Task 6.3HD

Task 6.3HD conducting research on QUIC and prepare a tutorial

Introduction

The internet environment is always changing, requiring communication protocols that are quicker and more dependable. A new transport protocol called QUIC (Quick UDP Internet Connections) is introduced with the goal of addressing TCP's (Transmission Control Protocol) shortcomings in the context of the current web. In-depth analysis of QUIC's features, benefits, security implications, and scalability is provided in this research. It also looks at utilizing the network protocol analyzer Wireshark to examine QUIC traffic.

Background

TCP, the internet's workhorse nowadays, provides dependable in-order packet delivery. But TCP can cause delay problems since it requires a three-way handshake and can be blocked by head-of-line (HoL) traffic, especially when real-time applications and crowded networks are involved.

What is QUIC?

Google created QUIC, an UDP-based transport protocol that improves upon TCP's drawbacks. QUIC uses UDP to enable faster, connectionless data transport as opposed to TCP. On the other hand, QUIC simplifies communication by integrating features like encryption, dependability measures, and congestion control directly into the protocol, building upon UDP.

Functionalities of QUIC

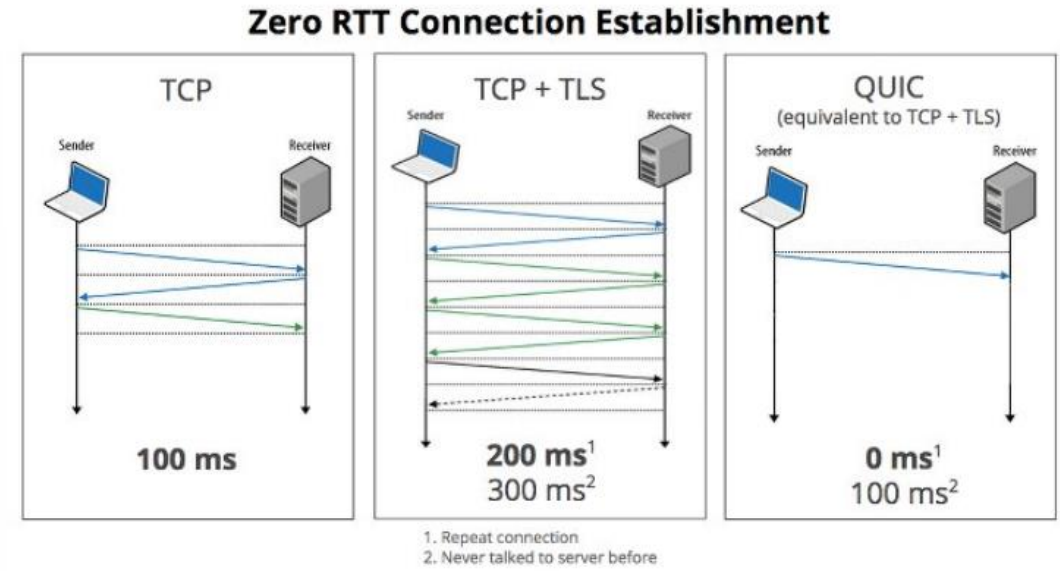
- **Multiplexing:** QUIC enables the simultaneous transmission of several data streams (requests and responses) over a single connection. By doing this, HoL (Head-of-line) blocking is removed, which enhances efficiency for apps that frequently transfer little amounts of data.

- **Reduced Latency:** QUIC connects more quickly and facilitates faster data flow by doing away with the three-way handshake that TCP requires.

- **Better Congestion Control:** To achieve the best data transfer speeds, QUIC uses dynamic congestion control algorithms that adjust to the network's circumstances.

- **Connection-Oriented Approach:** QUIC is based on UDP, but it functions similarly to TCP in terms of connections, guaranteeing dependable data transfer through features like acknowledgement packets and retransmissions.

- **Integral Security:** TLS 1.3 is used by QUIC to provide secure communication, encrypting data streams and guarding against manipulation and eavesdropping.



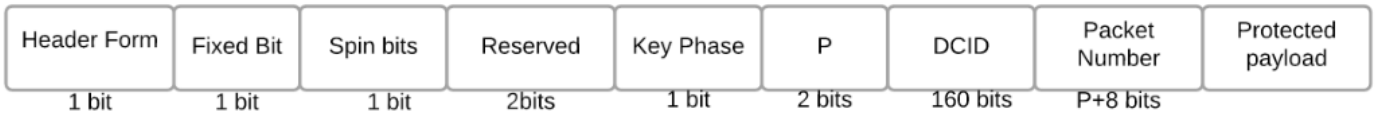
Frame Formats in QUIC

QUIC establishes a unique frame format for data interchange. These frames contain data regarding:

- **Streams:** For multiplexing and autonomous management, data is divided into streams.
- **Acknowledgments:** If more transmission is required, acknowledgements verify that the data packet was successfully received.
- **Connection management:** Establishing, ending, and controlling the flow of connections are all handled by frames.
- **Application Data:** The real application data being transferred is contained in these frames.

Long Header

Short Header



Frames with a QUIC header, which can be short or long, make up QUIC packets. The short header is used after the initial connection has been made, but the long header is used before that. Twelve octets make up the short header and twenty octets make up the long header.

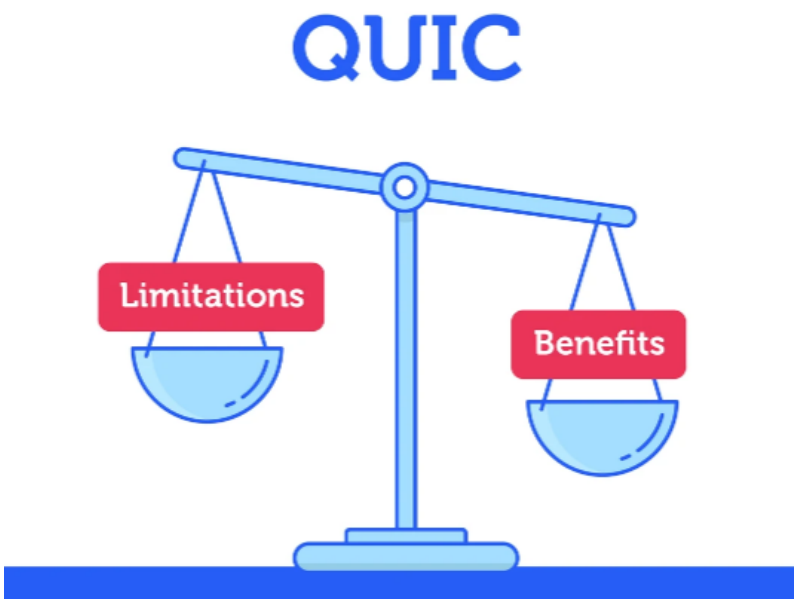
The following are additional elements of a QUIC packet:

- **Header Format (HF):** Determines the type of header
- **CID (Connection Identifier):** prevents packets for a QUIC connection from being sent to the incorrect endpoint.
- **Packet Number:** Defines the cryptographic nonce used to safeguard packets.

