Performance analysis of QUIC protocol in comparison with HTTP and HTTPS servers

Author-T. Anjali

Abstract — Since the discovery of internet, we have always managed to use http and TCP protocol. Google developed a new user interface protocol called QUIC (Quick UDP Internet Connection) that uses UDP as its basis. This research paper is an experimental proof of the performance analysis between QUIC and HTTPS on the basis of speed of transfer of accurate data.

Index Terms— UDP,TCP, QUIC, Handshake protocols

1. Introduction

HTTPS is built on adding SSL on HTTP and security certificates. HTTPS is known to provide better authentication and security compared to HTTP. However, https uses 8 hand shakes in order to transfer information from server to client. this includes the authentication of private and public key, accepting the follow of information, securing the details and the client's history. HTTPS works on the basis of TCP (Transmission Control Protocol). TCP is one of the main protocols in Internet protocol suite (IP). TCP provides reliable and order of delivery in contradictory to UPD (User datagram Protocol). UDP on the other side doesn't follow an order for delivery.

QUIC protocol is build with the basis of UDP. This as discovered by google, is found to be faster and equally or possibly more secure. The following paper helps in understanding of all these concepts in detail and later on analyse the speed and performance of QUIC.

2.Open system interconnection model

The open system interconnection model (OSI) is a conceptual model whose main goal is to build telecom layers in order to communicate between any communication system with standard communication protocol. This model divides the communication process into basic 7 layers originally. they are listed as follows

- Physical layer
- Data link layer
- Network layer
- Transport layer
- Session layer
- Presentation layer
- Application layer

Each layer has unique communication method. This method of exchanging is known as Protocol Data Unit (PDU). Each PDU contains a payload known as Service Data Unit (SDU). Along with protocol header and footer.

A layer serves the layer above it and is served by the layer below it. for example, layer N is served of provided by layer N-1 and in turn provides to the layer N+1. The process of communication takes place in the following steps.

- The topic to be transmitted in put at the top most layer of the transmitting divide (layer N)

- The PDU is passed on to layer N-1 where it is known as SDU
- This SDU is then attached with a header and footer and then sent to N-2 as a PDU
- This process continuous until the data reaches the lowest lever and the data to be transmitted is received.
- The process occurs in reverse action same as above to reach to the topmost layer

The layers of OSI are divided into two parts. Host layers and Media layers. Host layers consist of Application, Presentation, session and transport. the Media layers consist of Network, Data link and Physical layer.

S.NO	LAYER	FUNCTION
1	Physical Layer	The physical layer is responsible for the transmission and reception of unstructured raw data between a device and a physical transmission medium. It converts the digital bits into electrical, radio, or optical signals
2	Data Link Layer	The data link layer provides node-to-node data transfer—a link between two directly connected nodes. It detects and possibly corrects errors that may occur in the physical layer. It defines the protocol to establish and terminate a connection between two physically connected devices. It also defines the protocol for flow control between them.
3	Network Layer	The network layer provides the functional and procedural means of transferring variable length data sequences (called packets) from one node to another connected in "different networks"
4	Transport Layer	The transport layer provides the functional and procedural means of transferring variable-length data sequences from a source to a destination host, while maintaining the quality of service functions.
5	Session Layer	The session layer controls the dialogues (connections) between computers. It establishes, manages and terminates the connections between the local and remote application. It provides for full-duplex, half-duplex, or simplex operation, and establishes procedures for checkpointing, suspending, restarting, and terminating a session.
6	Presentation layer	The presentation layer establishes context between application-layer entities, in which the application-layer entities may use different syntax and semantics if the presentation service provides a mapping between them. If a mapping is available, presentation protocol data units are encapsulated into session protocol data units and passed down the protocol stack.
7	Application Layer	The application layer is the OSI layer closest to the end user, which means both the OSI application layer and the user interact directly with the software application. This layer interacts with software applications that implement a communicating component.

The use of OSI is age old and usually restricts from fast and secure connections. It also restricts from the use of many networks at the same time i.e. many clients to one server being a major restriction hence forth, the IP model has been adopted.

