Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

ARTICLE TYPE

Towards a Data-Driven Framework for Optimising Security-Efficiency Tradeoff in QUIC

Amith Murthy[1] | Muhammad Rizwan Asghar[1] | Wanqing Tu[2]

[1]School of Computer Science
The University of Auckland
Auckland, New Zealand
[2]Department of Computer Science
Durham University
Durham, UK

Correspondence
Muhammad Rizwan Asghar
Email: r.asghar@auckland.ac.nz

Abstract

Advances in computing and compression technology, coupled with high-speed networks, has beaconed an era of video streaming on the Internet. This has led to a need to enhance the security of communications transporting data without degrading its performance. The Transport Layer Security (TLS) protocol negotiates configurations for securing communication channels. Such conversations adversely impact latency, thereby presenting a fundamental tradeoff between security and efficiency. In this work, we present a conceptual framework, called SEC-QUIC (Secure and Efficient Configurations for QUIC), that focuses on optimising this tradeoff specifically for video transmissions by investigating various factors in Quick UDP Internet Connections (QUIC). Transport-layer-related elements, such as Maximum Transmission Unit (MTU) sizes, cipher suites, and ACK timer, are examined to evaluate the impact on the security-efficiency tradeoff in QUIC-based video transmissions using platform-based experiments. Subsequently, we develop a conceptual framework to leverage QUIC's dynamics based on the context of a connection to optimise the security-efficiency tradeoff. Our findings demonstrate the need to alter default configurations based on the contextual factors of a connection (e.g., resource constraints and network conditions) in QUIC-based video transmissions to balance the tradeoff. Experiments reveal an MTU of 1400 bytes is found to have 60% better throughput compared to an MTU of 1200 bytes while also 4% less CPU usage on average for the transmission of 100 MB video files. Overall, our experiments suggest that fine-tuning performance and security related configurations is an effective approach to optimising the security-efficiency tradeoff in video transmissions.

KEYWORDS:
QUIC, Security, Video streaming, Efficiency

1 | INTRODUCTION

Video is increasingly the dominant format through which media is consumed on the Internet and currently accounts to more than 60% of the total downstream volume of traffic globally, as well as being a major category in upstream traffic[1]. Meanwhile, there is a greater need and awareness of security surrounding the Internet as cybercrime becomes an increasing threat[2]. To ensure security, more than 50% of Internet traffic is encrypted to provide a certain degree of privacy protection for user data, with the

increasing adoption of TLS 1.3[1]. Furthermore, governments are beginning to impose stricter regulations, such as General Data Protection Regulation (GDPR)[3], to set guidelines on different security mechanisms.

Lampson[4] stresses that despite security being important to users, users are neither willing to sacrifice functionality nor to experience inconvenience to improve it. This is important as the transport layer plays a critical role in the layered Internet architecture by providing communication services directly to the application processes running on end points[5]. However, security protocols at the transport layer, such as TLS, incur an overhead when establishing connections and exchanging data, resulting in performance degradation[6,7]. This process can be made more efficient by improving Transmission Control Protocol (TCP) but doing so has reached a point of diminishing returns as upgrading TCP on key endpoints is an arduous and lengthy process[7,6]. This is primarily because TCP is implemented in the kernel space, which requires Operating System updates to push updates to the TCP stack. This coupled with middleboxes serving as key control points on the Internet has made it challenging to modify TCP[7]. The handshake latency incurred by TCP and TLS before any data transmission can start introduce handshake round trips. To partially alleviate these challenges, Google developed Quick UDP Internet Connections (QUIC), a new transport protocol that offers a solution and is increasingly being adopted as its traffic is estimated to be 7% of Internet traffic[8]. QUIC operating in the user space has facilitated its rapid deployment in various Google applications[8]. Notably, it is able to vastly improve Round Trip Time (RTT) overhead by exchanging cryptographic configurations during the handshake that results in lower RTTs to reduce overheads of secure connection establishment.

Typically, there is an inherent cost to security controls. In the case of data communications, particularly in video streaming applications, this cost primarily impacts efficiency by increasing the latency of operations in the transport of data objects. This is known as the security-efficiency tradeoff, and it is affected by many factors: network conditions, users' resource constraints, application requirements, protocol operations, and user preferences. Due to the wide variety of devices using in the Internet, the resource constraints of the devices vary greatly. Thus, the general security measures employed at different devices will have varying costs. This can lead to the use of security mechanisms that add unnecessary overheads to resource-constrained devices. Hence, a lightweight solution would be more appropriate. Furthermore, QUIC mainly struggles to provide performance gains over TCP in high-bandwidth, low-delay, low-loss networks[6]. Therefore, optimising the tradeoff in these environments can be hugely beneficial. However, due to the dynamic nature of networks and the heterogeneity of network devices, the tradeoff impacts caused by default configurations varies across the devices, depending on their resource constraints. Therefore, a data-driven framework that considers the context of a connection to optimise the tradeoff can yield performance gains by reducing the cost of security by selecting less resource-intensive cipher suite for packet payload encryption, the size of packets and sensitivity to loss detection. The context is considered to be the factors that most impact the security-efficiency tradeoff, such as volume and type of data being transmitted, the resource constraints of user devices, and the network conditions.