Additionally, PADDING frames—which have no semantic significance but can be used to enlarge a packet—can be included in QUIC packets.

Advantages of QUIC over TCP

- **Decreased Latency:** Lower total latency is the result of quicker connection formation and better handling of packet loss.
- **Better Congestion Control:** Data transport efficiency is maximized through dynamic adaptability to network conditions.
- **Security**: TLS 1.3's built-in encryption improves communication security.
- **Multiplexing:** Makes it possible to handle multiple concurrent data streams effectively.



Disadvantages of QUIC

- **Complexity:** May cause compatibility problems because QUIC communication requires server and client support.

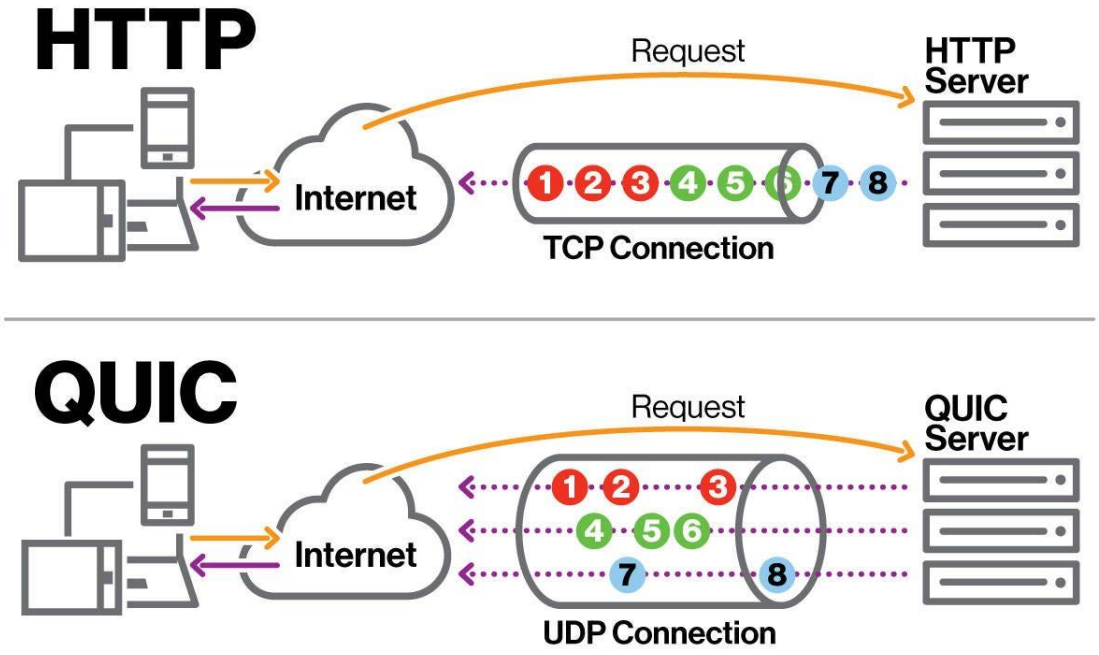
- **New Technology:** More recent technology is still being developed, but it has the potential to alter in the future and continues to be improved.

- **No Packet inspection:** No examination of the packet Because QUIC traffic cannot be decrypted by network firewalls in order to inspect packets, potentially harmful traffic may enter the network undetected by conventional security measures. Due to the assumption that QUIC packets on ports 80 (web server) and 443 (TLS) could contain malware, security vendors like Cisco and Palo Alto Networks regularly block them, forcing clients to revert to using HTTP/2 and TCP protocols.

- **Fingerprinting could be easier:** The possibility that malicious actors could intercept Internet users' packets as they travel between websites and utilize the observable packets to create unique patterns that link to particular websites is a recent worry that is currently being looked into. Web fingerprinting is what this is known as, and it appears that early in the connection traffic phase, TCP+HTTPS are more resilient to it.

Network Applications using QUIC

- **HTTP/3:** The most recent iteration of HTTP makes use of QUIC to speed up web browsing, which is especially apparent when fetching images or other small amounts of data often.
- **Streaming Services:** QUIC's lower latency allows for smoother video streaming on websites like YouTube.
- **Real-time Applications:** QUIC's decreased latency and faster data transfer can improve the performance of applications like online gaming and video conferencing.



Security Threats and Mitigation Strategies in QUIC

Even with encryption, QUIC can still be vulnerable to several security risks:

- **Attacks known as denial-of-service (DoS):** Because QUIC relies heavily on connections, it is prone to denial-of-service assaults. The key components of mitigation are resource protection and server-side rate limitation.
- **Middlebox Problems:** Potential disruptions may arise from firewalls and intrusion detection systems (IDS) that are not optimized for QUIC. For a seamless integration, compatibility tests and configuration adjustments are required.

Scalability of QUIC

QUIC enables servers to successfully manage many client connections by providing the efficient management of multiple connections through multiplexing. This enhances web applications with large traffic's scalability.

