A QUIC way to bypass your firewall

An overview of what the QUIC protocol is and how it affects the network and security infrastructure

Author: Boban Glisovic

swisscom

Content

Summary

An internet primer

What's QUIC and why is it eating the internet?

Security promises and implications of QUIC

Conclusion & future outlook

Summary

QUIC is a new transport protocol initially developed by Google and standardized by the Internet Engineering Taskforce in 2021. It addresses many longstanding challenges in the internet and combines most of the value provided by protocols like TCP, TLS, and HTTP/2. QUIC is an encrypted, connection-oriented protocol operating on top of UDP and is the foundation for HTTP/3. Today, about 25 percent of the total internet traffic throughput is already based on QUIC. This fast and widespread adoption is primarily driven by global content providers such as Google, Meta, Microsoft, Cloudflare, and numerous other Content Delivery Networks. Additionally, popular services such as YouTube, Instagram, or Microsoft Office 365 now offer HTTP/3 by default.

QUIC aims to make the internet faster, more reliable, and more secure. By streamlining the protocol's handshake process, QUIC effectively reduces latency and speeds up connection setup. Furthermore, its ability to parallelize multiple data streams improves performance and enhances the overall user experience. Over the past decade, the consumption of services via mobile devices has steadily increased. Connections based on radio networks are more prone to packet loss. QUIC's algorithm enhances error detection and loss response. For example, YouTube video buffering time was reduced by up to 18 percent.

Another significant achievement is its built-in encryption capability. Most importantly, it cannot be turned off, so communication remains secure. Furthermore, the encryption is not limited to the user data but spans the connection metadata. This positively impacts a user's privacy since it becomes harder for attackers to eavesdrop on the communication. Further, it prevents protocol ossification and increases flexibility and extensibility in the future. Additionally, QUIC maintains the overall throughput performance compared to TCP while enforcing encryption at the cost of increased CPU usage.

Summary

While QUIC comes with many significant improvements, there are some adverse implications on existing technologies – namely in network management and cybersecurity. For example, many security features have been built around inspecting the protocol's details. QUIC prevents this, bypassing some conventional security systems or lowering their effectiveness. As such, traditional firewalls can no longer track connections and may open unintended gaps, allowing attackers to exploit and infiltrate the system. Since QUIC uses encryption, specifically the connection metadata, advanced features such as deep packet inspection are no longer effective. This makes it impossible to scan for malware or apply data loss prevention rules.

QUIC is a promising development; looking at the current growth rate, it will likely stay and become the dominant internet transport protocol over time. However, because all major browsers support QUIC and enable it by default, except for Safari, users might use QUIC without noticing.

For organizations and their cybersecurity requirements, allowing QUIC comes with consequences. Therefore, it is a crucial topic to understand and carefully evaluate how to deal with it. Enterprise security providers today suggest blocking QUIC traffic on the network and disabling it on managed devices through policies. As of today, this will not negatively impact the users since an automatic downgrade to TCP will happen behind the scene. However, QUIC's flexibility means that new QUIC-only services could emerge suddenly, necessitating a review of organizations' cybersecurity strategies to accommodate new services and technologies while maintaining or strengthening security measures. Emerging concepts like Zero-trust, which do not rely on inline traffic management, help bridge that gap and provide features to mitigate some of the identified risks.

An internet primer

Internet communication as we know it today is composed of different protocols organized around the layered OSI model. In order to understand what QUIC is and how it impacts the protocol stack, an overview of the most critical protocols is provided.

TCP/IP: the foundation of today's web

More than 40 years ago, the Internet Protocol (IP) and the Transmission Control Protocol (TCP) were standardized by the Internet Engineering Taskforce (IETF). Most Internet and web traffic today is based on this protocol stack. IP was designed to transmit data blocks in a network where a unique IP address identifies all sources and destinations. Devices in the network understand the IP addresses and can forward the packet to the correct next destination. Like the postal service, network packets are transferred between multiple hops (e.g., local post office, distribution center) until it has arrived at their destination.

TCP fits the layered protocol architecture (OSI model) just above the Internet Protocol layer. It intends to provide a highly reliable end-to-end connection between two hosts in a network. TCP uses a range of port numbers to support multi-service communication between hosts.

TCP vs UDP

Next to TCP, there is mainly User Datagram Protocol (UDP) as an alternative transport protocol. However, they fundamentally differ in their functionality and are used for different use cases. For example, UDP is mainly used for time critical applications such as the Domain Name System (DNS). Hence, its relevance has grown over the past years with the rising popularity of streaming services (e.g., IPTV live streaming) and gaming in particular. Similarly, QUIC relies on UDP fundamentals; more on that in the next chapter.

UDP, compared to TCP, is connectionless and does not provide any reliability guarantees. It is focused on swiftly transmitting data, making it a good choice for live or real-time data. In addition, UDP can be used for multicast or broadcast traffic, whereas TCP is limited to host-to-host communication (unicast). TCP, on the other hand, is focused on providing reliability at the price of efficiency and latency. Features like packet ordering, flow control, or congestion control are only available in TCP.

TLS: Transport Layer Security

While sending application data over TCP/IP works fine, it comes with the drawback of cleartext information that any observer on the network can read. With the introduction of Transport Layer Security (TLS) and its predecessor, Secure Socket Layer (SSL), this problem has been mitigated. TLS provides encryption capabilities on top of TCP, allowing encryption of all user and application data. Today 81% of all websites use HTTPS - the combination of HTTP and TLS as the default for browsing their content.

HTTP: Hypertext Transfer Protocol

HTTP has been used since the early 1990s and was standardized in 1996 as HTTP/1.0. It is an application-level client-server protocol at the foundation of any data exchange on the web. Clients use the request/response-based protocol, usually web browsers, to fetch resources (e.g., HTML documents, images, etc.) from a corresponding web server.

What is QUIC and why is it eating the internet?