3. IP (Internet Protocol)

Internet Protocol is a conceptual method and set of protocols used in the internet and other networks. The IP provides end to end communication. this is done by specifically packetized, addressed, transmitted, routed and received of data. The functionality of this is done with four layers.

- Link Layer- the link layer defines the local network that its user uses without interfering with other networks or routers. This protocol in clues the tropology of the local network connections. this includes Eternals, ARP, PPP
- Internet layer – this layer helps exchange the datagram across layers and boundaries. It forms a uniform networking interface that hides the actual topology of the underlying connection
- Transport layer – helps perform end to end communication between two routers or remote networks. it provides a channel for application layer to communicate in UDP is the basic transport layer. Others include TCP
- Application layer- the application layer is the space where user can be comfortable enough to provide instructions of requirements for the system to process. it helps communicate between the user and the system interface. This is the protocol where HTTP, QUIC, SMTP and other application processes function

S.NO	TRANSMISSION CONTROL PROTOCOL(TCP)	USER DATAGRAM PROTOCOL(UDP)
1	TCP establishes a connection before communicating and closes the connection after the communication is done 0	UDP on the other hand is Datagram oriented program henceforth there are no overheads of creating and maintaining a connection
2	It is based on host to host connections	It is based on process to process connection
3	TCP guarantees delivery of the data in packets at the router	UDP doesn't guarantee delivery as its not in an order
4	Sequence checking is a feature in TCP. The order of the data packet in delivery can be checked	UDP doesn't guarantee the sequence of the data sent as it doesn't deal with packets of data
5	Sends messages as induvial packets	Sends messages as datagrams
6	Doesn't provide any port number	Provides port number to distinguish between user requests
7	TCP is comparatively slower than UDP	UDP is faster simpler and more efficient than TCP
8	TCP is used by HTTP, HTTPS, FTP and telnet	UDP is used by DNS, DHCP, TFTP and RIP

We have learned about the basics so far that contain what is OSI, what is IP and what are the difference between TCP and UDP. the application of all this knowledge helps us understand better the implementation of HTTP, HTTPS and QUIC protocols. The server client connections and the programming that's required.

4.HTTP (Hyper Text Transfer Protocol)

HTTP functions as a request- response protocol in the client server computing model. HTTP is designed to permit intermediate network elements to improve or enable communication between client and server. An HTTP client initiates a request by establishing

a transmission control protocol connection particular port on a server.

An HTTP session is a sequence of network request – response transactions. HTTPS request message consist of the

- A request lines
- Request header fields
- An empty line
- An optional line

There are various request methods like GET, HEAD, POST, OPTIONS, PUT, DELET TRACE etc.

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

Figure 1 methods of HTTPS

HTTPS (Hyper Text Transfer Protocol Server)

HTTPS is used for secure communication. The communication protocol in HTTPS is encrypted using TLS (Transport Layer Security). HTTPS is a secure form of HTTP> it wraps a layer of SSL or TLS around the HTTP server and then transmits it. HTTPS is based on Public/Private key cryptography where public key is used for encryption and private key is used for decryption which is unique for each client.

To use or create a HTTPS server client we must first generate an SSL certificate.

5.Transport Layer Security(TLS)

It is designed to provide communication security. They typically use long term public and private keys to generate a short-term session key which is then used to encrypt the data flow between client and server.

The working of TLS/SSL occurs in two steps

- Asymmetric Cryptography
- Symmetric Cryptography

5.1 ASYMETRIC CRYPTOGRAPHY

- Uses a mathematical related key pair to encrypt and decrypt data
- In a key pair, one key is shared with anyone interested to communicate. this is public key. the other key is kept secret hence its known as private key
- Keys are generated using a mathematical algorithm

5.2 SYMETRIC CRYTOGRAPHY

- The above generated key is stored and shared between the server and client
- Both the client and server contain one similar key that's used for communication

When a browser connects to an HTTPS server, the server answers with its certificate. the browser checks if the certificate if valid or not. This conversation between the server and the client takes place in set of steps known as handshakes. The handshakes can be noted as bellow

- The initial request is sent to the server for verification
- When the server responds that it is the desired server the client sends a hollow message
- At this point the communication becomes encrypted
- The server and client exchange the encryption key

- After this the communication can take place

So far, we have managed to establish the existing methods of communication uniformly accepted by all servers. But google has been trying to establish a protocol that's different from HTTPS known as QUIC. What is QUIC.