Analyzing QUIC with Wireshark

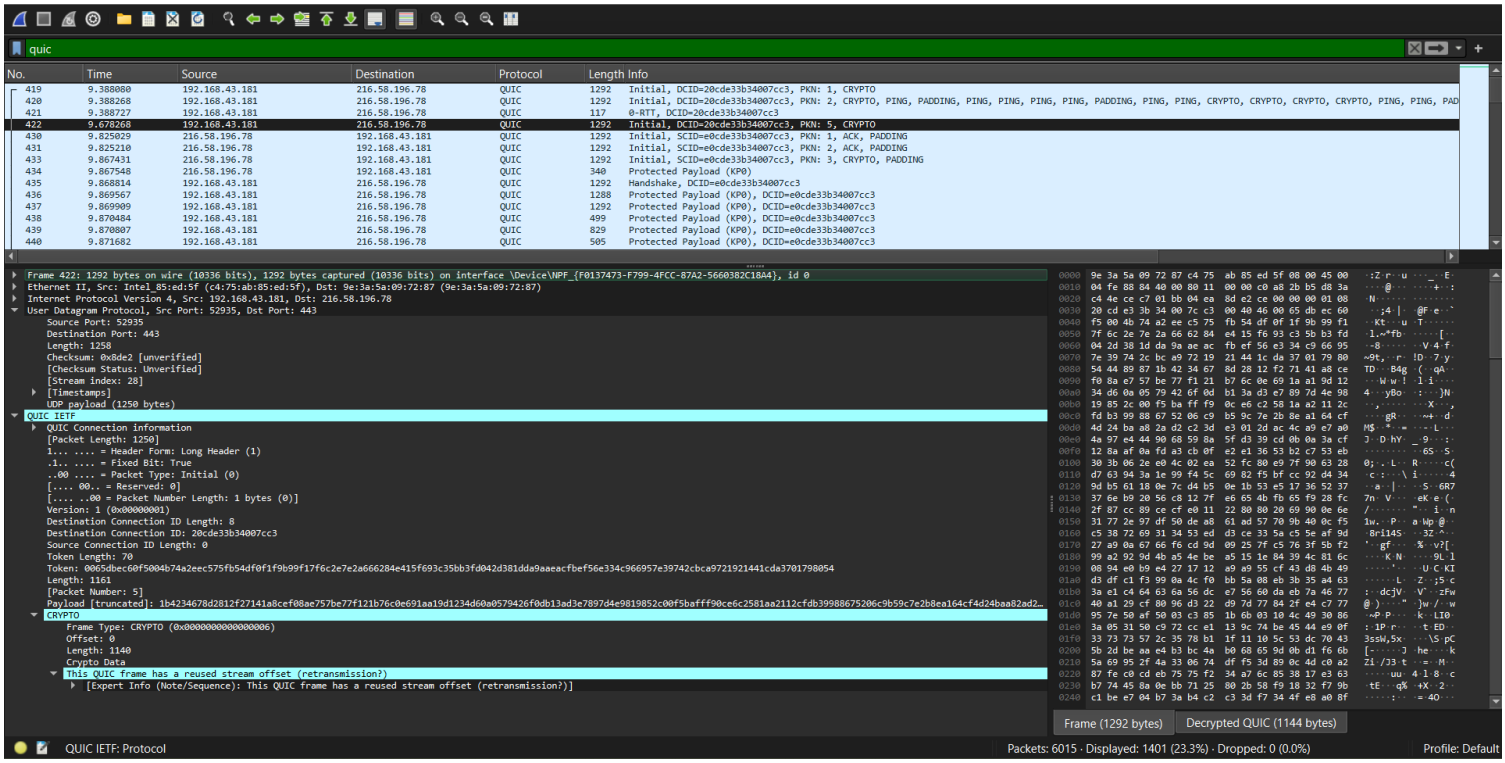
First Packet Transmission: Application data and connection establishment data are frequently carried in parallel with the first QUIC packet. When opposed to TCP, where separate packets are utilized for data transport and handshake, this lowers overall latency.

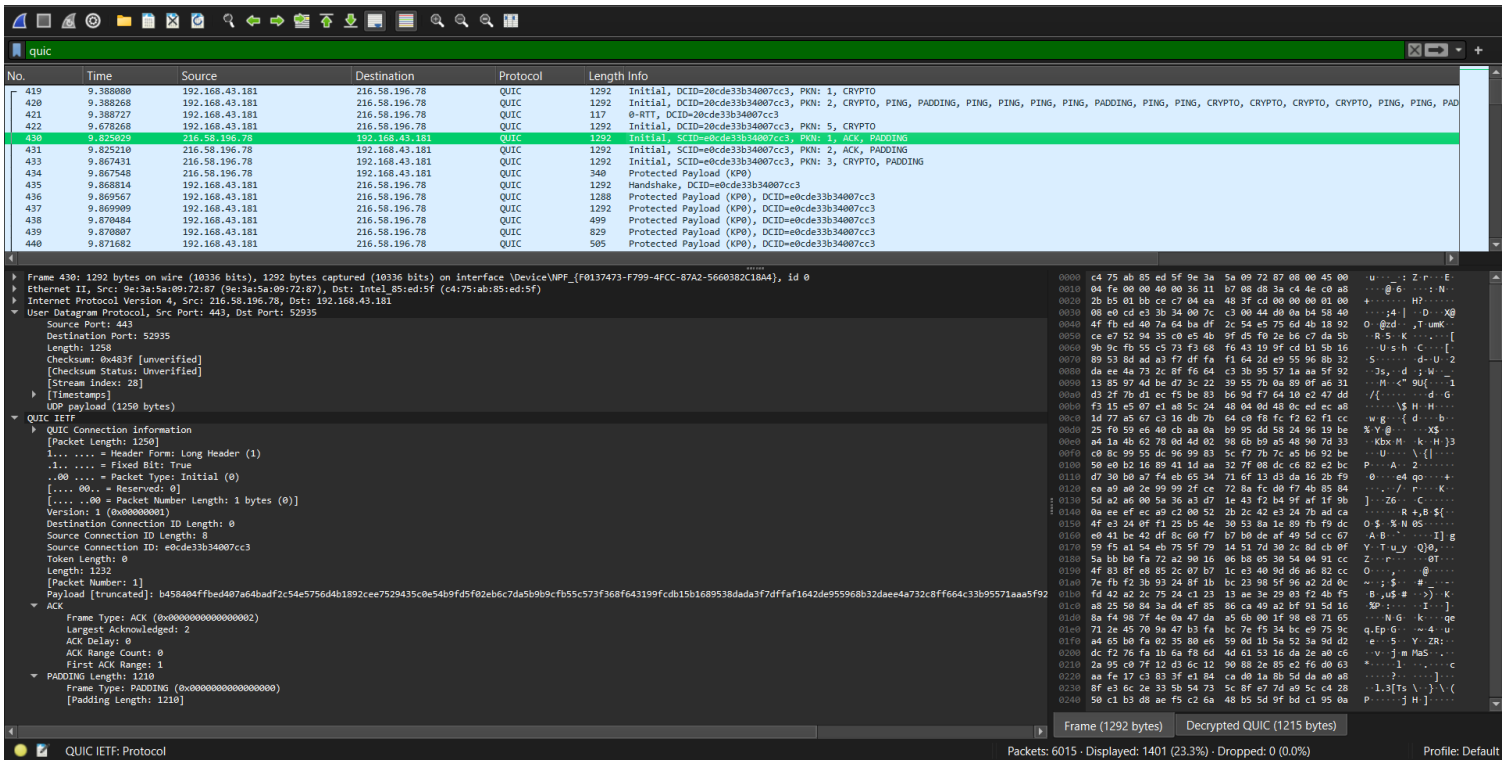
QUIC Specific Parameters: To distinguish QUIC traffic from other protocols, you will encounter parameters such as QUIC version, stream IDs (identifiers for specific data streams within the connection), and connection IDs (a unique identifier for the QUIC connection). You can learn more about the behavior and structure of the QUIC communication by examining these parameters.