QUIC (pronounced “quick,” no acronym) is a new secure transport protocol standardized and published by the IETF in May 2021 with a series of RFCs (RFC8999, RFC9000, RFC9001, RFC9002). The web is based on protocols designed long before anyone could anticipate today’s usage patterns and global scale. QUIC addresses many of the drawbacks encountered in today’s protocol stack. It aims to provide a performant (fewer roundtrips), reliable, secure, and privacy-enhanced alternative compared to TCP.

A transport protocol revolution

Google initially designed the first version of QUIC referred to as gQUIC (Google Quick UDP Internet Connection) to make web browsing faster and improve the user experience in modern web applications. gQUIC combined transport protocol and HTTP features into one protocol using custom cryptography. In 2017 these efforts were brought into an IETF working group. Together with many other large content providers such as Cloudflare, Akamai, and Facebook, it was first standardized in 2021 in the RFC 9000.

QUIC is a connection-oriented transport protocol providing reliable data delivery between two hosts, similar to TCP, with some critical differences built on top of UDP. Contrary to what might sound obvious, UDP was not chosen in the first place because of its performance improvements. QUIC could in theory, directly run on top of the IP protocol, just as any other transport protocol. The main reason behind this design choice is based on the fact that many network devices and clients have an opinionated view on what protocols (TCP, UDP) to expect on top of IP. Adapting millions of devices, firmware, operating systems, and even hardware to a new protocol is nearly impossible.

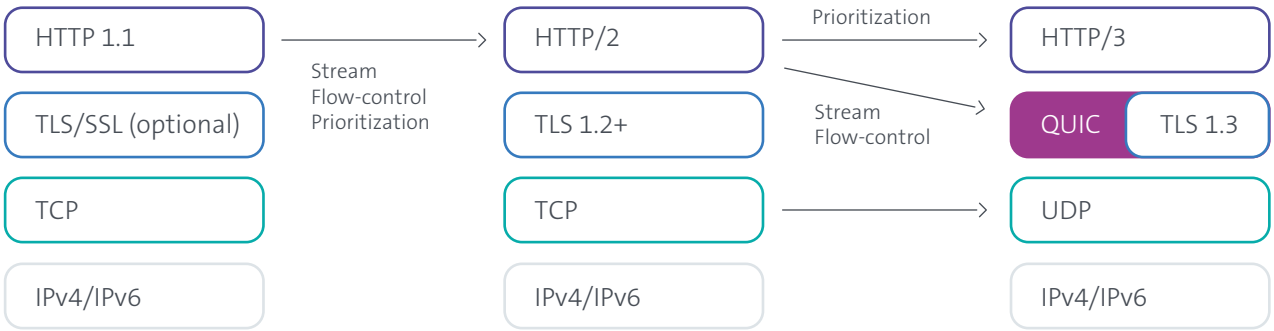

Figure 1 Evolution of protocol stacks

Versioning and deployment: Since QUIC is built on top of UDP, it can be fully implemented in the user space, dispensing the need for operating systems or firmware adaption. With a built-in version negotiation algorithm, QUIC versions can easily be updated and deployed across the internet. Middleboxes such as firewalls, Network Address Translation, and load balancers are widespread in today's internet architecture, and due to QUIC's design, deployment is not blocked by middleboxes.

Encryption: QUIC packets are always encrypted. Even most header and metadata information is encrypted or at least authenticated. As standardized in RFC 9001, TLS1.3 is used as the security component of QUIC. TLS1.3 is the most recent evolution of the TLS protocol. It provides two hosts with a way to establish secure communication over an untrusted network (e.g., the internet). TLS ensures the authentication of peers and provides confidentiality and integrity for data.

While TLS1.3 is also used on TCP-based transport (e.g., with HTTP/2), with QUIC, it is more deeply integrated into the transport. This prevents middleboxes from eavesdropping or modifying the connection.

Faster connection setup: One significant improvement with QUIC compared to the TCP/TLS stack is its connection setup process. The handshake process of a TCP/TLS based connection requires up to three round trips to send application data. With QUIC, it is possible to reduce these round trips significally and send application data in the first packet. This faster handshake process is called 0-RTT (round trip time).

While with QUIC the initial handshake between the client and server always includes at least one round trip, subsequent connections can benefit from already agreed security parameters and the 0-RTT feature.

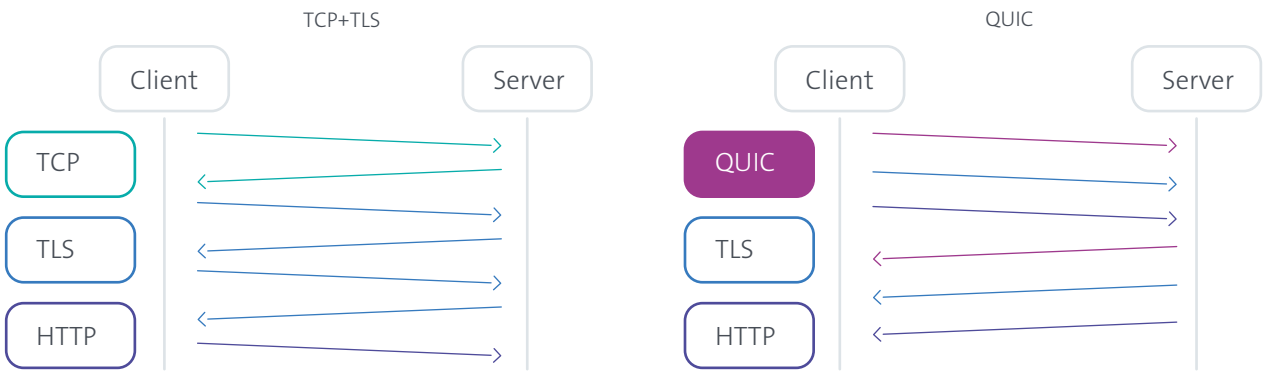

Figure 2 Comparison TCP/TLS Handshake vs. QUIC 0-RTT

What is QUIC and why is it eating the internet?