While there is previous literature on the performance analysis of QUIC's video streaming[9,10,11], several studies focus on developing a framework to intelligently configure parameters so as to improve video streaming Quality of Experience (QoE) while not taking the cost of security that adversely impacts efficiency in data transmissions into account. QUIC's integration of the TLS and TCP handshake process, and its implementation of congestion control at the application layer, atop UDP transport enables application-specific optimisations for security and data transmission configurations. Previous studies do not examine how the computational overhead of the configurations (i.e., MTU, cipher suites, and the ACK timer) may impact the tradeoff on a packet-level basis[12,9,11]. Our study investigates how the fine-tuning of these factors can optimise the overhead in the production of each packet and the overall transmission of large multimedia files. Thus, MTU is examined for its impact on total packet volume, cipher suites to evaluate the impact of their complexity on packet encryption, and the ACK timer for the optimisation of packet loss detection. In managing packet-level overhead by optimally configuring the factors that determine the overhead, the overall security-efficiency tradeoff can be balanced by considering the contextual factors of a connection to improve video streaming applications. Thus, this study is able to quantify the security and efficiency tradeoff affected by the studied metrics, and evaluate the impact of cipher suites, and transport layer dynamics in the transmission of video files by examining the per-packet computational overhead of various configurations in different use cases. The default configurations of maximum MTU size, loss detection, and cipher suites are set with a general-purpose view, intended to operate adequately in a diverse range of scenarios. However, given the need for greater efficiency in video streaming use cases, the default settings of such parameters can introduce unnecessary overhead by performing operations (e.g., encryption and loss detection) on more packets than the optimal setting. Thus, in considering the contextual factors of a given connection between the server and the optimal setting at which these parameters operate is leveraged to conceptualise a novel framework, called Secure and Efficient Configurations for QUIC (SEC-QUIC), to dynamically configure the parameters such that the tradeoff can be optimised on a per-connection basis. Therefore, this article aims to address the following research questions:

- How do transport layer dynamics (e.g., MTU, cipher suites, and ACK timers) in QUIC impact the security-efficiency tradeoff in video transmissions?
- How can these transport layer dynamics be leveraged to optimise the varying level of security and performance overheads, experienced by heterogeneous devices as a result of custom configurations?

The questions posed are studied via several experiments that investigate QUICs performance with different configurations for MTUs, cipher suites, and ACK timers in transporting several video objects under different network conditions. This establishes a knowledge base for the proposal of a conceptual data-driven framework. Our main research contributions include:

- Investigate findings based on the experiment platform to evaluate how QUIC performance is impacted by transport layer dynamics.
- A novel framework called SEC-QUIC is proposed to improve the security-efficiency tradeoff for a connection by identifying optimal configurations based on experiment statistics collected.

The rest of this article is organised as follows. Section 2 reviews related work. Section 3 presents SEC-QUIC, explains the platform for SEC-QUIC, and describes data collection for our study. Section 4 reports our results and the factors examined. Finally, Section 5 concludes this work and provides research directions for future work.

2 | BACKGROUND AND RELATED WORK

QUIC is a protocol designed by Google as a user-space transport that utilises User Datagram Protocol (UDP) in the transport layer and deploys congestion control, loss detection and security protocols in the application layer. Primarily developed to enable rapid deployment and continued evolution of the transport mechanisms, it replaces much of the traditional Hypertext Transfer Protocol Secure (HTTPS) stack: HTTP/2, TLS, and TCP. The use of UDP allows QUIC to traverse middleboxes, but still allows it to add its features on top. The features allow it to provide similar reliability and congestion control to that of TCP while removing many of its drawbacks, such as protocol entrenchment, implementation entrenchment, handshake delay, and head-of-line blocking delay. A unique feature of QUIC is its use of encrypted packets during the initial negotiation between client and server. This feature enables QUIC to minimise handshake latency for most connections by removing unnecessary handshake overhead.

2.1 | Security

Mathieu et al.[12] conduct a comparative security analysis of QUIC and TLS to evaluate their impacts on HTTP/2 based services. The work evaluates a number of vulnerabilities in both protocols and assesses their impact on web-based services. The authors argue that the TLS/TCP stack is open to a greater number of attacks, the exploitation of which can have more detrimental impacts on the server than the QUIC/UDP stack. While the authors argue that the QUIC is able to deliver content more reliably, the work does not focus on QUIC's performance.

Lychev et al.[13] examine how attacks on QUIC packets during the handshake can introduce latency overheads, countering the primary 0-RTT connection goal of the protocol. Two types of attacks are evaluated in detail: replay and packet manipulation attacks. Lychev et al. conclude that these attacks stem from the very parameters responsible for reducing latency in QUIC, such as the source address token that allows for 0-RTT connections. The authors find that parameters can be exploited to disrupt server operations if an adversary has access to the communication channel, thus SEC-QUIC aims to address this by automating configuration of parameters per connection.

2.2 | Video Streaming

Seufert et al.[9] perform a Quality of Experience (QoE) investigation on whether the benefits of QUIC are noticeable to an end user in the application of video streaming. The impact on QoE factors was evaluated to observe the difference between QUIC and TCP in streaming YouTube videos on different types of access networks. They find that contrary to its design goals, streaming over QUIC led to higher initial delays but the differences had a negligible impact on QoE.

Nathan et al.[10] develop a transport protocol for multi-user video streaming on top of QUIC. They argue that a QUIC based server makes it easier to test rate control changes and allows for updating congestion control specific parameters. Although Nathan et al. show that their solution can vastly improve QoE fairness for video streaming in QUIC using optimisations to congestion control algorithms, the study is primarily focused on improving fairness for multiple client endpoints sharing a bottleneck link. Meanwhile, SEC-QUIC focuses on reducing both security and performance overheads at the server and client endpoints.