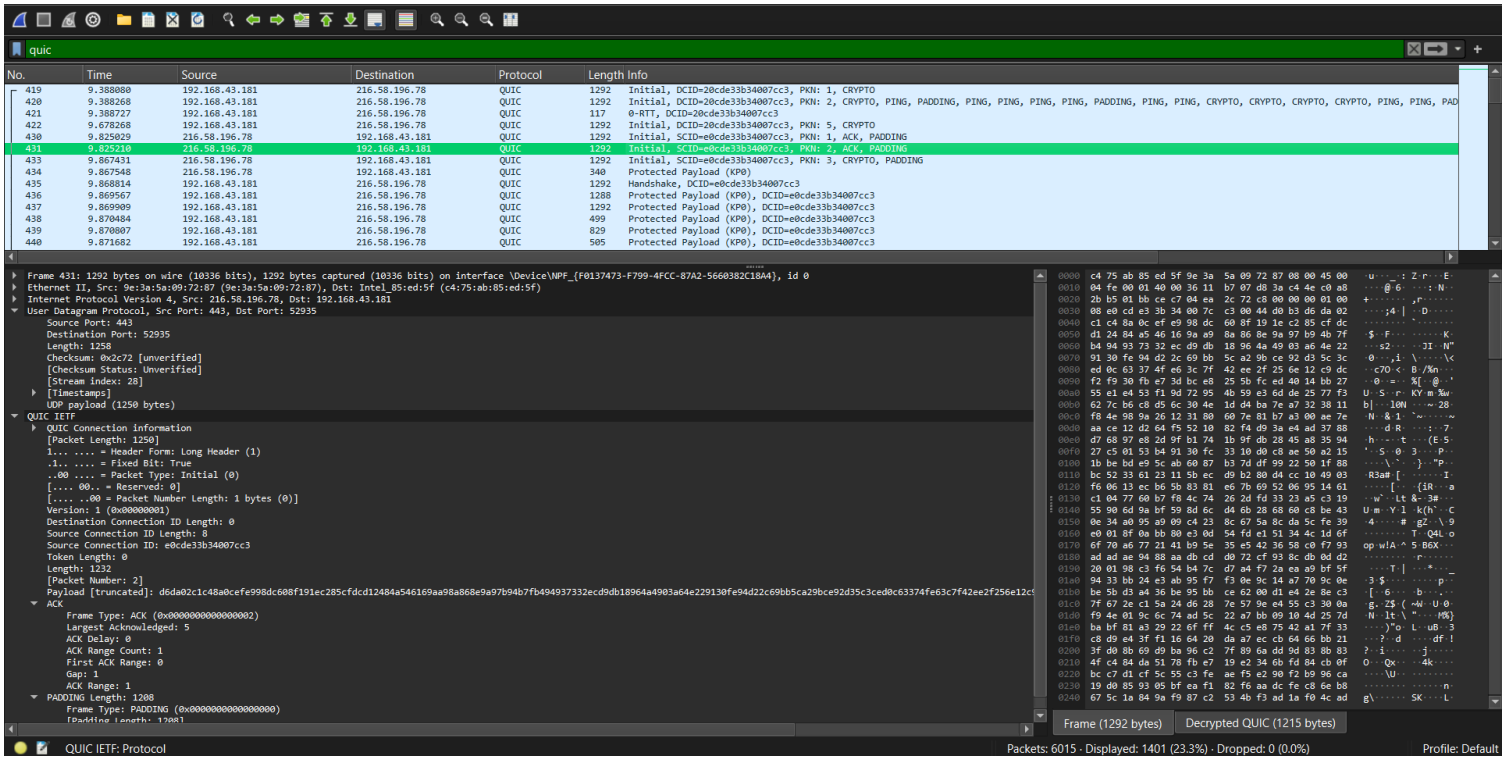
QUIC encrypts the payload of most messages after the initial handshake for security. Wireshark, by default, cannot decrypt this data without the necessary keys.

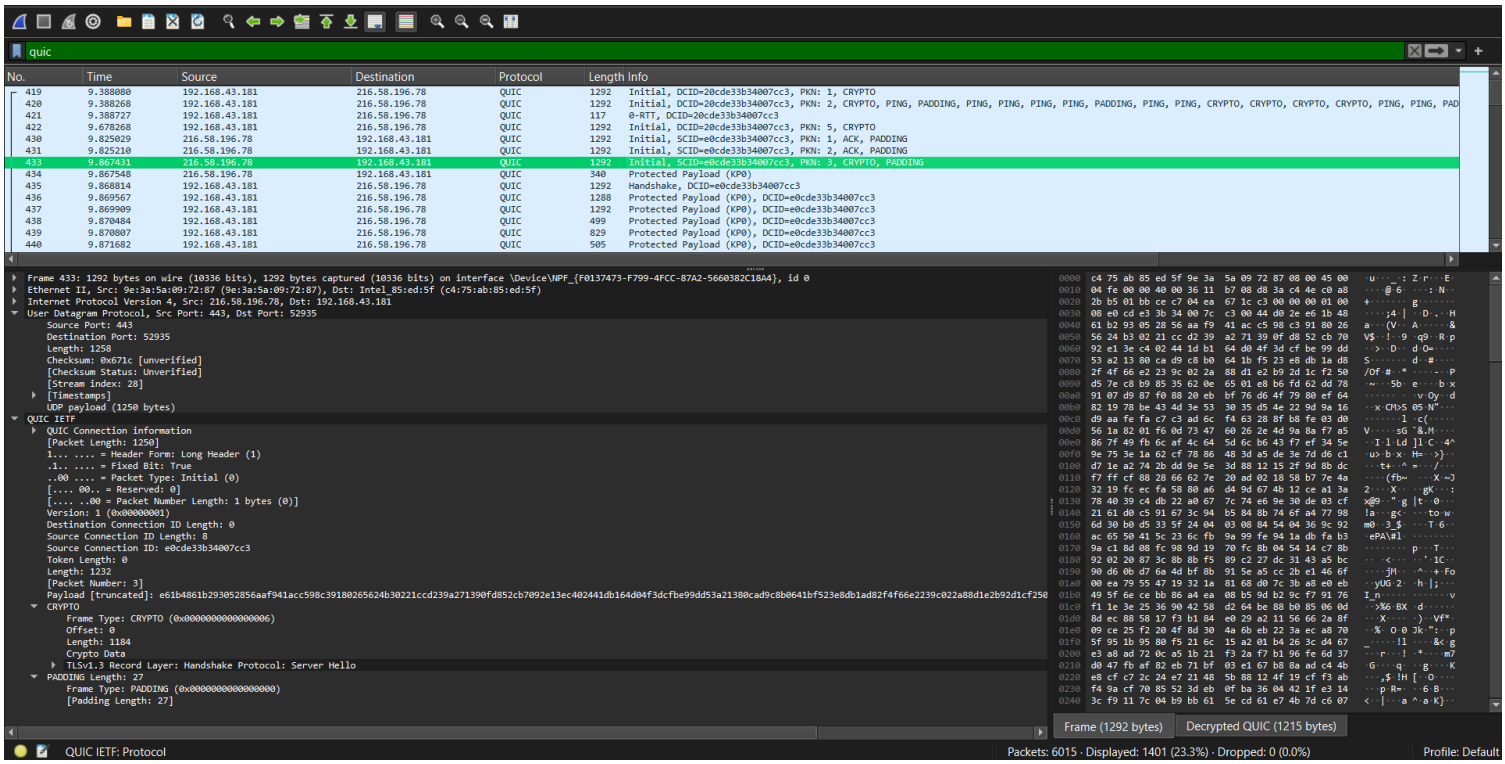
The initial packet typically contains some unencrypted information like connection IDs and possibly some transport parameters."

