport Protocol', and later named 'HTTP over QUIC'. And then, HTTP-over-QUIC was renamed finally HTTP/3. Much like HTTP/2 was once introduced to transport HTTP over the wire in a completely new way, HTTP/3 is yet again introducing a new way to send HTTP over the network. The switch to QUIC aims to fix HOL problem of HTTP/2. Because transport QUIC provides native multiplexing, lost packets only impact the streams where data has been lost. The practical effect of the upgrade to HTTP/3 is to reduce the latency of poor or lossy Internet connections. Standardization will be finalized soon, but HTTP/3 protocol is still officially an Internet Draft. Nevertheless, HTTP/3 is already supported by about 20% of the top 10 million websites according to W3Techs.

3.4. **Transport QUIC and HTTP/3 interoperability.** As transport QUIC and HTTP/3 evolve through a collaborative and iterative standardization process at the IETF, a particularly significant moment is encountered. It has been five years since HTTP/2 was published, and four decades since the completion of TCP, the underlying transport protocol that QUIC seeks to replace. Changing an Internet protocol, especially a transport protocol designed to replace TCP, requires all the communicating entities to be able to speak to each other without any issues. The Internet is fundamentally a multi-vendor ecosystem, and as a result, communication almost always involves multiple implementations of the same protocol. To be successfully deployed, various vendors need to build QUIC implementations, and these implementations need to interoperate with each other. Vendors, including Apple, Google, Microsoft, Mozilla, and Fastly, have been working hard on their own implementations, many of which are now quite mature. These implementers gather periodically to test their implementations against each other, and most of them also participate in a continuously running automated interoperability testing tool called the QUIC Interop Runner. The Interop Runner shows the current state of transport QUIC and HTTP/3 interoperability between participating implementations, on a suite of correctness and performance tests. The community of implementers working on these protocols has learned that having open and continuous communication with each other is essential for implementing and deploying these protocols. These implementers have been in close touch with each other over the past years as the protocol has evolved and, excitingly, most implementations are close to being fully interoperable with each other.

3.5. **WebTransport.** WebSocket standardized by the IETF in RFC6455 is a computer network protocol for client-server communication, providing full-duplex communication over a single TCP connection. WebSocket is TCP-based, thus having all of the drawbacks of TCP that make it a poor fit for latency sensitive applications. As transport QUIC and HTTP/3 standardization are complete, the WEBTRANS WG was recently launched by the IETF to develop WebTransport framework for a replacement for WebSocket [17,20]. WebTransport framework uses the UDP-based HTTP/3 protocol as a bidirectional transport and enables clients constrained by the Web security model to communicate with a remote server using a secure multiplexed transport. Using WebTransport resolves HOL blocking which can be an issue with WebSocket. Additionally, there are performance benefits when establishing new connections, as the underlying QUIC handshake is faster than starting up TCP over TLS.

3.6. **Challenges due to encryption and UDP.** The deployment of encrypted protocols on the Internet is moving rapidly. This is good news as encryption is important to secure Internet traffic and protect user privacy. Unfortunately, this rapid deployment takes away some of the capabilities which proved effective in improving quality of experience for users for example, network assisted rapid loss recovery, and domain specific congestion control. Encryption of the transport protocol can block operators' visibility, which means that operators are losing the awareness of traffic traversing through their networks while still being kept accountable for traffic optimization, network management, policy enforcement, as well as regulatory rules of governments and demands by society. Therefore, transport QUIC's mandatory encryption presents challenges for specialized use cases where end-to-end connectivity is not possible, not feasible, or not wanted.

As mentioned before, the most important key characteristic of the transport QUIC is to run on top of UDP rather than TCP. Transport QUIC may also struggle to deploy due to protocol ossification issue. It is known that many operators, enterprises, and organizations block or rate-limit UDP traffic outside of their DNS port, as they have been most recently exploited as attacks. Particularly, some of the existing UDP protocols and popular server implementations for them have been vulnerable for amplification attacks where one attacker can make a huge amount of outgoing traffic to target innocent victims. QUIC tries to mitigate amplification attacks by requiring that the initial packet must be at least 1200 bytes and that a server must not send more than three times the size of the request in response.

3.7. **Multiplexed Application Substrate over QUIC Encryption (MASQUE).** To address challenges due to QUIC's mandatory encryption, the Multiplexed Application Substrate over QUIC Encryption (MASQUE) WG was recently launched by the IETF [17,21]. It is known that transport protocol proxying enables endpoints to communicate when end-to-end connectivity is not possible, or to apply additional encryption where desirable such as a Virtual Private Network (VPN). In addition, proxying can also improve client privacy by hiding a client's IP address from a target server. MASQUE includes new proxying features based on the end-to-end encrypted QUIC transport protocol to identify, enhance, and manage encrypted traffic using a collaborative and therefore even more powerful approach. MASQUE proposes the use of QUIC as a substrate protocol to open a tunnel to network proxy nodes. Such a proxy node, or MASQUE server, can offer various services like QUIC proxy, UDP proxy or IP-forwarding. In addition, the QUIC-based tunneling also enables secure communication between an endpoint and the proxy. This is an opportunity to offer additional services like faster loss recovery by the proxy, exposure of up-to-date network information that can help to assist congestion control, or even in-network bandwidth aggregation of multiple access links.