Bhat et al.[11] investigate the impact of using QUIC over Dynamic Adaptive Streaming over HTTP (DASH) and provide a comparative analysis with DASH over TCP. The quality adaptive DASH algorithms are designed for DASH over TCP and the work investigates how these can be modified to take advantage of the benefits provided by QUIC. Notably, the study argues that the lack of head-of-line blocking in QUIC can be leveraged to provide performance gains for DASH. Congestion control cannot be varied according to the needs of the applications. However, in QUIC, the congestion control can be disabled, and DASH algorithms can implement their own while continuing to use the benefits of packaging and loss recovery provided by QUIC. While this study demonstrates QUIC's performance in video streaming applications, it does not leverage QUIC configurations to optimise its performance.

As Nathan et al.[10] assert, QUIC enables easy updates to congestion control parameters. In our work, we aim to leverage this in SEC-QUIC to manage security and efficiency specific parameters to optimise the tradeoff for video data. Unlike the aforementioned studies, SEC-QUIC focuses on the efficiency of transmission of video data from server to client without compromising security. Moreover, the parameters investigated in this study differ from those in previous studies[9,11].

3 | OUR SOLUTION

3.1 | SEC-QUIC: The Proposed Framework

We propose a framework, named SEC-QUIC, that aims at optimising the tradeoff between security and efficiency based on the context of the connection for video transmissions using QUIC. The context is defined by analysing a number of factors including the nature of the application, resource constraints between the end points, network conditions, and other data-driven parameters. Figure 1 illustrates SEC-QUIC. Essentially, SEC-QUIC takes parameters as input and produces optimal configurations for QUIC-based video transmissions. To do so, SEC-QUIC has a repository of configuration templates for standard use case scenarios. Based on the scenario, the decision maker module in SEC-QUIC fetches a configuration template from the repository and load it in order to make a decision. For making such decisions, it dynamically considers additional inputs, if needed. The outcome of SEC-QUIC is a set of configuration parameters.

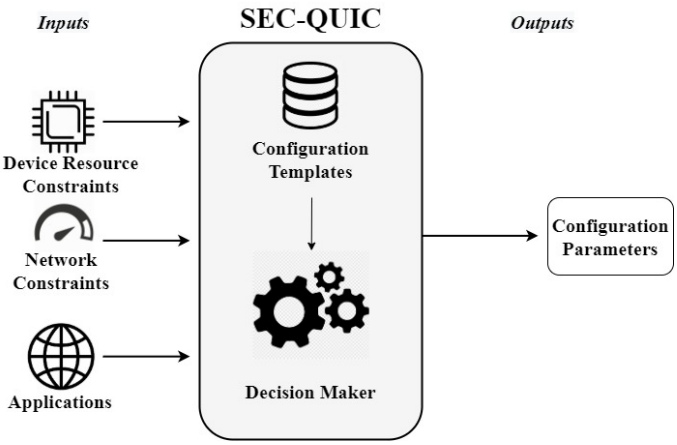


FIGURE 1 The SEC-QUIC framework.

SEC-QUIC is intended for optimising the tradeoff in scenarios by dynamically configuring the parameters to reduce the cost of security by decreasing computational cost as well. Such scenarios where security has a higher burden on efficiency is typically

encountered in clients with constrained resources where the cost of encryption and decryption of traffic adversely impacts the efficiency of transmission. Thus, we measure the cost of security in the transmission of both small and large video objects and investigate parameters that impact efficiency. These parameters include:

- MTU: determines the size of the protocol data unit that will be transmitted over the Internet. By minimising the volume of traffic between end points, we can increase the efficiency of the protocol. Unfortunately, there is an inherent risk in increasing MTU size as large payloads can congest the network. However, in high bandwidth networks, it can be leveraged to improve efficiency particularly in the transfer of large objects.

- Cipher suites: have computational costs associated with them, the cost differs according to the complexity of the algorithm and therefore affect efficiency at different rates.

- The ACK timer: is a key mechanism in packet loss detection, it defines a window of time a sender must wait to receive an acknowledgement from the receiver for a data segment. The window of time determines the protocols sensitivity to detecting packet loss and is vital in optimising the efficiency of transmissions as lost packets need to be detected and re-transmitted promptly for efficient transmission of data.

SEC-QUIC serves to mainly improve transmission efficiency of data segments. It would be most useful in connections that transport a large volume of traffic. This is because optimising the tradeoff in these cases can yield substantial benefits. Therefore, we conduct experiments to investigate the tradeoff in varying traffic loads to illustrate how security and network configurations affect efficiency of transmission. Furthermore, we know that QUIC fails in providing performance gains over TCP in network conditions, where there are high bandwidth and a low packet loss rate [6]. Therefore, by leveraging its unique implementation in the HTTPS stack, using application-level information, such as identifying a mobile web client through the user agent, SEC-QUIC can use data-driven understanding to optimise the tradeoff.

In what follows, we discuss elements of SEC-QUIC:

- **Configuration Templates** store a set of frequently used templates for a typical range of scenarios. A template provides a format for instantiating configurations. The database enables the data-driven function of SEC-QUIC and this acts like a cache in providing the Decision Maker with information for a particular context.