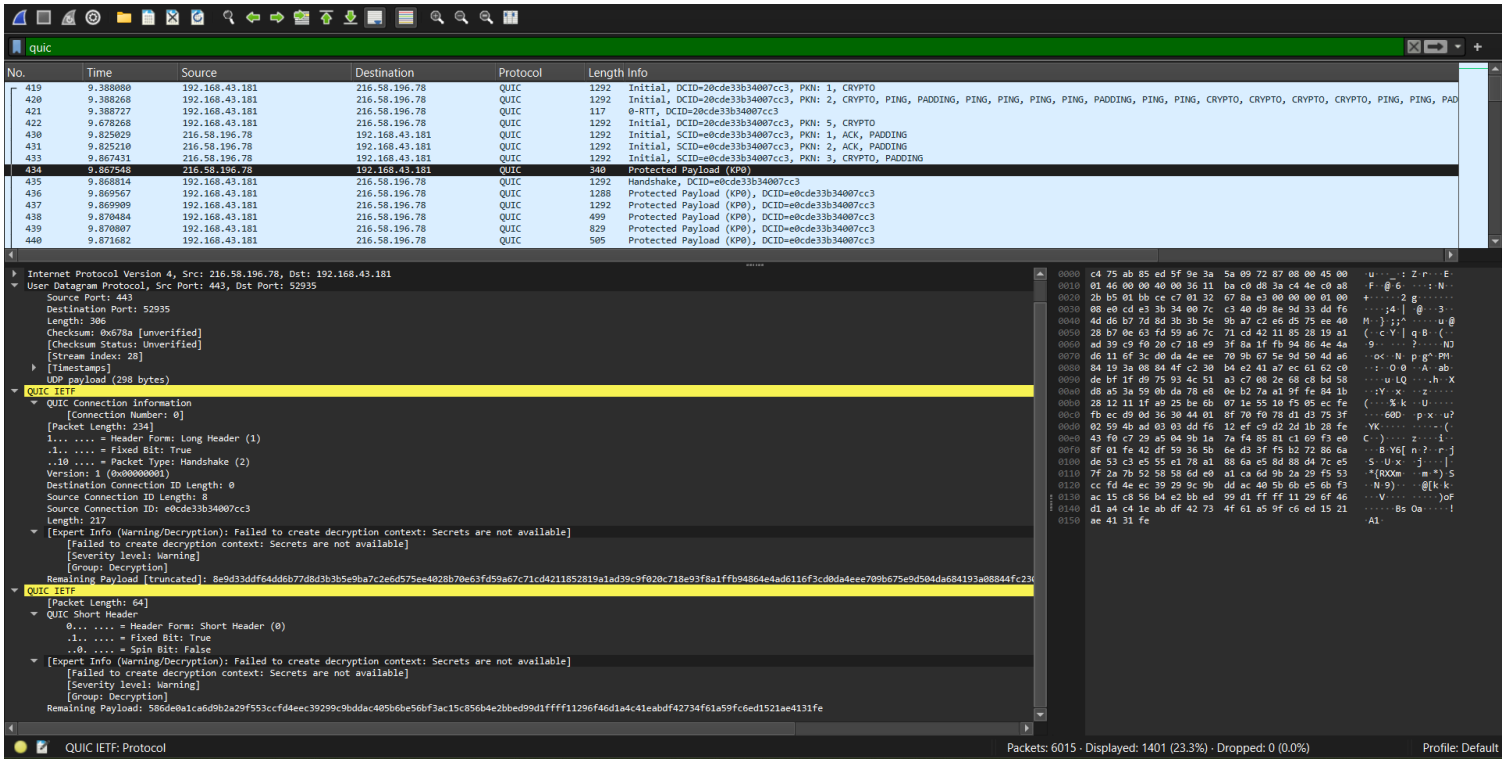
The "protected messages" you see likely refer to QUIC packets after the initial handshake. These packets are encrypted and will appear unreadable in Wireshark.