Multiplexed data streams: Multiplexing was first introduced with HTTP/2 on the application level and transferred to the transport layer with the evolution of QUIC and HTTP/3. The primary objective is to address the head-of-line blocking problem known from TCP and HTTP/1.1 as a significant contributor to an improved user experience, especially in environments suffering from network congestion and packet loss (e.g., cellular network). Furthermore, multiplexing allows the same underlying connection to encapsulate multiple parallel and independent streams of data (e.g., different resources required to load a website) resulting in two major improvements. First, streams that are successfully received can already be processed by the recipient (e.g., elements of a website can be loaded earlier), and second, in case of packet loss, only the data of the affected stream must be resent compared to TCP where all packets had to be resent. Lastly, the ability to prioritize individual streams was introduced, allowing the application to influence certain aspects of their communication.

Connection migration: QUIC allows the client-side migration of a connection among different network interfaces. Switching on a mobile device from Cellular to Wifi, a QUIC session can be seamlessly continued as it no longer depends on the underlying IP address or port to identify the connection.

Too big to fail? – use cases and deployment status

Although QUIC RFC was only published in May 2021, by that time, around 12% of global internet traffic was based on QUIC in combination with HTTP/3, according to Cloudflare Radar. The main driver behind the fast and widespread adoption in its early days can be found in the ease of deployability. Since the QUIC protocol can entirely run in user space, no operating system kernels or firmware had to be upgraded to support it. Furthermore, since it is based on UDP transport – well known and supported by many devices for decades – middleboxes pass along the traffic if UDP is allowed. While the needs and learnings influenced QUIC from HTTP, several other application-level protocols are enhanced and adapted to use QUIC as a transport protocol.

HTTP/3

Running HTTP over QUIC is the most prominent use case for QUIC transport. However, most of today's end-user traffic is based on HTTP, where improving performance, privacy, and security has a huge impact. HTTP/3 is a slimmed version of HTTP/2, the features introduced in HTTP/2, like stream multiplexing or enhanced flow control, have been moved to QUIC.

So, are you using QUIC already? – Most likely, yes! If you consume popular web services, social media, or streaming platforms, QUIC might be used already. For example, watching a YouTube video results in pure QUIC network traffic, which can be seen in Figure 3 sniffing a YouTube session.

No.	Time	Source	Destination	Protocol	Length	Info
363…	2.376658	10.0.0.185	172.217.168.78	QUIC	79	Protected Payload (KP0), DCID=d5c150e9cf0c27a2
364…	2.378036	10.0.0.185	172.217.168.78	QUIC	76	Protected Payload (KP0), DCID=d5c150e9cf0c27a2
364…	2.378391	10.0.0.185	216.58.215.234	QUIC	1292	Initial, DCID=9a97df480d9422b4, PKN: 1, PADDING, CRYPTO, PING, PING, PING, PADDING, PING, CRYPTO, PADDING,
364…	2.378465	10.0.0.185	216.58.215.234	QUIC	115	0-RTT, DCID=9a97df480d9422b4
364…	2.378796	10.0.0.185	216.58.215.234	QUIC	864	0-RTT, DCID=9a97df480d9422b4
364…	2.379507	172.217.168.78	10.0.0.185	QUIC	71	Protected Payload (KP0)
364…	2.380150	172.217.168.78	10.0.0.185	QUIC	68	Protected Payload (KP0)
364…	2.381102	10.0.0.185	172.217.168.78	QUIC	76	Protected Payload (KP0), DCID=d5c150e9cf0c27a2
364…	2.382395	172.217.168.78	10.0.0.185	QUIC	68	Protected Payload (KP0)
364…	2.398967	216.58.215.234	10.0.0.185	QUIC	1292	Protected Payload (KP0)
364…	2.398967	216.58.215.234	10.0.0.185	QUIC	820	Protected Payload (KP0)
364…	2.398968	216.58.215.234	10.0.0.185	QUIC	240	Protected Payload (KP0)
364…	2.398968	216.58.215.234	10.0.0.185	QUIC	67	Protected Payload (KP0)
364…	2.399251	10.0.0.185	216.58.215.234	QUIC	120	Handshake, DCID=da97df480d9422b4

Figure 3 Packet capture while watching YouTube

DNS-over-QUIC (DoQ)
Inspired by one of the initial DNS-Encryption efforts with DNS-over-TLS, a corresponding DNS-over-QUIC (RFC 9250) was published in May 2022. While DNS-over-HTTPS adoption has grown recently, work towards DNS-over-HTTP/3 (DNS-over-HTTPS-over-QUIC) has been started and is heavily pushed by Google.

SMB-over-QUIC
SMB (Server Message Block) is a communication protocol used to enable file sharing and network browsing (e.g., Windows network file shares). Microsoft has built its QUIC implementation called MSQUIC based on IETF QUIC. With SMB-over-QUIC, they introduce an alternative to the TCP network transport providing secure, reliable connectivity to fileservers over untrusted networks. SMB leverages QUIC to benefit from encryption, the 0-RTT connection setup, and parallel independent streams.

To adopt QUIC, both the client (e.g., Application or Browser) and the corresponding backend service need to support QUIC. With many big tech companies and global content providers actively driving the standardization of QUIC, many of their services have already supported QUIC since day one.

Client-side QUIC support

Native mobile applications like Whatsapp, Instagram, or Chromium-based web browsers had very early support for HTTP/3 and the underlying QUIC protocol. Taking a deeper look into the usage statistics of relevant web browsers, a picture of the client-side QUIC deployment spread can be derived.

All major browsers support HTTP/3, most enable it by default except Safari. HTTP/3 being the most important protocol relying on QUIC as its transport, inherently browsers become the essential clients driving the adoption.

Comparing the usage distribution across the different browsers and considering their defaults, already 60 percent of today's users regularly use HTTP/3 and QUIC.

