

Bosi Cheng A53271697

Group Partner: Pinzhu Qian A91006019

Question 1:

The "retransmission ambiguity" problem. The retransmitted TCP segment carries the same sequence numbers as the original packet. The receiver of a TCP ACK cannot determine whether the ACK was sent for the original transmission or for a retransmission, and the loss of a retransmitted segment is commonly detected via an expensive timeout. Each QUIC packet carries a new packet number, including those carrying retransmitted data. This QUIC obviates the need for a separate mechanism to distinguish the ACK of a retransmission from that of an original transmission, thus avoiding TCP's retransmission ambiguity problem.

Question 2:

QUIC is based on UDP and its basic transmission unit is Packet whose length never exceeds MTU's length. The encryption and verification is based on Packet.

Question 3:

Because QUIC improves four impacts: Protocol Entrenchment, Implementation Entrenchment, Handshake Delay and Head-of-line blocking delay. On the one hand, it is an ancient protocol with a long history and a wide range of uses. On the other hand, users' usage scenarios have higher requirements for transmission performance. The following long-standing problems and contradictions have become more and more prominent.

QUIC is based on UDP but in early time TCP is more widely adopted. The reason the users, applications, network, end devices, etc that make QUIC relevant in 2017 is

First, the solution to the team's first blocking problem is more thorough. The UDP-based QUIC protocol can solve such problems more thoroughly, so that the real implementations of different streams can be transmitted independently of each other without interfering with each other.

Another feature is the connection retention when switching networks. Based on the TCP protocol, since the IP changes after switching the network, the previous connection cannot be maintained. The UDP-based QUIC protocol can be built with different connection identification methods in TCP to restore the previous connection with the server after the network completes the handover.

case 1:

GET /index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: an HTML file called index.html exists in the doc root. This also applies to jpg and png format files.

Expected behavior: the server returns a 200 status code reply followed by the text of the index.html file

case 2:

GET /index.html HTTP/1.1<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because the Host header is missing.

case 3:

GET /index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: no HTML file called index.html exists in the doc root. This also applies to jpg and png format files.

Expected behavior: the server returns a 404 status code because the requested content wasn't valid.

case 4:

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because the request doesn't have initial line and the server can not understand.

case 5:

GET /index.html HTTP/1.1<CR><LF>

Host: www.css.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: the Host `www.css.ucsd.edu` doesn't exist.

Expected behavior: The server replies with a 400 status code response because the host name is invalid.

case 6:

GET /index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

Connection: close<CR><LF>

<CR><LF>

GET /index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: an HTML file called index.html exists in the doc root. This also applies to jpg and png format files.

Expected behavior: the server returns a 200 status code then the server will not response to the second request because server has closed connection with the client after sending response for the first request

case 7:

GET /index.html HTTP/1.1<CR><LF>

Host www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because a colon is missing.

case 8:

GET /index.html HTTP/1.1<CR><LF>

Host:www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because a space is missing.

case 9:

GET /index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because the request is malformed without **<CR><LF>**.

case 10:

GET /index.html HTTP/1.1**<CR><LF>**

Host: www.cs.ucsd.edu**<CR><LF>**

User-Agent: MyClient/1.0**<CR><LF>**

Content.Length: 121**<CR><LF>**

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because the request is malformed with two colons in one pair.

case 11:

GET /index.html HTTP/1.0<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: the server returns a 400 status code because http 1.0 protocol doesn't have Host parameter.

case 12:

GET /index.html HTTP/1.1<CR><LF>

Host:www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

Cookie123<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because the header only supports
key: value pair.

case 13:

GET index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: an HTML file called index.html exists in the doc root. This also applies to jpg and files.

Expected behavior: the server returns a 400 status code because index.html doesn't have '/' in the front, so it is not treated as if the client requests the URL.

case 14:

GET /index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

<CR><LF>

Pre-conditions: none

Expected behavior: The server replies with a 400 status code response because the user-agent key is missing.

case 15:

GET \index.html HTTP/1.1<CR><LF>

Host: www.cs.ucsd.edu<CR><LF>

User-Agent: MyClient/1.0<CR><LF>

<CR><LF>

Pre-conditions: an HTML file called index.html exists in the doc root. This also applies to jpg and files.

Expected behavior: the server returns a 400 status code because index.html has syntax/semantic error, so it is not treated as if the client requests the URL. This also applies when index.html ends up with '/'.