quic
No. Time Source Destination Protocol Length Info
419 9.388080 192.168.43.181 216.58.196.78 QUIC 1292 Initial, DCID=20cde33b34007cc3, PKN: 1, CRYPTO
420 9.388268 192.168.43.181 216.58.196.78 QUIC 1292 Initial, DCID=20cde33b34007cc3, PKN: 2, CRYPTO, PING, PADDING, PING, PING, PING, PING, PADDING, PING, PING, CRYPTO, CRYPTO, CRYPTO, CRYPTO, PING, PING, PAD
421 9.388727 192.168.43.181 216.58.196.78 QUIC 117 0-RTT, DCID=20cde33b34007cc3
422 9.678268 192.168.43.181 216.58.196.78 QUIC 1292 Initial, DCID=20cde33b34007cc3, PKN: 5, CRYPTO
430 9.825029 216.58.196.78 192.168.43.181 QUIC 1292 Initial, SCID=e0cde33b34007cc3, PKN: 1, ACK, PADDING
431 9.825210 216.58.196.78 192.168.43.181 QUIC 1292 Initial, SCID=e0cde33b34007cc3, PKN: 2, ACK, PADDING
433 9.867431 216.58.196.78 192.168.43.181 QUIC 1292 Initial, SCID=e0cde33b34007cc3, PKN: 3, CRYPTO, PADDING
434 9.867548 216.58.196.78 192.168.43.181 QUIC 340 Protected Payload (KP0)
435 9.868814 192.168.43.181 216.58.196.78 QUIC 1292 Handshake, DCID=e0cde33b34007cc3
436 9.869567 192.168.43.181 216.58.196.78 QUIC 1288 Protected Payload (KP0), DCID=e0cde33b34007cc3
437 9.869909 192.168.43.181 216.58.196.78 QUIC 1292 Protected Payload (KP0), DCID=e0cde33b34007cc3
438 9.870484 192.168.43.181 216.58.196.78 QUIC 499 Protected Payload (KP0), DCID=e0cde33b34007cc3
439 9.870807 192.168.43.181 216.58.196.78 QUIC 829 Protected Payload (KP0), DCID=e0cde33b34007cc3
440 9.871682 192.168.43.181 216.58.196.78 QUIC 505 Protected Payload (KP0), DCID=e0cde33b34007cc3
Frame 430: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\NPF_{F0137473-F799-4FCC-87A2-5660382C184A}, id 0
Ethernet II, Src: 9e:3a:5a:09:72:87 (9e:3a:5a:09:72:87), Dst: Intel_85:ed:5f (c4:75:ab:85:ed:5f)
Internet Protocol Version 4, Src: 216.58.196.78, Dst: 192.168.43.181
User Datagram Protocol, Src Port: 443, Dst Port: 52935
Source Port: 443
Destination Port: 52935
Length: 1258
Checksum: 0x483f [unverified]
[Checksum Status: Unverified]
[Stream index: 28]
[Timestamps]
UDP payload (1250 bytes)
QUIC IETF
QUIC Connection information
[Packet Length: 1250]
1... .... = Header Form: Long Header (1)
.1.. .... = Fixed Bit: True
..00 .... = Packet Type: Initial (0)
[.... 00.. = Reserved: 0]
[.... ..00 = Packet Number Length: 1 bytes (0)]
Version: 1 (0x00000001)
Destination Connection ID Length: 0
Source Connection ID Length: 8
Source Connection ID: e0cde33b34007cc3
Token Length: 0
Length: 1232
[Packet Number: 1]
Payload [truncated]: b458404ffbed407a64badf2c54e5756d4b1892cee7529435c0e54b9fd5f02eb6c7da5b9b9cfb55c573f368f643199fcdb15b1689538dada3f7dffaf1642de955968b32daee4a732c8ff664c33b95571aaa5f92
ACK
Frame Type: ACK (0x0000000000000002)
Largest Acknowledged: 2
ACK Delay: 0
ACK Range Count: 0
First ACK Range: 1
PADDING Length: 1210
Frame Type: PADDING (0x0000000000000000)
[Padding Length: 1210]
0000 c4 75 ab 85 ed 5f 9e 3a 5a 09 72 87 08 00 45 00
0010 04 fe 00 00 40 00 36 11 b7 08 d8 3a c4 4e c0 a8
0020 2b b5 01 bb ce c7 04 ea 48 3f cd 00 00 00 01 00
0030 08 e0 cd e3 3b 34 00 7c c3 00 44 d0 0a b4 58 40
0040 4f fb ed 40 7a 64 ba df 2c 54 e5 75 6d 4b 18 92
0050 ce e7 52 94 35 c0 e5 4b 9f d5 f0 2e b6 c7 da 5b
0060 9b 9c fb 55 c5 73 f3 68 f6 43 19 9f cd b1 5b 16
0070 89 53 8d ad a3 f7 df fa f1 64 2d e9 55 96 8b 32
0080 da ee 4a 73 2c 8f f6 64 c3 3b 95 57 1a aa 5f 92
0090 13 85 97 4d be d7 3c 22 39 55 7b 0a 89 0f a6 31
00a0 d3 2f 7b d1 ec f5 be 83 b6 9d f7 64 10 e2 47 dd
00b0 f3 15 e5 07 e1 a8 5c 24 48 04 0d 48 0c ed ec a8
00c0 1d 77 a5 67 c3 16 db 7b 64 c0 f8 fc f2 62 f1 cc
00d0 25 f0 59 e6 40 cb aa 0a b9 95 dd 58 24 96 19 be
00e0 a4 1a 4b 62 78 0d 4d 02 98 6b b9 a5 48 90 7d 33
00f0 c0 8c 99 55 dc 96 99 83 5c f7 7b 7c a5 b6 92 be
0100 50 e0 b2 16 89 41 1d aa 32 7f 08 dc c6 82 e2 bc
0110 d7 30 b0 a7 f4 eb 65 34 71 6f 13 d3 da 16 2b f9
0120 ea a9 a0 2e 99 99 2f ce 72 8a fc d0 f7 4b 85 84
0130 5d a2 a6 00 5a 36 a3 d7 1e 43 f2 b4 9f af 1f 9b
0140 0a ee ef ec a9 c2 00 52 2b 2c 42 e3 24 7b ad ca
0150 4f e3 24 0f f1 25 b5 4e 30 53 0a 1e 89 fb f9 dc
0160 e0 41 be 42 df 8c 60 f7 b7 b0 de af 49 5d cc 67
0170 59 f5 a1 54 eb 75 5f 79 14 51 7d 30 2c 8d cb 0f
0180 5a bb b0 fa 72 a2 90 16 06 b8 05 30 54 04 91 cc
0190 4f 83 8f e8 85 2c 07 b7 1c e3 40 9d d6 a6 82 cc
01a0 7e fb f2 3b 93 24 8f 1b bc 23 98 5f 96 a2 2d 0c
01b0 fd 42 a2 2c 75 24 c1 23 13 ae 3e 29 03 f2 4b f5
01c0 a8 25 50 84 3a d4 ef 85 86 ca 49 a2 bf 91 5d 16
01d0 8a f4 98 7f 4e 0a 47 da a5 6b 00 1f 98 e8 71 65
01e0 71 2e 45 70 9a 47 b3 fa bc 7e f5 34 bc e9 75 9c
01f0 a4 65 b0 fa 02 35 80 e6 59 0d 1b 5a 52 3a 9d d2
0200 dc f2 76 fa 1b 6a f8 6d 4d 61 53 16 da 2e a0 c6
0210 2a 95 c0 7f 12 d3 6c 12 90 88 2e 85 e2 f6 d0 63
0220 aa fe 17 c3 83 3f e1 84 ca d0 1a 8b 5d da a0 a8
0230 8f e3 6c 2e 33 5b 54 73 5c 8f e7 7d a9 5c c4 28
0240 50 c1 b3 d8 ae f5 c2 6a 48 b5 5d 9f bd c1 95 0a
Frame (1292 bytes)
Decrypted QUIC (1215 bytes)
QUIC IETF: Protocol
Packets: 6015 · Displayed: 1401 (23.3%) · Dropped: 0 (0.0%)
Profile: Default