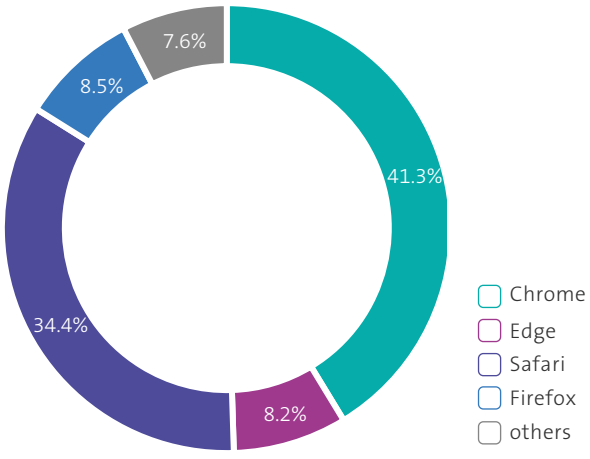

Figure 4 Distribution of browser usage in Switzerland, 2021, source: statista.com

If in doubt: fallback to an older HTTP version and TCP

While browsers generally support and prefer connections via HTTP/3, they need prior knowledge to assume whether a requested website supports it. Negotiating HTTP-related parameters (such as HTTPS or HTTP/3) is done through HTTP itself (HTTP Strict Transport Security, Alternative Service headers). The initial connection of a browser to a website today is always based on TCP (e.g., HTTP/2) and its first response indicates if the website offers HTTP/3 protocol support. With this limitation, implementing an HTTP/3 only website is hardly feasible.

Backend-side QUIC support

Billions of people today use services like Youtube, Instagram, Office 365, and Content Delivery Networks. They account for a large amount of internet traffic, resulting in QUIC already making up 25 percent of internet traffic throughput. Considering that the RFC was only released in 2021, this is an impressive achievement and shows the power of large tech companies. Those are the ones mainly driving the RFC standardization and participating in the IETF QUIC working group.

In comparison, Switzerland is lagging in its QUIC adoption. Despite Swiss network operators providing low-latency, high-bandwidth connectivity to most end users, the impact of QUIC is less noticeable than for others. Further, Switzerland has very high coverage with Apple iOS/macOS devices, with Safari being the default browser (29% Safari usage compared to 9% globally). Therefore, the use of QUIC may significantly increase as soon as HTTP/3 and QUIC are enabled by default in Safari.

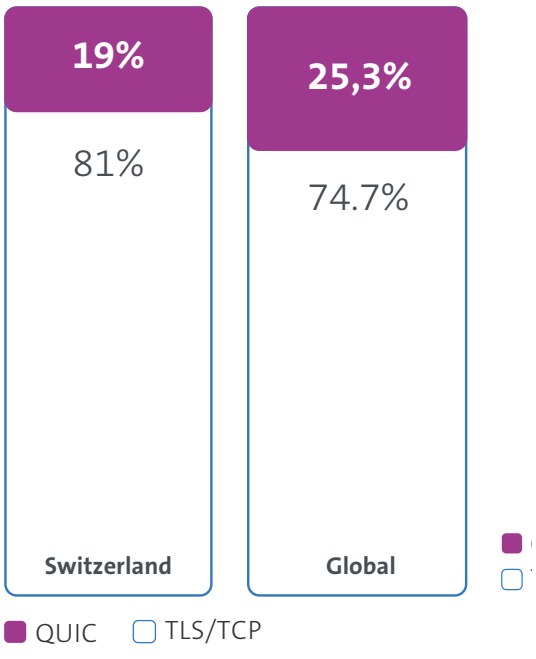

Figure 5 QUIC traffic throughput statistics 12/2022, source: Swisscom network monitor and w3techs

Enabling services to support QUIC and adjusting the application code is not required. Modern services usually are exposed to the internet through load balancers, reverse proxies, or web application firewalls. A typical pattern is to use these devices to terminate the QUIC connection; this eliminates the need for change on the corresponding backend service to serve QUIC traffic.

More vendors like BIGIP F5, HAproxy, and more are starting to support QUIC and HTTP/3 with their more recent software versions; this will further accelerate the deployment of QUIC-based traffic across the internet.

Performance comparison

Undoubtedly, QUIC brings some performance improvements and increases the end-user experience. For example, Google's data indicates that desktop users' search response latency decreased by 8% and mobile users by 3.6%. The rebuffer rates of YouTube playbacks were even reduced by 18% for desktop users respectively 15.3% for mobile users.

While the effective performance improvements which can be achieved highly depend on the underlying network infrastructure, there is an ongoing debate about the computational efficiency of QUIC. QUIC is still new, and implementations have not undergone a multi-decade optimization phase yet. Measurements show that the general throughput is comparable to TCP, while QUIC performs better on small files. The research also indicates an increased CPU resource usage by a factor of 2-3 for QUIC. Conversely, Fastly or Google show that parity in terms of resource usage can be achieved. Providing implementations with the support of hardware offloading to the NIC (Network Interface Card) will help.

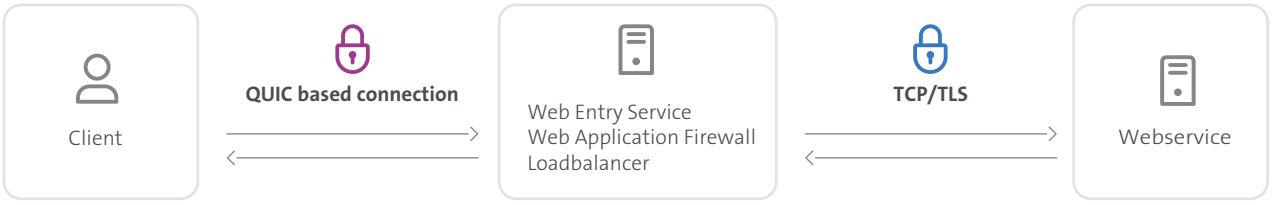

Figure 6 Possible deployment strategy with QUIC

Security promises and implications of QUIC

Protocols running on top of QUIC transport come with benefits in the areas of reliability, performance, and privacy. However, this introduces some challenges regarding network security, operations, and cyber defense mechanisms.