- **Decision Maker** is responsible for understanding the dynamics of QUIC to leverage the information provided by the configuration templates. Essentially, it configures the connection to optimise the tradeoff. It can dynamically readjust configuration parameters to improve security and/or performance aspects.

- **Device Resource Constraints** refer to an array of features related to resources, such as processing power. The variety in devices that have access to the web means that QUIC will be operating on a multitude of devices. These devices have different capabilities and computational constraints. As a result, security measures will have varying computational overhead depending on the CPU chips in the device. We suppose that the tradeoff can be reduced if more appropriate security mechanisms are used based on the constraints of the device in a connection.

- **Network Conditions** capture the type of connectivity, network bandwidth, RTT and loss rate in the network. QUIC's performance varies on the network conditions in which it is operating. To reduce the tradeoff, we investigate how QUIC's congestion control mechanisms can be leveraged under different network conditions to improve the efficiency of transferring video files.

- **Applications** infer the type of data being transmitted and its size. Moreover, an application considers the type of client: mobile or desktop. SEC-QUIC can use this information to evaluate the impact of the data type and size on the client to produce the optimal configurations.

By conducting a series of experiments, we identify scenarios where configurations can be fine-tuned to yield performance gains. The experiments demonstrate how QUIC's security-efficiency tradeoff can be balanced on a per scenarios basis via increasing the maximum packet size, cipher suites , and packet loss detection mechanisms to manage resource consumption at the sender and the efficiency of file transmissions.

3.2 | Use Case Scenarios

The diverse capabilities of client end points utilising QUIC for video streaming results in devices with constrained resources incurring a higher cost of security (i.e., cipher suite) with the use of default configurations. Thus, the cost of security on resource consumption, such as CPU and memory usage, is reflected in the efficiency of data transmissions. As such, this can degrade performance for certain applications and network environment use cases.

Uninterrupted Playback

Applications that provide live video streaming services require uninterrupted video playback and the performance of such applications is directly dependent on the transport layer. As such, the efficiency of transport layer operations impacts video playback interruption. Unlike common file transfers, the Quality of Service (QoS) of live playback is highly time-sensitive. Unfortunately, the security overhead of cipher suites can lead to a degradation in performance and the loss of packets can be costly in terms of QoS. The reason is by the time lost packets are identified and re-sent, it can be too late and lead to playback interruptions. Therefore, loss detection and the resource overhead of cipher suites are critical factors in improving the performance of video streaming applications. Thus, a key requirement of SEC-QUIC is providing video stream reliability by optimising the security-efficiency tradeoff.

Heterogeneous Network Environments

Mobile data networks, such as 4G and LTE, exhibit properties that are different from those of traditional fixed networks, such as constraints in bandwidth and varying packet loss. These differences result in performance degradation (efficiency) at the transport layer, with default configurations of transport layer dynamics typically implemented for fixed networks. This creates a need for transport layer protocol optimisation to be performed on-the-fly for video streaming applications, especially in cases where an end host is switching between different networks while maintaining a connection, to ensure throughput performance is sufficient. Yamin et al.[14] emphasised the need for security in location-based services as they expose user data to trust-based exploits. Security controls implemented at this layer further add to the performance overhead[15,14]. Thus, a key requirement for SEC-QUIC is to optimise transport layer performance in heterogeneous networks by leveraging its data-driven nature to improve the efficiency of transmissions according to the network conditions.

3.3 | Experiment Platform

Although several QUIC client and server implementations have been standardised by the Internet Engineering Task Force (IETF), the Chromium[16] implementations is chosen due to several reasons. The most important one is it is the latest version of the QUIC protocol and any recent changes implemented can be observed in our testing. Chromium provides the greatest flexibility in testing as it allows for adjusting parameter values, it is a commonly used implementation in the literature. The server is run on an Amazon EC2 instance (Ubuntu 16.04, 16 GB memory, AMD EPYC 2.5GHz quad-core) while the client application is run on a Windows 10 machine with 16 GB memory and Intel(R) Core i7 1.8GHz quad-core processor.

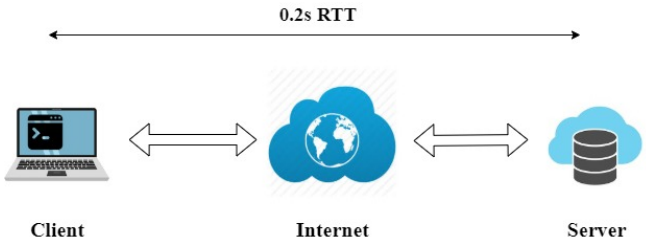


FIGURE 2 Test bed setup of client server architecture.

The testbed architecture, illustrated in Figure 2 , is similar to that of the seminal work by Kahki et al.[17] on evaluating QUIC's performance and allows for the investigation of MTU, cipher suites, the ACK timer, and their impact on the tradeoff. The testbed enables an accurate root cause analysis of the impact of parameters, such as file sizes and cipher suites, on the tradeoff by evaluating QUIC over traditional networks and isolating parameter configuration changes in different experiments. The

parameter settings refer to their default configurations, which are highlighted in bold in Table 1 , using specifications, and the latest Chromium default implementations [18,16]. To optimise the tradeoff based on different scenarios, we measure QUIC's performance under different configurations and network conditions to understand how each scenario requires fine-tuning of the default configurations to optimise the tradeoff.

TABLE 1 Experiments parameters and configurations.

