WELCOME to my Presentation on :

QUIC Internet Protocol

Presented by :

Shakib Hossain
IT22011
Department of Information and Communication Technology,
MBSTU

Instructed by :

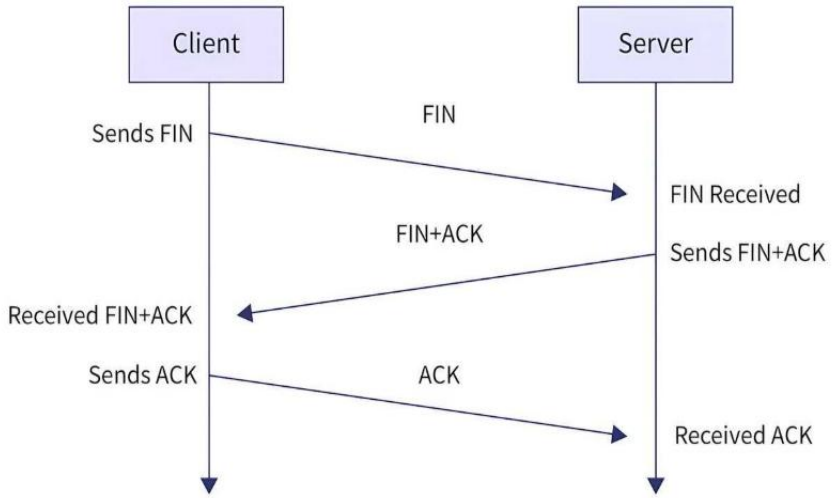
Dr. Nazrul Islam
Associate Professor,
Department of Information and Communication Technology,
MBSTU

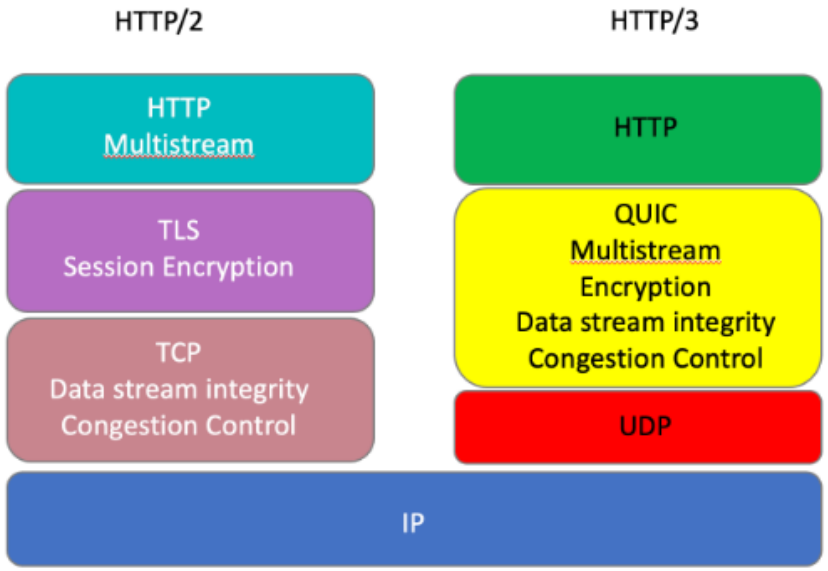
Table of indexes :

Introduction to QUIC :

QUIC (Quick UDP Internet Connections) is a modern transport layer protocol developed by Google, aimed at improving the performance and security of internet connections.

It effectively addresses latency issues and ensures reliable and secure connections through an optimized handshake process.





Purpose of QUIC:

The **main purpose of QUIC (Quick UDP Internet Connections)** is to make **Internet communication faster, more reliable, and more secure.**

1. Making transmission more reliable.
2. Improving performance.
3. Enhancing transmission speed and reduce latency.
4. Providing string encryption for security.

Technical Overview :

Actually QUIC is a updated UDP protocol system that overcomes the issues in old protocols like reliability, speed and security through a optimized handshaking process :

1. Reliability of TCP protocol
2. Speed of UDP
3. Encryption of TLS 1.2

TCP Handshake: (1 RTT)

TCP or Transfer Control Protocol mainly uses HTTP/2 , a 3 way handshake process to connect with server before data transfer begins.

Step 1: SYN (Synchronize)

Client → Server

Client sends a **SYN** (synchronize) packet to start a connection.

Client → Server: SYN, Seq = x

I want to start a connection. My sequence number starts at x.