Big techs like Google, Meta, or Microsoft and global Content Delivery Network (CDN) providers such as Cloudflare or Akamai are the main drivers for adopting HTTP/3 and the underlying QUIC transport. Due to their global impact, adoption is progressing rapidly and is expected to continue. However, many security providers need to catch up, adopting QUIC in their products, and are, in many cases depending on the collaboration of key players.

Retaining network traffic visibility in the encryption era?

According to Google's transparency report, the rate of network traffic encryption stands at 94%. This is good for security. The more traffic that is encrypted, the harder it is for attacks to compromise a system, eavesdrop on communication, or steal data. But the same encryption technologies are used by cybercriminals to obfuscate and protect their activities. According to Zscaler's study, more than 80 percent of attacks in 2021 used encrypted communication channels. It is a fact that criminals are ahead of enterprises in their use of encryption.

QUIC encryption capabilities are deeply integrated into the protocol. In comparison to the TCP/TLS-based encryption approach, with QUIC, not only the user payload data is encrypted, but most of the QUIC header information is as well.

Encrypted Handshake: With the introduction of the TLS1.3 extension called Encrypted Client Hello (ECH), previous clear text information like the server name indicator is encrypted. A simple and effective option to determine the destination of an encrypted connection is lost. It is no longer possible to read the destination host/domain and apply simple allow or denylists for safe or problematic destinations, nor can a proxy intercept the connection. Additionally, with the adoption of encrypted DNS like DNS-over-HTTPS or DNS-over-QUIC, it is becoming impossible for network security administrators to monitor the destinations and apply necessary security measures.

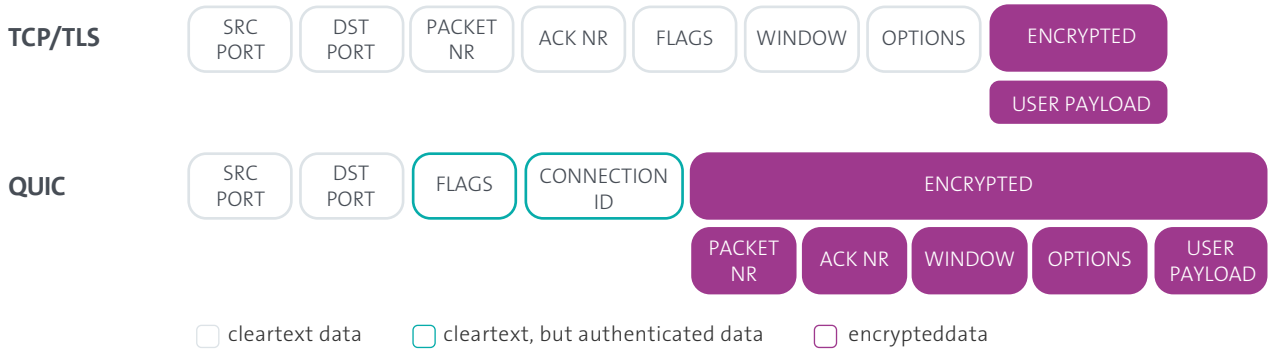

Figure 7 Comparison of TCP/TLS and QUIC encryption

Classify encrypted and evasive traffic with deep packet inspection.

Deep Packet Inspection (DPI) describes the process of intercepting communication to apply threat detection (e.g., malware scanning) or data loss prevention measures. A typical DPI either works as a man-in-the-middle system and will decrypt the packets, inspect them, and finally re-encrypt and forward them to their destination or passively mirror the traffic and process it offline. Being able to decrypt encrypted traffic requires the possession of the encryption keys. Today DPI solutions have yet to be as widespread due to their resource-intense nature, often requiring specialized hardware to cope with the decryption and encryption load. With the growing adoption of cloud-based services, zero-trust concepts, and especially Secure Access Service Edge (SASE) solutions, deep packet inspection capabilities can be offloaded and offered more performantly, often embedded in a next-generation firewall (NGFW) or data loss prevention system (DLP). To successfully apply deep packet inspection, not only the DPI solution itself needs to be available, but only devices and endpoints that the organization manages can effectively be integrated.

Chrome, for example, does not even allow you to define custom root certificates to be trusted for HTTP/3 QUIC connection. While there is a theoretical workaround, it cannot be applied in practice and thus making it impossible to actively intercept traffic from the browser. Currently, there is no evidence if the status quo will remain or if the functionality will be added to give administrators more control over their clients behavior.

Recent developments in QUIC and the TLS1.3 ECH extension make it inherently more difficult – if not impossible - to intercept a client's traffic. DPI solutions cannot intercept the connection and reply with its answer (e.g., a custom certificate). In addition, connection tracking in QUIC is much harder for observers since valuable information about the connection state is encrypted, and UDP is a connectionless protocol. Today's DPI solutions do not support QUIC decryption and their providers (e.g., Palo Alto Networks or Zscaler) suggest blocking it so clients will fall back to TCP/TLS.

Losing the ability to inspect encrypted traffic has enormous implications for DLP tools. Leakage of confidential data will remain undetected and can hardly be prevented. The importance of data is more significant than ever and will continue with the rise of digitalization. New methods must be developed to help organizations to protect themselves from data breaches and ensure compliance with regulatory requirements like GDPR.

„ In the future, maintaining the security level based on middleboxes will be difficult.”

Rethinking traffic analysis

The challenges imposed by the evolvement of network protocols require rethinking of how we deal with network traffic analysis. Security providers need to invest and adapt their solutions to be able to deal with recent developments. An approach not requiring traffic decryption is among the most encouraging solutions.