6.QUIC (Quick UDP Internet Connections)

Quick UDP Internet Connections (QUIC) is a new transport protocol developed for internet that was developed by Google.

Quic is encrypted with default internet transport protocol. The initial QUIC handshake combines the typical three-way handshakes that we get with TCP, with the TLS 1.3 handshake, which provides authentication of the end points as usual as negotiation of cryptographic parameters. For those familiar with the TLS protocol, QUIC replaces the TLS record layer with its own framing formal, while keeping the same TLS handshake message.

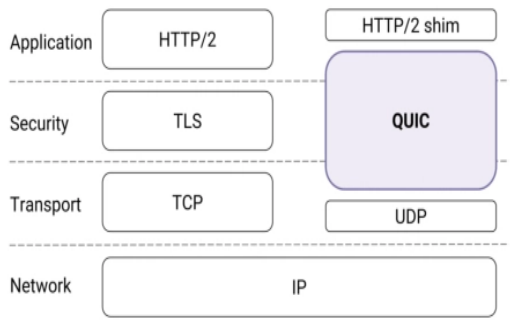

Figure 2 – handshake layers

The key features of QUIC over the existing TCP+TLS+HTTPS2 are

- Dramatically reduced connection establishment time
- Improved congestion control
- Multiplexing without head of line blocking
- Forward error corrections
- Connection migrant

The improvements that QUIC provide over HTTPS are

6.1 BUILT IN SECURITY

One of QUIC's main features when compared to now existing TCP is providing a secured by default transport protocols. QUIC handles this by inbuilt encryption and cryptography which in case of HTTPS are established by higher level protocols like TLS. the initial handshake established by QUIC eliminates 1-3 handshakes that take TLS 1.3 to establish in case of HTTPS.

QUIC replaces the TLS record layer with its own frame work while keeping the original TLS message intact. This not only ensures the connection is always authentic and encrypted but also makes the initial connection establish faster and in a smaller number of round trips.

The following flowchart will help understand the development of QUIC protocol better.

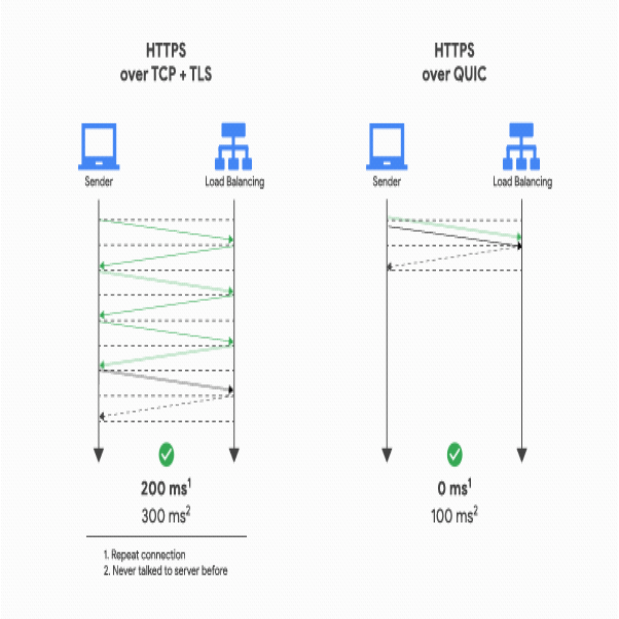

Figure 3 – communication number decreases

6.2 HEAD-OF-LINE BLOCKING

One of the main improvements delivered by HTTPS2 was the ability to multiplex different HTTP requests at the same TCP connection. This allows HTTPS 2 application to process request concurrently and use the available bandwidth and time better. creating a new connection requires repeating previous handshakes many a times, multiplexing of HTTP decreases all the addition repeating. But the back foot of this proposal was many inputs at a time usually cause in loses of data or packets mid-way. Hence forth QUIC goes a lit bit deeper and provides first class support of multiplexing such that different HTTP streams can in turn be mapped to different QUIC transport streams, but, while they still share the same QUIC connections so no additional handshakes will be required