Parameter	Values
RTT	0.20s
Extra packet loss	0.1%, 1%
Number of files	5
File sizes	1KB, 10KB, 1MB, 10MB, 100MB
Client	Windows chromium `quic_client`
MTU	**1200**, 1400 Bytes
Cipher suites	**AES**, ChaCha20 (CC20)
ACK timer	15, **25**, 35 (milliseconds)

By examining QUIC's performance in different scenarios listed above, we can identify scenarios where the default configurations are not optimised. We explore QUIC's dynamics, in particular the tradeoff between security and efficiency in video transmissions, by investigating:

- The impact varying MTU sizes have on efficiency in the event of loss in the network and how they can be leveraged to optimise the tradeoff by reducing resource costs at the sender.
- The per packet overhead of AES and ChaCha20 and their performance under different scenarios.
- ACK timer windows and their impact on resource consumption at the server and transmission efficiency.

3.4 | Data Collection

An end point transmits its data object as a series of packets. It is critical to improving this process to increase the efficiency of video transmissions. We test the factors that influence the tradeoff on a variety of video file sizes to understand the effect these parameters have on the performance of the protocol as the traffic volume increases. We quantify the tradeoff through the following metrics and outline how they relate to measuring the tradeoff under different scenarios:

- Average CPU Usage: There is a cost associated with sending and receiving packets. We measure this cost on the sender side as servers carry out critical functions such as monitoring the networks RTT and estimate bandwidth to avoid congestion in the network, running timers for loss detection as well as retransmitting lost packets. We aim to measure two main sources of computational cost for QUIC including the overhead per-packet and encryption process. Thus, the computational cost of QUIC functions is critical in evaluating both the security and efficiency aspects of the tradeoff as inefficient use of computational resources is both a security and efficiency issue.
- Throughput: The transmission rate details the number of data segments being sent and received over the network link. Throughput measures the actual transfer rate achieved between sender and receiver. Thus, it will be effective in evaluating the efficiency of transmission under different scenarios.
- File Transmission Time: The time elapsed between the client sending a request (for a video file) to the server and the client receiving the last bit of the video file from the server is measured to quantify the impact on efficiency.

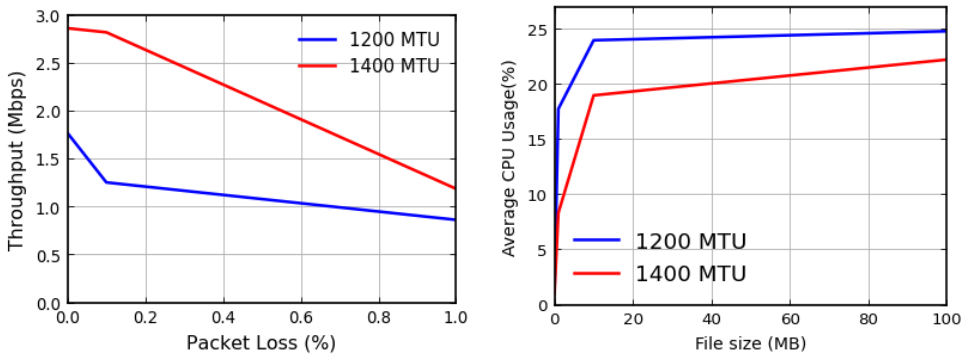
4 | OUR FINDINGS

This section details the results obtained and discusses the impact of the parameters investigated on the security-efficiency tradeoff in QUIC. The parameters tested include varying MTU sizes, cipher suites, and ACK timer timeouts under varying network

conditions detailed in Table 1 . For each parameter tested, the results captured describe QUIC's performance either in terms of throughput at the client side or the file transmission time and the average CPU usage of the server during the transmission of the data object. These parameters are used to evaluate the security and efficiency aspects of the tradeoff. By observing the impact on QUIC's performance, we are able to develop an initial understanding to design a conceptual framework to intelligently optimise the tradeoff for based on the connection's context.

4.1 | Impact of MTU on Tradeoff

The MTU specifies the UDP payload size. The maximum size of the payload allowed within a packet is vital in the overall efficiency of transmission as it specifies the amount of data each packet can carry. Thus, increasing its size means that a lower volume of packets is required to transmit a data object, and vice versa. As a result, it reduces the processing required at the endpoints. QUIC, like TCP, implements that a receiver sends an ACK for every two packets that it receives. Although, this is a standard setting, receiving and processing ACK increases the computational cost for the server. Moreover, QUIC encrypts its ACK packets that increase overhead at both endpoints. A direct consequence of increasing the MTU decreases the overall ACKs sent by the receiver and can reduce the computational overhead at the sender. However, a low ACK rate can reduce the throughput in a connection and risk the result of a slow start early in the connection. Since a larger MTU is only decreasing the volume of ACKs sent in the transfer of an object and not the ACK frequency itself, this risk can be largely avoided as illustrated in Figure 3 (a) where the throughput of 1400 MTU size is consistently higher.



(a) (b)

FIGURE 3 The effect of MTU on: (a) throughput and packet loss at the client of 100Mb object transfers and (b) average CPU usage at the server.