Network detection and response: NDR solutions use advanced analytical techniques such as machine learning to detect suspicious network activity. They continuously monitor and analyze raw network traffic to generate a baseline of normal behavior. Patterns deviating from that baseline are detected and can alert security teams to inform them about potential threats in their environment. The anomaly detection is based on telemetry data collected from network devices. Telemetry data consists of usually visible data like host/port, packet sizes, or connection states. Depending on the data quality and the applied models, potential threats might be quickly recognized, but features like DLP cannot be replaced.

Considering the functionality of QUIC, most of the telemetry data required by NDR to be effective is encrypted. With very little information left, achieving good results in detecting potential threats will be difficult. Currently, NDR solutions primarily address the encrypted traffic analysis in TCP/TLS communication, where much more telemetry data is available.

Privacy-preserving DPI:
Another promising approach is a technique called privacy-preserving deep packet inspection. Until now, one was faced with either providing security through a traditional DPI – technically no longer possible with QUIC and TLS1.3 – or respecting privacy and compromising security. A privacy-preserving DPI system aims to put security and privacy aspects into symbiosis.

The system leverages key-homomorphic encryption allowing the analysis of DPI rules to happen on encrypted data directly. To achieve this, the sender (i.e., the client) must take a more active role. It has to provide encrypted tokens based on DPI rules for every TLS session to the DPI middlebox for analysis.

To apply any rules (e.g., search for malware fingerprints, data loss prevention), either the client or server needs to be in control of the intermediary. The respective endpoint is then actively involved in generating the encrypted tokens used for detection.

Currently, there are no marketable products available to provide this functionality. However, there's research ongoing with protocols like Blindbox, Pine, or P2DPI, which are promising. One limiting factor today is the computing power of the client required to create the tokens. Nevertheless, such a system could become a future-oriented approach to analyzing encrypted traffic.

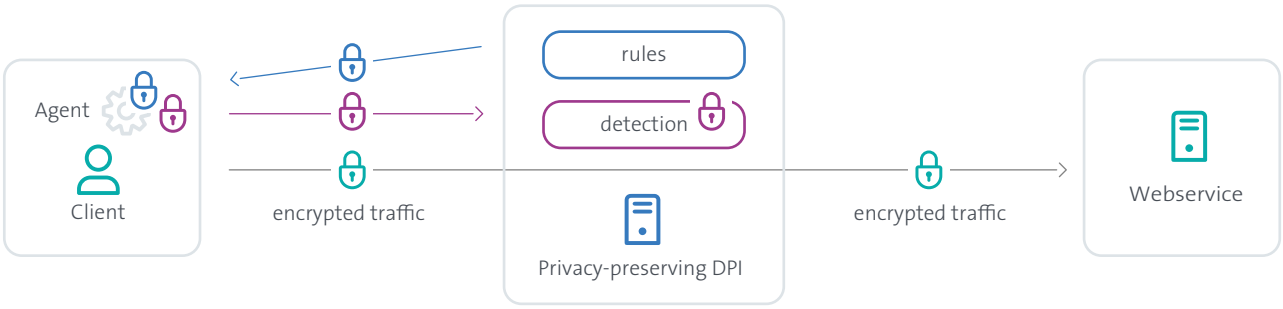

Figure 8 Example of a privacy-preserving DPI approach leveraging the client to generate on-demand encrypted detection rules

Network protection and perimeter security

New network protocols and features may introduce unknown attack vectors. Therefore, maintaining solid and up-to-date threat protection on the network is essential. Over the past decades, most investments in security devices have focused on addressing the challenges around TCP and application-level protocols depending on it.

Denial of service (DoS) systems must adapt to support QUIC as a new transport protocol and provide the required level of protection. Due to the very little information provided in cleartext, it is becoming more difficult for such systems to detect and mitigate such attacks. QUIC is particularly vulnerable to a QUIC reflection attack, where attackers spoof IP addresses to provoke the server to send back large replies in their initial packets. While the QUIC architecture tries to mitigate this scenario by design, in practice, such an attack can still be carried out[1].

Firewalls provide basic perimeter security in network environments ranging from a residential "home" network up to a large-scale enterprise network. Most traditional firewalls operate at the level of the transport protocol, meaning they either allow or block access based on the source or destination host and ports. Therefore, applying good firewall practices is essential to guarantee basic safety for your networks.

[1] https://www.cloudflare.com/en-gb/learning/ddos/what-is-a-quic-flood/

Security promises and implications of QUIC

“Stateful firewalls are dead!”

Stateful firewall and connection tracking: A stateful firewall – compared to a stateless one – not only evaluates the ruleset on every incoming packet but also adds some context to packets belonging to the same data stream. This is referred to as connection tracking, and it works especially well with TCP. Since TCP provides information about the connection state, a stateful firewall can maintain a state table on all connections. Concerning UDP, firewalls can track packets in a table but only guess if a returning packet with the same host/port combination belongs to the same connection.

Although QUIC carries the same information about its connection state, the fact that relevant fields are encrypted does not leave any other option for firewalls to track connections besides the basics inherited by UDP. To assess the impact of the lack of connection tracking, it is essential to understand the dynamics of connection management. Most client applications (e.g., a web browser) open random source ports for every outgoing connection. Whenever a request is passed through the firewall, it will automatically ensure that the matching response will be allowed on its way back. Once a connection is terminated, the state table will be updated, and the corresponding dynamic rules will be removed.

UDP hole punching

Since UDP can only be tracked packet by packet, stateful firewalls are prone to bypassing techniques. Among the most prominent is UDP hole punching. Since UDP packets carrying QUIC datagrams cannot be associated with a connection – as it is encrypted information – any packet from the server side containing the correct host/port combination can "punch a hole" in the firewall. While this technique holds true for all UDP traffic, the missing connection teardown for QUIC increases the risk. Additionally, UDP comes with a TTL (Time-to-live) expiry, which easily allows an attacker to keep the hole open long after the initial connection was shut down.

Such attacks show the indirect implications of enabling QUIC inside a network. With the mentioned UDP hole-punching attack, researchers proved how this situation could be exploited to open a reverse shell. QUIC's design principle prevents middleboxes from interpreting and interacting with the transport protocol parameters and has unwanted side effects on today's security devices, such as firewalls. Some potential approaches to mitigate the situation are being discussed in the IETF working group. An RFC (RFC 9312) on QUIC manageability has been published.