Step 2: SYN + ACK (Synchronize + Acknowledge)

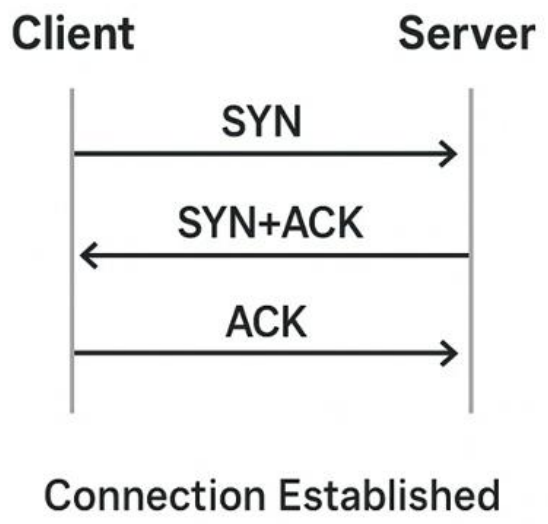
Server → Client

The server receives the SYN.

It replies with its own **SYN** and an **ACK** .

Server → Client: SYN + ACK, Seq = y, Ack = x + 1

I received your request (x). My sequence number starts at y.



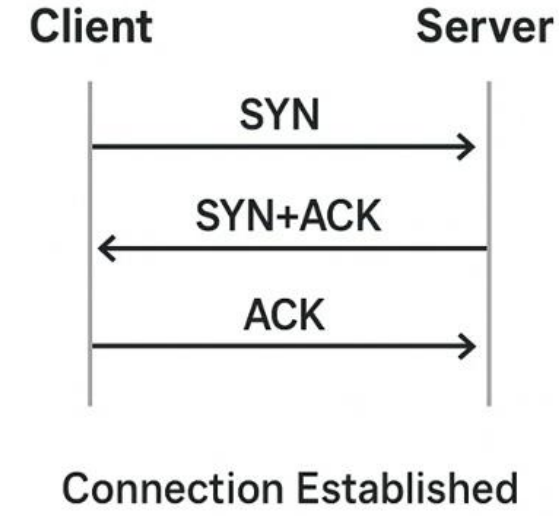
Step 3: ACK (Acknowledge)

Client → Server

The client sends back an **ACK** packet to acknowledge the server's SYN.

Client → Server: ACK, Seq = x + 1, Ack = y + 1

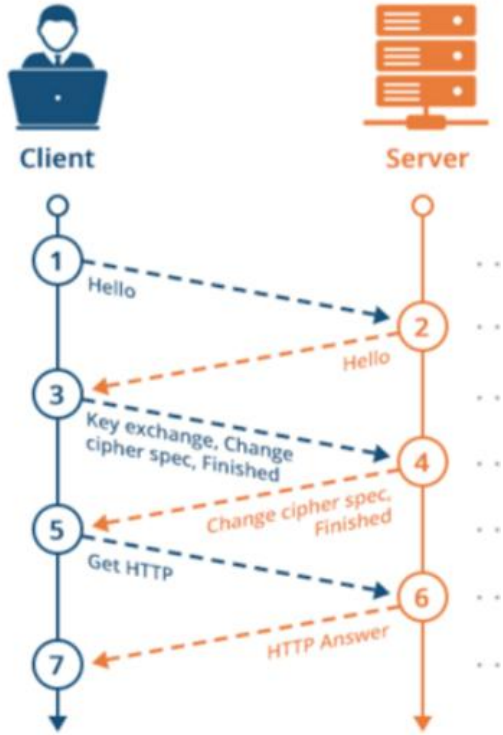
I received your SYN (y). Let's start communication.



TLS 1.2 Handshake: (2 RTT)

TLS 1.2 use additional 2 round trip for encryption process before transmission starts:

1.Client → Server : “clienthello”
2.Server → Client : “serverhello+certificate”
3.Client → Server : “Key Exchange”
4.Server →Client : “Finished encryption”