4. **Comparative Analysis from Various Perspectives.** Figure 1 shows the historic view of various transport layer IETF standardization works such as SCTP, MPTCP, TFO, Transport QUIC, HTTP/3, WebTransport, MASQUE according to TCP-based and UDP-based protocols. In Table 1, these standardization works are compared and analyzed from a variety of views such as working group, standard number, key characteristics, advantage, and limitation.

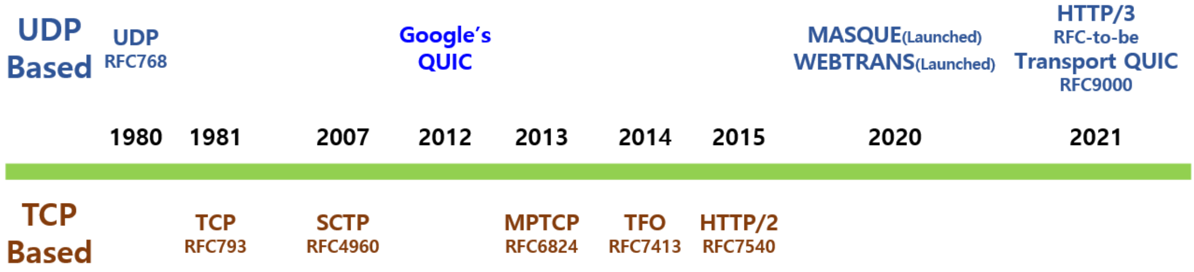

FIGURE 1. Historic view of transport layer IETF standardization works

This comparative survey research can provide Internet architects and engineers with insight and future prospects on the transport layer IETF standardization trend to improve the limitations of traditional TCP. In addition, most recently, the IETF standards focus is shifting to both supporting QUIC's further deployment in different network architectures, and extending it to support other applications in various WGs such as Transport Area WG, Transport Services WG Delay/Disruption Tolerant Networking WG. Therefore, this comparative survey research will serve as an excellent reference for future progress of these various WGs.

5. **Conclusions.** This paper has surveyed, classified, compared and analyzed various standardization works, which might provide Internet architects and engineers with insight and future prospects on the transport layer IETF standardization trend to improve the limitations of traditional TCP. Standardization works, such as SCTP, MPTCP, TFO, with small changes to TCP have been described and then ossification problems of them are discussed. Then, standardization works, QUIC, HTTP/3, WebTransport, MASQUE, with drastic changes using UDP to overcome TCP's inherent limitations and challenges of them have been described. Finally, comparative analysis from various perspectives has been performed.

With the new standardization of the UDP based transport layer protocol, the focus of IETF standardization works is shifting to support its further deployment in other network architectures and to extend it to support other applications. In addition, the research work going on in the transport area today will touch every corner of the Internet, and new ideas are arriving all the time. Therefore, as the related standardization and research field is expanding, it is expected to give great opportunities to Internet architects and engineers.

TABLE 1. Comparative analysis from various perspectives

	Protocol	**WG**	**Standard**	**Key Characteristics**	**Advantages**	**Limitations/ Challenges**
Small changes to TCP	SCTP	TSV	RFC 4960	A reliable, in-sequence transport protocol with integrating components of TCP/UDP	· Resolving TCP's HOL blocking · Providing some degree of fault tolerance with multi-homing · Protecting against SYN flooding attacks	· Ossification · Allowing limited IP addresses · Supporting only static IP NAT · Not supporting dynamic policy
	MPTCP	MPTCP	RFC 6824	Multiple paths to be used simultaneously for a single TCP connection	· Better resource utilization and throughput · Smoother reaction to failures	· Ossification · Security concerns · Visibility issue due to multipath transmission
	TFO	TCPM	RFC 7413	An extension to speed up the opening of successive TCP connections and reduce TTFB	· Improving TCP three-way handshake · Skipping a round-trip delay · Lowering the latency in the start of data transmission	· Ossification · Privacy concerns
Drastic changes using UDP	QUIC	QUIC	RFC 9000	A latency-reducing, reliable, multiplexed, and always-encrypted transport protocol	· Improving performance and reliability · Resolving TCP's HOL blocking · Already tested and deployed	· Visibility issue due to entire transport encryption · Ossification possibility due to blocking or rate-limiting UDP traffics outside of DNS port 53 in many operators, enterprises, and organizations
	HTTP/3	QUIC	RFC-to-be	A mapping of HTTP semantics on top of QUIC	· Resolving HOL blocking of HTTP/2 · Better transmission speed · Shorter loading times · More stable connection	
	WebTransport	WEBTRANS	Recently launched	A replacement for TCP-based WebSocket	· HTTP/3 based bi-directional, persistent, low-latency, extensible web communications	
	MASQUE	MASQUE	Recently launched	A flexible proxying built into a standard webserver with HTTP/3	· Resolving visibility issue and ossification possibility of QUIC	–

REFERENCES

[1] R. Kahn and M. A. Dennis, *Internet*, Encyclopedia Britannica, 2021.