Tunnels and VPNs with QUIC?

Within the IETF, a dedicated working group called MASQUE (Multiplexed Application Substrate over QUIC Encryption) was formed in 2020 to unlock QUIC's potential for proxying. They are currently working on drafts to standardize proxy capabilities around QUIC. Due to its performant and privacy-aware design, QUIC is a natural candidate to be used as an end-to-end tunneling protocol.

With the introduction of nested tunneling, allowing to tunnel UDP datagrams within QUIC, MASQUE provides the application layer with the required building blocks to support efficient tunneling of TCP or UDP traffic. Furthermore, these blocks can be combined to build different deployment architectures catering to different needs or use cases.

One example of applying MASQUE is Apple's feature called iCloud Private Relay. Based on a nested tunnel architecture using multiple proxies, it minimizes the availability of connection metadata of an individual party, helping to increase a user's privacy even further.

If QUIC traffic is allowed, it is impossible to distinguish if it is a standard connection or someone trying to establish a VPN connection via nested tunneling. This again creates new attack surfaces for protected networks such as in enterprises.

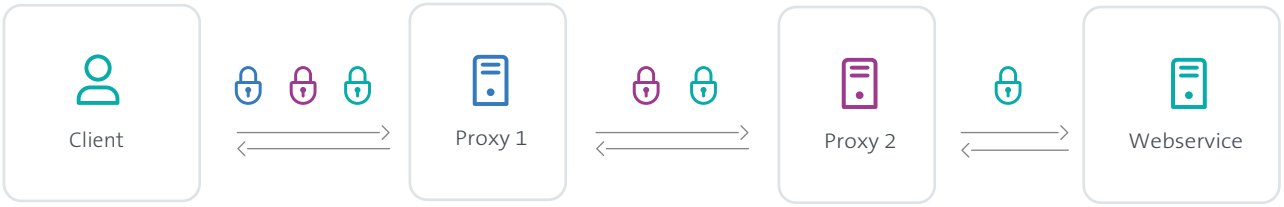

Figure 9 Example iCloud Private Relay with nested tunneling

QUIC and Privacy

As stated, QUIC aims to provide a privacy-enhanced communication protocol by maintaining the end-to-end principle where almost every bit of information is encrypted. Furthermore, QUIC uses advanced cryptographic algorithms, making it very difficult for an attacker to eavesdrop on the communication. With TLS1.3, a cryptographic cipher supporting Perfect Forward Secrecy (PFS) is mandatory. PFS secures each session with a unique session key. So, in case an attacker gets a private key or passphrase, decrypting the session is only possible if he gets the unique session key as well. Generally, it is safe to assume that QUIC provides good privacy protection against 3rd party observers. However, if in control of one of the endpoints, certain aspects of the protocol might be exploitable or used in unintended ways which could compromise a user's privacy (e.g., tracking mechanisms).

Website Fingerprinting

QUIC and TLS 1.3 provide a good level of privacy and protect their communication from eavesdropping. As mentioned, on one hand, this is great; on the other, some threat protection measures become difficult. However, also with QUIC, privacy is not perfect. Even though most information is encrypted, QUIC is still vulnerable to so-called website fingerprinting attacks (WFP). WFP is an attack based on statistical data or machine learning models. Attackers are trying to identify what websites are being browsed by the victim. Chinese and Swiss researchers found that QUIC is even more vulnerable to such attacks in some cases compared to TCP/TLS-based HTTPS. Alternatively, QUIC provides the necessary flexibility to introduce countermeasures to fingerprinting attacks like the QCSD research project.

Connection migration as a tracking mechanism

QUIC decouples the connection from IP and ports by introducing a dedicated connection ID, allowing the migration of existing connections across different networks. With this handy feature, it is possible to get a seamless experience when switching networks (e.g., wifi, cellular, etc.). Still, it also becomes a potential tracking mechanism for web servers to track their users.

The tracking mechanism can be compared to cookie-based web tracking. However, it is an integral part of the QUIC protocol which could be abused by content providers to track their users without requiring any consent. Using QUIC automatically allows tracking subsequent connections from a user. Continuous tracking is only possible if the browser/client application is not restarted. There are already proposals that could mitigate the situation, but it is currently questionable if this is in everybody's interest. Consequently, content providers could abuse the connection ID feature to profile and identify users across the web. As such, advertising companies could leverage it as a partial replacement for their "ad cookies."

Recommendation

It is essential to actively manage QUIC from a security perspective in an enterprise environment. Current best practices are to block and disable QUIC on the relevant infrastructure, which will result in a TCP/TLS fallback or entirely block QUIC-only services.

Blocking QUIC and HTTP3

Some firewall vendors recognize QUIC traffic and can be configured to block corresponding connections. For firewalls which do not yet detect QUIC, it is suggested to block UDP traffic in general and allow only specific trusted services if required. HTTP/3 is usually performed via UDP port 443, but the port number is not part of the standardization and cannot relied upon.

Disable QUIC and HTTP/3 on clients

Web browsers can be configured to allow or deny QUIC and HTTP/3 connections. Examples of how to change the configuration can be found in the corresponding browser documentation[2]. Devices that are managed through an MDM (Mobile Device Management) solution or similar, policies disabling QUIC can be rolled out[3,4].

[2] https://www.watchguard.com/help/docs/help-center/en-us/Content/en-US/Endpoint-Security/manage-settings/disable-http-3-protocol-for-web-access-control.html?TocPath=Troubleshooting%7C_____193

[3] https://knowledge.broadcom.com/external/article/173741/http3-aka-quic-protocol-bypasses-cloud.html

[4] https://letsconfigmgr.com/memac-managing-google-chrome/

Conclusion & outlook

It is essential to actively manage QUIC from a security perspective in an enterprise environment. Current best practices are to block and disable QUIC on the relevant infrastructure, which will result in a TCP/TLS fallback or entirely block QUIC-only services.