Therefore, it is beneficial for the sender to send the largest packet size possible. However, this ideal does not always equate to best practice as large MTU sizes can have detrimental impacts on the network and the efficiency of transmission. This is due to packets with higher payloads being more likely get dropped on the network path resulting in them not reaching the intended destination[6]. Thus, there is an inherent tradeoff in increasing the MTU size where larger sizes may increase efficiency at the risk of unreachability. For this reason, the QUIC specification recommends a default minimum of 1200 bytes[19]. We explore this relationship for the transport of video objects to demonstrate how increasing MTU for the transport for data objects can yield performance benefits. Subsequently, Figure 3 (b) shows the impact MTU sizes has on the average CPU usage of the server in the transmission of video objects. While Figure 3 (a) contrasts the performance between 1200 MTU size and 1400 MTU for the transmission of all video objects under different packet loss rates in the connection.

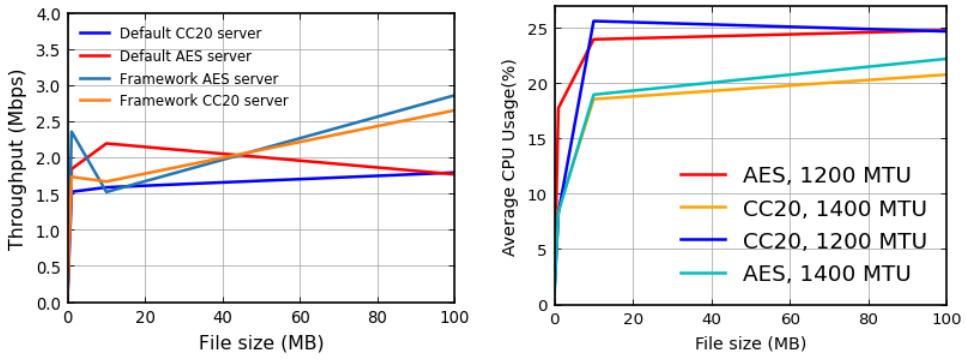
Critically, we observe little difference in performance between the payload sizes in short live traffic of 1KB, 10KB, and 1 MB. This is largely due to the minimal number of packets required to transmit the object resulting in a negligible difference between MTU sizes. In long live traffic (e.g., 10MB and 100MB), however, a small payload size becomes an inhibiting factor requiring the transmission of a greater volume of packets. This adversely affects performance and can be observed in the greater resource consumption at the server as shown in Figure 3 (b). Our findings indicate that in scenarios where the network path can support

larger MTU sizes, the security-efficiency tradeoff can be optimised as illustrated in Figures 3 (a) and 3 (b). As such, SEC-QUIC can leverage MTU to effectively balance the tradeoff by increasing the MTU size in networks where its unreachability does not degrade its performance while yielding performance gains over the default value of 1200 bytes. In doing so, both the throughput and resource consumption at the server and client can be reduced - as clients produce less ACKs and the server is able to transmit data at a faster rate.

To further investigate the data-driven nature requirement of SEC-QUIC, we first investigate the dynamics of QUIC by observing the impact on latency produced by the change in MTU size on the transport of video data.

4.2 | Impact of Cipher Suites on Tradeoff

The cipher suite utilised in a communication channel is extremely important in terms of security, but it also affects the efficiency of communication. Primarily, a security mechanism is employed to provide the main principles of security: confidentiality, integrity, and authentication of traffic in a connection. This has a computational cost that adversely affects efficiency as it slows down the overall process of communication by encrypting all outgoing packets. This cost is mainly dependent on the cryptographic algorithm used for encryption and the key size. The greater the complexity of the cryptographic algorithm, the higher the computational cost due to more modes of operation being performed on the plain text to convert into cipher text. Thus, the negotiated cipher suite for a connection between two end points will have an impact on the efficiency of data transmission as the payload for each packet is encrypted. Furthermore, due to QUIC encrypting ACKs, the cost of a cipher suite results in both endpoints performing encryption and decryption operations unlike the TLS/TCP stack. This, in turn, adds to the computational overhead for the transmission of data objects.

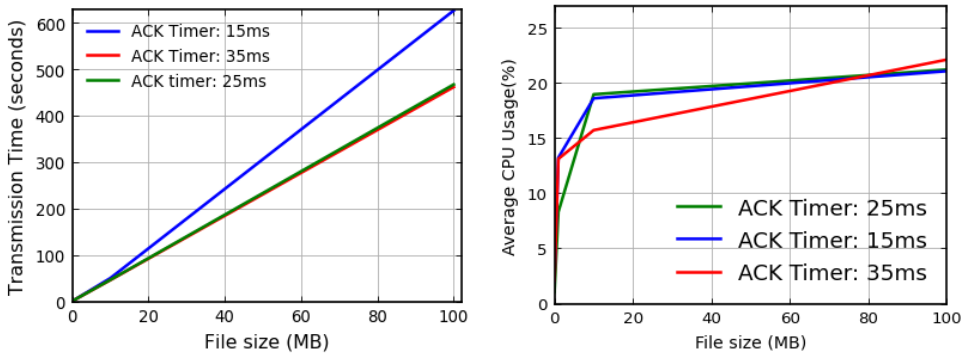


(a) (b)

FIGURE 4 The effect of cipher suite on: (a) throughput and (b) server's CPU consumption.