6.3 STEPS TO TEST QUIC PROTOCOL ARE AS FOLLOWS

1) building a QUIC client and server . a sample server and client are provided in chromium project . the basic requirements to play around with QUIC are included in the link attached http://www.chromium.org/developers/how-tos/get-the-code

To build QUIC and text it the following code is to be type in the command window

```
ninja -C out/Debug QUIC server QUIC client
```

2) lets prep test data using an example as www.example.org

```
mkdir /tmp/quic-data
cd /tmp/quic-data
wget -p --save-
headers https://www.example.org
```

3)in order to run a server we need to generate a certificate with a private and public key. Type the next following in the command prompt

```
cd net/tools/quic/certs
./generate-certs.sh
cd -
```

4)running the QUIC client and server. To first run the QUIC server you have recently installed type the following code in the command prompt

```
./out/Debug/quic_server \
 --quic_response_cache_dir=/tmp/quic-
data/www.example.org \
 --
certificate_file=net/tools/quic/certs/out/leaf_cer
t.pem \
 --
key_file=net/tools/quic/certs/out/leaf_cert.pkcs
8
```

Similarly, to run the QUIC client the following code is to be run

```
./out/Debug/quic_client --host=127.0.0.1 --
port=6121 https://www.example.org/
```

The above steps help in creating a QUIC protocol and running it.

The time taken to communicate is calculated by the difference between strike time and reaction time. The following were the outcomes taken and analysed .

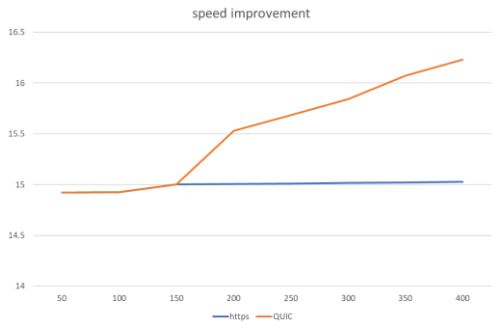

Figure 4 – speed improvement analysis

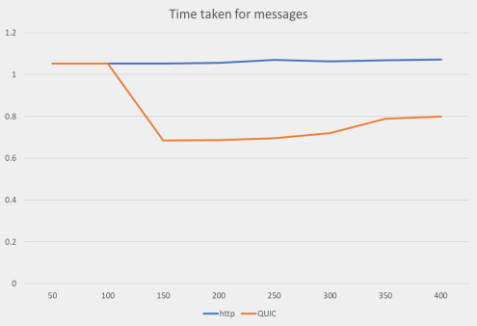

Figure 5 – time improvement analysis

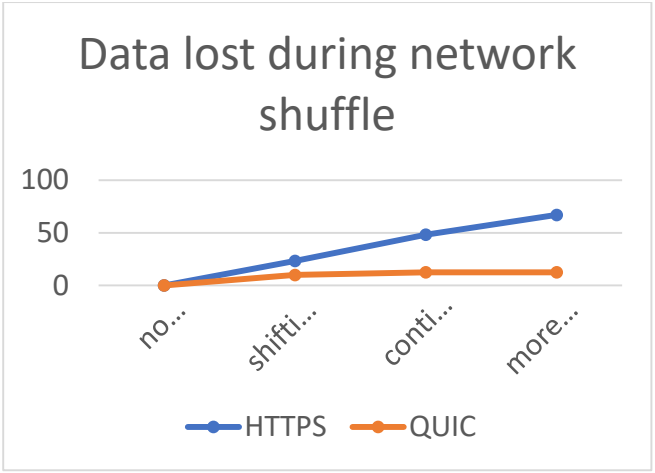

Figure 6 – data accuracy improvement

the future

7. Conclusion

The advanced speed and accurate data transfer and the process of skipping the delay and buffer time conclude that QUIC protocol introduced by GOOGLE will play a key role in

Acknowledgements

We are very grateful to experts for their appropriate and constructive suggestions to improve this template.