Conclusion & future outlook

The adoption of QUIC across the internet will further grow. The impact on the quality of experience for large content providers excels in terms of end-to-end control and flexibility in new feature implementations. The big players strengthen their dominance even more. With QUIC, they no longer only control the endpoints but also the communication in between. This might introduce new challenges and risks where additional regulation might be required.

While most performance benefits primarily manifest in mobile usage. QUIC adoption continues to expand in other use cases. Still, it has some downsides that need to be evaluated carefully before opting in on QUIC. The development of QUIC went pretty fast, and many supporting technologies have not kept up. Overall, QUIC is a good development, trying to bring the best of many protocols together and provide users with a better, more secure, and private experience.

The Internet is still optimized for TCP. TCP delivery has existed for decades, and over the years, applications have been optimized by building off-load capabilities in software (e.g., operating systems) and hardware (e.g., intelligent NICs). None of this exists for QUIC. It is not baked into operating systems and firmware yet. The available implementations are at the user space level, making them easier to roll out but less efficient. This results in increased CPU utilization of 2x - 3x compared to TCP. Once QUIC is widely adopted, such optimization is expected to be implemented.

Security and network vendors suggest disabling QUIC for now.

Conclusion & future outlook

The majority of security products today cannot properly deal with QUIC. By design, QUIC tries to prevent protocol ossification by respecting the end-to-end principle. As stated, this has drawbacks for most middlebox devices. For example, firewalls, load balancers, or deep packet inspection systems rely on patterns and metadata no longer available with QUIC.

While the QUIC working group acknowledges some of the aspects in their recent RFC 9312 (Manageability of the QUIC Transport Protocol), there needs to be more evidence provided by security and network product suppliers to address the challenges around QUIC actively. As a result, we would expect to see more engagement by these vendors to shape the QUIC standard's future development actively.

Some products have announced QUIC and HTTP/3 support, like Cloudflare Zero Trust or Fortinet. Both mention that – in theory - their products support HTTP/3 inspection. However, having a closer look at their documentation, they both acknowledge that due to limitations in Chrome, Safari, or Firefox, inspection is likely not working, and thus an automatic fallback will happen.

For the time being, it is best to follow the best practice guidance and disable QUIC on your enterprise networks. As a result, connections will automatically be downgraded to TCP/TLS, supported by the various products and vendors. However, the first QUIC-only services are appearing (e.g., Apple iCloud Private Relay), but well-established services keep offering alternative communication based on TCP/TLS, and it is expected that both will co-exist in the upcoming years. Nonetheless, the status quo should not be accepted as the final solution. Switching to a QUIC-only setup for any service can technically happen overnight.

If corporate organizations cannot keep up with the developments in the long run, we will likely end up with a two-class internet. The QUIC-only based consumer services (e.g., Youtube, Instagram, etc.) and a set of services commonly used by enterprises are still based on TCP connections.

Moving from a traditional perimeter-based security approach towards a Zero-Trust model can help organizations mitigate some of the mentioned security implications. Instead of relying on the firewall functions, implementing a Zero-Trust strategy to authenticate and manage access from users to enterprise resources will regain complete control. In addition, policy management and enforcement are largely decoupled from the underlying network protocols, allowing fine-grained access control and auditing.

Encryption, Inspection, and data protection

The rise of encrypted network traffic is a significant development. Providing better security and privacy in communication is desirable. However, the challenges imposed by "almost-anything" encrypted communication are not limited to QUIC. Especially with the evolvement of TLS 1.3, TLS extensions, and encrypted DNS, it will become much more challenging for organizations to manage encrypted traffic. Inspection capabilities – primarily used in the context of malware and threat protection, and data loss prevention – will stop working.

New techniques and products must be developed to cope with these changes. Solutions not requiring decryption of traffic are among the promising. Network detection and response products today already leverage, for example, machine learning to analyze encrypted traffic. They already provide good results in the context of threat detection (e.g., command and control traffic).

On the other side, it is clear that passive observation and pattern recognition will not be sufficient for data loss prevention (DLP) systems. As the ongoing research around privacy-preserving DPI suggests, building new strategies requires advanced encryption techniques such as homomorphic encryption and, even more importantly, access and control over one of the endpoints, typically the client initiating a connection.

Endpoint protection and device management

Network security remains essential, but being able to cope with new challenges, good endpoint security, including device management, is unavoidable. Client devices suffer from limited resource availability, which cannot be easily scaled out. In addition, current QUIC implementations are consuming more resources to perform the same tasks than TCP. Introducing more security functionality to the clients will further compound the situation.

The lack of standardized interfaces between applications and operating systems makes it challenging to find a generic approach to enforce security with QUIC. Despite prohibiting the use of UDP for applications on a client device, configuration settings need to be evaluated on a per-application basis.

The relevance of purely network-based security measures is weakened, and it becomes even more important to embed endpoint security into the cybersecurity strategy.

Browsers can be configured to allow or deny QUIC-based traffic in general. Unfortunately, there is no fine-grained option (e.g., allowlist specific trusted HTTP/3 endpoints) available. When it comes to certificate management, the situation is even worse. With the introduction of QUIC, popular browsers do not respect any custom trusted root certificates to be used for QUIC connections. This is why proxies, secure web gateways, or deep packet inspection systems cannot be put into the communication path.

Likewise, techniques like a sandboxed execution environment or embedding homomorphic encryption capabilities into an application will be required to pave the road for new endpoint protection and data leak prevention strategies being implemented.

This situation puts lots of responsibility on application developers to behave responsibly and endpoint managers to actively protect their assets. In the future, administrators will have to invest more in endpoint protection with solutions like Extendend Detection and Response (XDR) to regain security capabilities.

The QUIC protocol is still relatively new. Its development will further evolve and will enable new use cases for applications but likely also create opportunities to better secure communication in the internet.