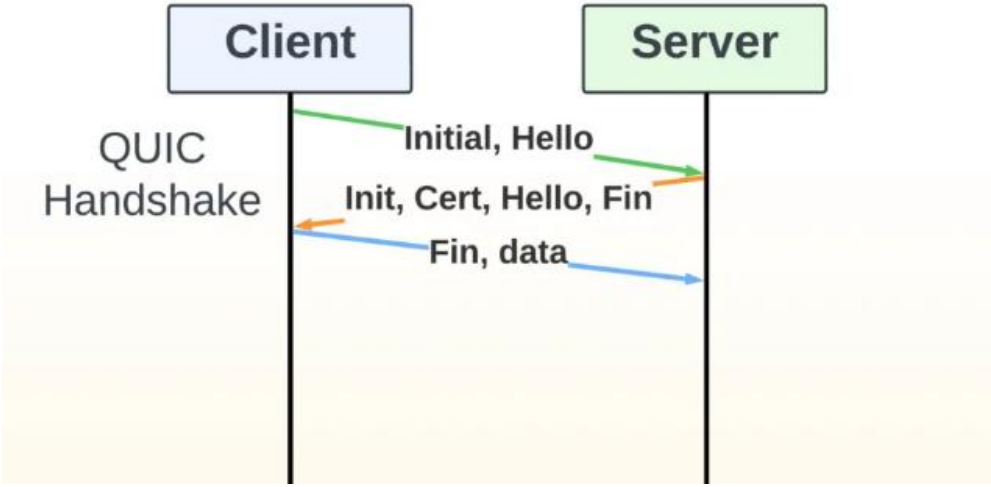


NOTE : TCP+TLS/1.2 →(1RTT+2RTT)= 3RTT

How QUIC works:

QUIC uses its **own single handshake** that combines **connection setup + encryption** — based on **TLS 1.3** — and it usually completes in **1 RTT**. It has its own in-built encryption process(no extra handshake is needed).

1. Client→Server:
   "ClientHello" + key info
2. Server→Client:
   "ServerHello"+ "Certificate" + Finished
3. Client→Server:
   "Finished"+ Encryption starts immediately



NOTE : In total of 1RTT.

Limitations of older Protocols → Solved by QUIC :

Features	Older Protocols	QUIC Internet Protocol
Connection setup latency	TCP requires a 3 way handshake(1RTT) for establishing connection + Needs another 1 or 2 RTT for TLS encryption.	QUIC combines TCP +TLS handshake in a single 1RTT. (It has a built-in security)
Head of Line Blocking	In TCP , if one packet is lost , the remaining must wait even if they maintain synchronization.	No Head of line blocking, lost of packet does not block the entire stream.
Connection Breaking	In TCP if the user switch to another network TCP connection breaks, reconnection needed.	QUIC allows connection migration(wi-fi → mobile data) as it uses unique connection ID instead of just port binding.

Limitations of older Protocols → Solved by QUIC :

Features	Older Protocols	QUIC internet Protocols
Built-in Encryption	TCP or UDP itself does not provide encryption Need to use TLS for security which is optional.	Here encryption is mandatory (Initially uses TLS/1.3 for encryption) It's a built-in encryption.
Multiplexing Efficiency	Http/2 multiplexing over one TCP connection, lost of one packet affects the whole transmission.	True stream-level multiplexing in QUIC itself and each stream transmitted independently.

Features and Advantages of QUIC:

The limitations of the older protocols are the advantages of QUIC:

1. Low latency in connection setup
2. Reliability of packet transmission
3. No Head-of-Line blocking
4. Connection migration
5. Built-in encryption
6. Efficient Multiplexing etc.

Application of QUIC:

QUIC is developed by Google and its an enhanced version of internet connection protocol making it applicable to modern connection and transmission:

1.Web Browing : Quic use the HTTP/3 protocol , the latest version of web protocol.

2. Video Streaming : Platforms like youtube, Netflix etc use quic for video streaming a as it has less interruption.

3.Real time service : Real time apps requires less latency like video calls, onlone gaming use quic protocol system.

4.Mobile apps : Mobile apps that frequently switch between networks use quic protocol as it has no connection breaking.

It has also many beneficial and effective uses .

Limitations and Challenges:

Though QUIC is an enhanced and advance protocol system of internet, but still it has many challenges to face. Some of the limitations are mentioned here:

1. **Limited Support on Older Systems:** Not all operating systems, routers, or firewalls fully support it yet.
2. **Increased CPU Usage and Overhead:** Continuous encryption or decryption increases CPU load compared to TCP.
3. **Difficulty in Network Monitoring and Debugging:** Network administrators can't easily inspect packets for troubleshooting or traffic shaping.
6. **Lack of Mature Ecosystem:** Tools, libraries, and debugging utilities are **less mature** compared to TCP/TLS stack.

Thanks to all