Therefore, we examine the impact cipher suites have on the tradeoff in this scenario. The supported cryptographic algorithms in Chromium's QUIC example server and client are limited to AES and ChaCha20[16]. The cipher suite is selected during the connection negotiation. The client's choice is given preference when negotiating the cipher suite and the default preference is set to AES in QUIC. We observe the per packet cost introduced by both cipher suites under different scenarios. We find in this test that the difference between the cipher suites is widened as the volume of traffic increases. Notably, we find that there is negligible difference in performance between the two cipher suites in the transport of short and medium live traffic. It is in the transport of long live traffic that AES proves to be significantly more efficient. While this observation is restricted to a limited environment, it demonstrates the impact cipher suites can have on the tradeoff.

In resource-constrained devices, cipher suites are likely to have a greater impact due to the computational overhead incurred by more complex cryptographic algorithms. Due to the large variation in devices that use QUIC, the primary encryption algorithms have a higher cost to those devices that are constrained in resources. Therefore, the use of a lightweight encryption algorithm for those devices can reduce the cost of security and improve the performance of QUIC. While the client hello protocol will state the cipher suites supported by the client, QUIC is configured to choose the most computationally heavy algorithm stated



(a) (b)

FIGURE 5 The effect of ACK timer intervals on: (a) file transmission time and (b) average CPU usage at the server.

by the client. However, in the case of video streaming, this comes at a higher cost compared to that of other applications where the amount of traffic is smaller.

We observe that AES outperforms CC20 in the case of using a MTU of 1400 bytes while also requiring less computational usage. This is illustrated in Figures 4 (b) and 4 (a) where the framework settings outperform the default configurations. This is further shown in Figure 4 (a), where there is a small difference in latency between AES and ChaCha20 for the transmission of both small and large files. However, when the traffic becomes larger, the computational overhead of the cipher suites begins to impact latency – Figures 4 (b). We propose that this effect would be greater in resource-constrained devices. Thus, by considering these factors, the Decision Maker in SEC-QUIC can optimise the tradeoff by reducing the security cost through the configuration of lightweight cipher suites.

4.3 | Impact of ACK Timer on Tradeoff

The ACK timer specifies the time interval in which QUIC must wait to retransmit a packet. It is a mechanism through which packet loss in the network is detected, thus requiring a retransmission of the segment to ensure data delivery in the absence of any feedback from the receiver[20]. Loss detection is vital in ensuring the reliability of data delivery in a transport protocol. This parameter can have a major impact on the efficiency of data transmission as in some cases, such as low-loss networks, it can be more beneficial for the sender to be more aggressive in expecting an ACK. However, it is important to strike the right balance as an overly aggressive approach can lead to unnecessarily adding to network congestion causing a drop in performance due to the sender forcing ACKs for segments before the receiver has actually received the data. Thus, it is critical to have the correct implementation to preserve network stability and avoid congestion collapse[20].

As such, our aim is to illustrate how SEC-QUIC can leverage this factor in low loss networks to improve the efficiency of transmission and thereby optimise the tradeoff. We observe that, in our scenario, a shorter time interval of 15 milliseconds (ms) is overly aggressive than the 25ms default configuration set by QUIC, as illustrated in Figure 5 (a), resulting in longer transmission times due to the retransmission of more packets. Moreover, we find that the default ACK time intervals are more economical in terms of resource consumption, see Figure 5 (b). This reveals the inherent difficulty in fine-tuning timers due to its intrinsic limitations, i.e., networks are dynamic and finding a value that achieves the desired performance is challenging as loss detection requires more information than timeout timers[21]. However, given a data-driven approach, such as SEC-QUIC, external contextual information (e.g., network conditions and device constraints) can be leveraged to fine-tune timers. Thus, striking a balance in which efficiency is improved without compromising the state of the network is a crucial consideration for SEC-QUIC so that it not only calculates the ideal solution for the end points in a connection but also estimates the impact of congestion in the network.

4.4 | Discussion

The findings from evaluating MTU, cipher suites, and the ACK timer reveal how these configurations can be leveraged to balance the security-efficiency tradeoff at the transport layer. SEC-QUIC conceptualises the use of a data-driven approach, where a semantic knowledge base containing a variety of configuration templates that fine-tune QUICs dynamics based on the use case scenario to balance the security-efficiency tradeoff. For instance, in the case of uninterrupted live video playback on a mobile device, SEC-QUIC would aim to lower the cost of encryption, configure an MTU with a high reachability rate and higher sensitivity to loss detection by having a shorter ACK timer as high reliability of data delivery is more desirable in live video streaming cases. Thus, SEC-QUIC can balance the tradeoff by similarly altering parameters in use cases where video stream reliability is a key requirement. On the other hand, in the case of common file downloads, the cost of lost packets is not as high on QoS, which can allow for the use of larger MTU sizes to increase the efficiency of file transmission by leveraging a higher throughput rate. Similarly, switching between networks while either live video streaming or regular file downloads required SEC-QUIC to consider the network conditions of the network to accordingly change the configuration policy dynamically. As such, the requirement to optimise efficiency by improving throughput can be achieved through increasing MTU as illustrated in Figure 3 (a). On the other hand, in order to be effective and accurate, SEC-QUIC would require the collection of real-world examples to create a large database. This can introduce additional overhead to the handshake due to the need to identify the optimal configuration templates for a given connection. Overall, SEC-QUIC would serve to fulfil the two main requirements of video stream reliability and improve efficiency in heterogeneous networks by leveraging transport layer dynamics, as demonstrated in the experiments:

- Increasing MTU, from 1200 bytes to 1400, increases throughput in network environments with a range of packet loss rates (Figure 3 (a)).
- The overhead differences between encryption algorithms degrade performance in the transfer of long flow traffic, such as 100 MB, due to the compounding impact of computational complexity on flows with large packet volumes.
- Although optimally configuring the ACK timer for desired performance is a non-trivial task, the similarity of resource consumption between various timeout values illustrates how they can be leveraged to improve loss detection, due to the lack of additional overhead, in a data-driven framework.