quic
No. Time Source Destination Protocol Length Info
419 9.388080 192.168.43.181 216.58.196.78 QUIC 1292 Initial, DCID=20cde33b34007cc3, PKN: 1, CRYPTO
420 9.388268 192.168.43.181 216.58.196.78 QUIC 1292 Initial, DCID=20cde33b34007cc3, PKN: 2, CRYPTO, PING, PADDING, PING, PING, PING, PING, PADDING, PING, PING, CRYPTO, CRYPTO, CRYPTO, CRYPTO, PING, PING, PAD
421 9.388727 192.168.43.181 216.58.196.78 QUIC 117 0-RTT, DCID=20cde33b34007cc3
422 9.678268 192.168.43.181 216.58.196.78 QUIC 1292 Initial, DCID=20cde33b34007cc3, PKN: 5, CRYPTO
430 9.825029 216.58.196.78 192.168.43.181 QUIC 1292 Initial, SCID=e0cde33b34007cc3, PKN: 1, ACK, PADDING
431 9.825210 216.58.196.78 192.168.43.181 QUIC 1292 Initial, SCID=e0cde33b34007cc3, PKN: 2, ACK, PADDING
433 9.867431 216.58.196.78 192.168.43.181 QUIC 1292 Initial, SCID=e0cde33b34007cc3, PKN: 3, CRYPTO, PADDING
434 9.867548 216.58.196.78 192.168.43.181 QUIC 340 Protected Payload (KP0)
435 9.868814 192.168.43.181 216.58.196.78 QUIC 1292 Handshake, DCID=e0cde33b34007cc3
436 9.869567 192.168.43.181 216.58.196.78 QUIC 1288 Protected Payload (KP0), DCID=e0cde33b34007cc3
437 9.869909 192.168.43.181 216.58.196.78 QUIC 1292 Protected Payload (KP0), DCID=e0cde33b34007cc3
438 9.870484 192.168.43.181 216.58.196.78 QUIC 499 Protected Payload (KP0), DCID=e0cde33b34007cc3
439 9.870807 192.168.43.181 216.58.196.78 QUIC 829 Protected Payload (KP0), DCID=e0cde33b34007cc3
440 9.871682 192.168.43.181 216.58.196.78 QUIC 505 Protected Payload (KP0), DCID=e0cde33b34007cc3
Frame 431: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\NPF_{F0137473-F799-4FCC-87A2-5660382C184A}, id 0
Ethernet II, Src: 9e:3a:5a:09:72:87 (9e:3a:5a:09:72:87), Dst: Intel_85:ed:5f (c4:75:ab:85:ed:5f)
Internet Protocol Version 4, Src: 216.58.196.78, Dst: 192.168.43.181
User Datagram Protocol, Src Port: 443, Dst Port: 52935
Source Port: 443
Destination Port: 52935
Length: 1258
Checksum: 0x2c72 [unverified]
[Checksum Status: Unverified]
[Stream index: 28]
[Timestamps]
UDP payload (1250 bytes)
QUIC IETF
QUIC Connection information
[Packet Length: 1250]
1... .... = Header Form: Long Header (1)
.1.. .... = Fixed Bit: True
..00 .... = Packet Type: Initial (0)
[.... 00.. = Reserved: 0]
[.... ..00 = Packet Number Length: 1 bytes (0)]
Version: 1 (0x00000001)
Destination Connection ID Length: 0
Source Connection ID Length: 8
Source Connection ID: e0cde33b34007cc3
Token Length: 0
Length: 1232
[Packet Number: 2]
Payload [truncated]: d6da02c1c48a0cefe998dc608f191ec285cfdcd12484a546169aa98a868e9a97b94b7fb494937332ecd9db18964a4903a64e229130fe94d22c69bb5ca29bce92d35c3ced0c63374fe63c7f42ee2f256e12c
ACK
Frame Type: ACK (0x0000000000000002)
Largest Acknowledged: 5
ACK Delay: 0
ACK Range Count: 1
First ACK Range: 0
Gap: 1
ACK Range: 1
PADDING Length: 1208
Frame Type: PADDING (0x0000000000000000)
[Padding Length: 1208]
0000 c4 75 ab 85 ed 5f 9e 3a 5a 09 72 87 08 00 45 00
0010 04 fe 00 01 40 00 36 11 b7 07 d8 3a c4 4e c0 a8
0020 2b b5 01 bb ce c7 04 ea 2c 72 c8 00 00 00 01 00
0030 08 e0 cd e3 3b 34 00 7c c3 00 44 d0 b3 d6 da 02
0040 c1 c4 8a 0c ef e9 98 dc 60 8f 19 1e c2 85 cf dc
0050 d1 24 84 a5 46 16 9a a9 8a 86 8e 9a 97 b9 4b 7f
0060 b4 94 93 73 32 ec d9 db 18 96 4a 49 03 a6 4e 22
0070 91 30 fe 94 d2 2c 69 bb 5c a2 9b ce 92 d3 5c 3c
0080 ed 0c 63 37 4f e6 3c 7f 42 ee 2f 25 6e 12 c9 dc
0090 f2 f9 30 fb e7 3d bc e8 25 5b fc ed 40 14 bb 27
00a0 55 e1 e4 53 f1 9d 72 95 4b 59 e3 6d de 25 77 f3
00b0 62 7c b6 c8 d5 6c 30 4e 1d d4 ba 7e a7 32 38 11
00c0 f8 4e 98 9a 26 12 31 80 60 7e 81 b7 a3 00 ae 7e
00d0 aa ce 12 d2 64 f5 52 10 82 f4 d9 3a e4 ad 37 88
00e0 d7 68 97 e8 2d 9f b1 74 1b 9f db 28 45 a8 35 94
00f0 27 c5 01 53 b4 91 30 fc 33 10 d0 c8 ae 50 a2 15
0100 1b be bd e9 5c ab 60 87 b3 7d df 99 22 50 1f 88
0110 bc 52 33 61 23 11 5b ec d9 b2 80 d4 cc 10 49 03
0120 f6 06 13 ec b6 5b 83 81 e6 7b 69 52 06 95 14 61
0130 c1 04 77 60 b7 f8 4c 74 26 2d fd 33 23 a5 c3 19
0140 55 90 6d 9a bf 59 8d 6c d4 6b 28 68 60 c8 be 43
0150 0e 34 a0 95 a9 09 c4 23 8c 67 5a 8c da 5c fe 39
0160 e0 01 8f 0a bb 80 e3 0d 54 fd e1 51 34 4c 1d 6f
0170 6f 70 a6 77 21 41 b9 5e 35 e5 42 36 58 c0 f7 93
0180 ad ad ae 94 88 aa db cd d0 72 cf 93 8c db 0d d2
0190 20 01 98 c3 f6 54 b4 7c d7 a4 f7 2a ea a9 bf 5f
01a0 94 33 bb 24 e3 ab 95 f7 f3 0e 9c 14 a7 70 9c 0e
01b0 be 5b d3 a4 36 be 95 bb ce 62 00 d1 e4 2e 8e c3
01c0 7f 67 2e c1 5a 24 d6 28 7e 57 9e e4 55 c3 30 0a
01d0 f9 4e 01 9c 6c 74 ad 5c 22 a7 bb 09 10 4d 25 7d
01e0 ba bf 81 a3 29 22 6f ff 4c c5 e8 75 42 a1 7f 33
01f0 c8 d0 e4 3f f1 16 64 20 da a7 ec cb 64 66 bb 21
0200 3f d0 8b 69 d9 ba 96 c2 7f 89 6a dd 9d 83 8b 83
0210 4f c4 84 da 51 78 fb e7 19 e2 34 6b fd 84 cb 0f
0220 bc c7 d1 cf 5c 55 c3 fe ae f5 e2 90 f2 b9 96 ca
0230 19 d0 85 93 05 bf ea f1 82 f6 aa dc fe c8 6e b8
0240 67 5c 1a 84 9a f9 87 c2 53 4b f3 ad 1a f0 4c ad
Frame (1292 bytes)
Decrypted QUIC (1215 bytes)
QUIC IETF: Protocol
Packets: 6015 · Displayed: 1401 (23.3%) · Dropped: 0 (0.0%)
Profile: Default