[2] Y. Wang, X. Chen and X. Ye, Prediction of network protocol data flow based on a recurrent neural network, *International Journal of Innovative Computing, Information and Control*, vol.15, no.4, pp.1381-1395, 2019.

[3] J. Luo, J. Jin and F. Shan, Standardization of low-latency TCP with explicit congestion notification: A survey, *IEEE Internet Computing*, vol.21, no.1, pp.48-55, 2017.

[4] B. H. Kim and D. Calin, On the split-TCP performance over real 4G LTE and 3G wireless networks, *IEEE Communications Magazine*, vol.55, no.4, pp.124-131, 2017.

[5] P. Kim, An analysis of Google's research and development works for improving Internet service, *The Journal of Korean Institute of Communications and Information Sciences*, vol.44, no.2, pp.288-298, 2019.

[6] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella and M. Zorzi, A survey on recent advances in transport layer protocols, *IEEE Communications Surveys & Tutorials*, vol.21, no.4, pp.3584-3608, 2019.

[7] P. Natarajan, F. Baker, P. D. Amer and J. T. Leighton, SCTP: What, why, and how, *IEEE Internet Computing*, vol.13, no.5, pp.81-85, 2009.

[8] S. Ahmad and M. J. Arshad, Enhancing fast TCP's performance using single TCP connection for parallel traffic flows to prevent head-of-line blocking, *IEEE Access*, vol.7, pp.148152-148162, 2019.

[9] Q. Peng, A. Walid, J. Hwang and S. H. Low, Multipath TCP: Analysis, design, and implementation, *IEEE/ACM Transactions on Networking*, vol.24, no.1, pp.596-609, 2016.

[10] L. Chao, C. Wu, T. Yoshinaga, W. Bao and Y. Ji, A brief review of multipath TCP for vehicular networks, *Sensors*, vol.21, no.8, 2021.

[11] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain and B. Raghava, TCP fast open, *Proc. of the 7th ACM Conference on Emerging Networking EXperiments and Technologies (CoNEXT'11)*, pp.1-12, 2011.

[12] E. Sy, T. Mueller, C. Burkert, H. Federrath and M. Fischer, Enhanced performance and privacy for TLS over TCP fast open, *Proceedings on Privacy Enhancing Technologies Symposium*, vol.2020, pp.271-287, 2020.

[13] S. Chen, S. Jero, M. Jagielski, A. Boldyreva and C. Nita-Rotaru, Secure communication channel establishment: TLS 1.3 (over TCP fast open) versus QUIC, *Journal of Cryptology*, vol.34, no.3, p.26, 2021.

[14] Y. Cui, T. Li, C. Liu, X. Wang and M. Kühlewind, Innovating transport with QUIC: Design approaches and research challenges, *IEEE Internet Computing*, vol.21, no.2, pp.72-76, 2017.

[15] A. Langley et al., The QUIC transport protocol: Design and Internet-scale deployment, *Proc. of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'17)*, pp.183-196, 2017.

[16] A. Yu and T. A. Benson, Dissecting performance of production QUIC, *Proc. of the Web Conference 2021 (WWW'21)*, pp.1157-1168, 2021.

[17] M. Kosek, T. Shreedhar and V. Bajpai, Beyond QUIC v1: A first look at recent transport layer IETF standardization efforts, *IEEE Communications Magazine*, vol.59, no.4, pp.24-29, 2021.

[18] R. Marx, J. Herbots, W. Lamotte and P. Quax, Same standards, different decisions: A study of QUIC and HTTP/3 implementation diversity, *Proc. of the ACM SIGCOMM Workshop on the Evolution, Performance, and Interoperability of QUIC (EPIQ'20)*, pp.14-20, 2020.

[19] I. V. Harish Kumar, *Performance Evaluation of OpenStack with HTTP/3*, Thesis, Blekinge Institute of Technology, 2021.

[20] J. Posnick, *Experimenting with WebTransport*, web.dev, 2021.

[21] A. Z. Sarker, M. Westerlund, M. Ihlar, M. A. Torre and M. Kuehlewind, *A Collaborative Approach to Encrypted Traffic*, Ericsson, 2020.