5 | CONCLUSIONS AND FUTURE WORK

In this article, we studied the tradeoff between security and efficiency for QUIC-based video transmissions. By conducting a series of experiments, we investigated a set of network and application parameters and security schemes. Consequently, this work helped us in understanding the relationship between different communication and security settings and their achievable performance. On this basis, we presented a conceptual framework, SEC-QUIC, that considers contextual information of a connection to provide a set of configurations to optimise the tradeoff. Moreover, through our results, we observed the factors that influence the tradeoff to illustrate how SEC-QUIC can leverage these factors to increase efficiency or reduce the cost of security.

Despite gaining insights into QUIC behaviour of video traffic transmissions, the client and server in Chromium source code might not fully be representative of the real systems employed by Google. The results, therefore, might differ as a consequence. However, the Chromium QUIC implementation is the most relevant version available for research to evaluate QUIC as it is provided by Google and remains the most popular implementation used in literature. Moreover, we did not collect resource consumption information on the client side and thus lacked a holistic view of how the configurations can affect the tradeoff at the client end. Future work should, therefore, focus on minimising these limitations by obtaining more data from clients to consolidate our understanding of the tradeoff. In this article, we studied two cipher suites: AES and ChaCha20. In the future, we aim to investigate QUIC's performance with the array of algorithms supported in TLS 1.3 in a variety of devices, mobile and desktop, to evaluate the cost of security across devices. Thus, we aim to investigate the tradeoff in resource-constrained devices to build the data-driven aspects supporting our SEC-QUIC design. In summary, we showed how the tradeoff between security and efficiency can be optimised through leveraging connection configurations, i.e., transport layer dynamics. We propose a framework that considers the contextual information of a connection to provide the optimal configurations that allow for both secure and efficient transmission of data.

References

1. Cullen C. Sandvine releases 2019 Global Internet Phenomena Report. https://www.sandvine.com/press-releases/sandvine-releases-2019-global-internet-phenomena-report; 2019. Last Accessed: April 1, 2021.
2. Dupont B. The ecology of cybercrime. In: Routledge. 2019 (pp. 389–407)
3. General Data Protection Regulation (GDPR) Compliance Guidelines. https://gdpr.eu; . Last Accessed: April 1, 2021.
4. Lampson B. Computer security in the real world. Computer 2004; 37(6): 37–46. doi: 10.1109/mc.2004.17
5. Kurose JF, Ross KW. Computer networking: a top-down approach. Boston, MA, USA: Addison Wesley . 2013.
6. Langley A, Iyengar J, Bailey J, et al. The QUIC Transport Protocol. In: ACM Press; 2017
7. Honda M, Nishida Y, Raiciu C, Greenhalgh A, Handley M, Tokuda H. Is it still possible to extend TCP?. In: ACM Press; 2011
8. Ford B. Structured streams. ACM SIGCOMM Computer Communication Review 2007; 37(4): 361. doi: 10.1145/1282427.1282421
9. Seufert M, Schatz R, Wehner N, Casas P. QUICker or not? -an Empirical Analysis of QUIC vs TCP for Video Streaming QoE Provisioning. In: IEEE; 2019
10. Nathan V, Sivaraman V, Addanki R, Khani M, Goyal P, Alizadeh M. End-to-end transport for video QoE fairness. In: ACM Press; 2019
11. Bhat D, Rizk A, Zink M. Not so QUIC. In: ACM Press; 2017
12. Saverimoutou A, Mathieu B, Vaton S. Which secure transport protocol for a reliable HTTP/2-based web service: TLS or QUIC?. In: IEEE; 2017
13. Lychev R, Jero S, Boldyreva A, Nita-Rotaru C. How Secure and Quick is QUIC? Provable Security and Performance Analyses. In: IEEE; 2015
14. Yamin M, Abi Sen AA. Improving privacy and security of user data in location based services. International Journal of Ambient Computing and Intelligence (IJACI) 2018; 9(1): 19–42.
15. Dey N, Santhi V. Intelligent techniques in signal processing for multimedia security. Springer . 2017.
16. Anon . Playing with QUIC. https://www.chromium.org/quic/playing-with-quic; . Last Accessed: April 1, 2021.
17. Kakhki AM, Jero S, Choffnes D, Nita-Rotaru C, Mislove A. Taking a long look at QUIC: an approach for rigorous evaluation of rapidly evolving transport protocols. In: ; 2017: 290–303.
18. J. Iyengar E. QUIC: A UDP-Based Multiplexed and Secure Transport draft-ietf-quic-transport-34. tech. rep., RFC 9000, January; : 2021.
19. Iyengar J, Thomson M. QUIC: A UDP-Based Multiplexed and Secure Transport. Internet-Draft draft-ietf-quic-transport-29, Internet Engineering Task Force; : 2020. Work in Progress.
20. Paxson V, Allman M. Computing TCP's Retransmission Timer. tech. rep., RFC 2988, November; : 2000
21. Psaras I, Tsaoussidis V. Why TCP timers (still) don't work well. Computer Networks 2007; 51(8): 2033–2048.