You can find example captures with complete handshakes online or through tutorials like this one (https://www.youtube.com/watch?v=59X0iRI3HwU) and (https://www.youtube.com/watch?v=fHBUOlvS3ts) (notice it requires server-side decryption).

Future of QUIC

There is still research and development being done on QUIC. Here are a few possible avenues for future research:

- **Wider Adoption:** In order for QUIC to take the lead as the primary transport protocol, more server and client support is required.

- **Integration with Current Infrastructure:** To ensure a smooth integration, compatibility with current network infrastructure, such as firewalls and intrusion detection systems, must be taken care of.

- **Standardization:** The QUIC standard will be further defined and solidified by the IETF QUIC Working Group's ongoing efforts, guaranteeing uniform vendor implementation.

Conclusion

The conclusion emphasizes how much better online browsing could be with QUIC. Below is a summary of the main ideas and the reasons behind them:

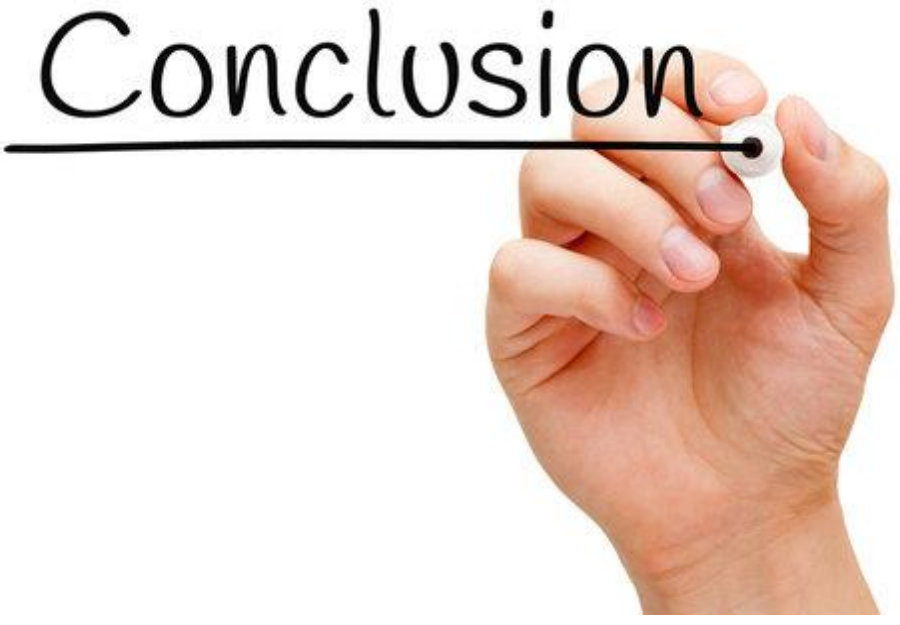
- **Faster Connections:** The handshake procedure and data retransmission techniques of TCP, the currently popular protocol, might cause delays. Because QUIC is based on UDP, it can potentially result in faster loading times and more seamless web experiences by avoiding these delays.

- **Enhanced Congestion Management:** Online traffic bottlenecks also exist! Data transmission is governed by congestion control to prevent overstuffing networks. In instances where TCP is overloaded, QUIC's method is intended to be more effective, resulting in less buffering and more seamless data transfer.

- **Built-in Security:** QUIC has encryption built right in. Security is really important. This simplifies the procedure and might enhance performance by doing away with the requirement for an additional layer, such as TLS (Transport Layer Security), which is utilized with TCP.

- **Greater Adoption and Revolution:** The advantages of QUIC will expand as more servers and browsers integrate it. The way we engage with websites could be drastically altered by its widespread adoption, making them feel faster and more responsive.

Although QUIC has potential, it is still developing. It's crucial to remember that not all websites and browsers now support it. With its benefits, QUIC is anticipated to have a significant influence on how the internet develops in the future.



Recommendations for Further Exploration

For a deeper understanding of QUIC, consider exploring these resources:

IETF QUIC Working Group: https://datatracker.ietf.org/wg/quic/about/

A QUIC Tutorial: https://www.debugbear.com/docs/website-monitoring-getting-started

Wireshark QUIC Capture and Analysis: https://www.wireshark.org/docs/dfref/q/quic.html

References

1. Auvik Networks. (2021, September 14). *What is QUIC? Everything You Need to Know | Auvik Networks*. Auvik Networks Inc. https://www.auvik.com/franklyit/blog/what-is-quic-protocol/
2. *QUIC*. (2020, February 25). Wikipedia. https://en.wikipedia.org/wiki/QUIC
3. Lin, L. (2023, March 7). *What is QUIC? How Does It Boost HTTP/3?* CDNetworks. https://www.cdnetworks.com/media-delivery-blog/what-is-quic/#:~:text=QUIC%20includes%20a%20built%2Din
4. Wang, F. (2023, April 26). *QUIC Protocol: the Features, Use Cases and Impact for IoT/IoV*. Www.emqx.com. https://www.emqx.com/en/blog/quic-protocol-the-features-use-cases-and-impact-for-iot-iov
5. *Google Edge Network*. (n.d.). Peering.google.com. https://peering.google.com/#/learn-more/quic
6. 262588213843476. (n.d.). *QUIC header format proposal*. Gist. Retrieved June 4, 2024, from https://gist.github.com/martinthomson/744d04cbcec9be554f2f8e7bae2715b8
7. Saleh Alawaji. (2021). *IETF QUIC v1 Design*. Wustl.edu. https://www.cse.wustl.edu/~jain/cse570-21/ftp/quic/index.html
8. *What are the main differences and similarities between QUIC and TCP?* (n.d.). Www.linkedin.com. Retrieved June 4, 2024, from https://www.linkedin.com/advice/3/what-main-differences-similarities-between-1e
9. Hadar, R. (2022, April 4). *QUIC: is it the game changer for Internet delivery?* Compira Labs. https://www.compiralabs.com/post/quic-is-it-the-game-changer-for-internet-delivery