QUIC

2024/08/26

Tam

Why QUIC Now?

- HTTP is the protocol for communication between browsers and web servers.
- It has been decided that QUIC will be adopted in the next version of HTTP, HTTP/3.

QUIC ≠ HTTP/3

- Just because you can use QUIC does not mean you can use HTTP/3.
- QUIC was simply adopted as the transport layer for HTTP/3.
- The transport layer in HTTP/1 and 2 corresponds to TCP and TLS, etc.

QUIC = TCP & TLS ?

- To put it bluntly, that's true.
- The reason we are adopting QUIC is because its features are important.
- In other words, to understand the advantages of QUIC, you also need to understand the disadvantages of TCP & TLS.

What is TCP?

- A connection-based two-way communication channel. (This is a rough way of saying it.)
- In simple terms:

1. If one sends the string "Hello!" to the other,
2. The other end receives the string “Hello!” as is.

- The connection is one-way, from the client (service user) to the server (service provider).
- Once a connection is established, data can flow in both directions.
- Even if you send multiple strings, they will be received in the order they were sent.

How to use HTTP communication over TCP

```
% telnet www.google.com 80  < ------- Connect to Google Sites using the telnet command
Trying 172.217.161.228...
Connected to www.google.com.
Escape character is '^]'.
GET /  < --------------------------- Type it in and press ENTER
HTTP/1.0 200 OK
Date: Sun, 28 Feb 2021 07:04:15 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2021-02-28-07; expires=Tue, 30-Mar-2021 07:04:15 GMT; path=/; domain=.google.com; Secure
Set-Cookie: NID=210=WVVRF7ZfcovqExYo_5d0YW9cbESA8v9MhcxQ1CoO_enDhboMmApg0lD4qWWwvhePM_gDK
OOQxMlfZlqIpG4JR1egcP5-OG9zZ-B1eXc9N3Xv3PubRfo2EKWSTKkaTv9ZhUdxP2sLSnjxD8fFAB
IbsDfObJibCL-TDQZ1FxO2xsU; expires=Mon, 30-Aug-2021 07:04:15 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ja">
<head><meta content="&#19990;&#30028;&#20013;&#12398;&#12354;
(HTML follows.)
```

What is TLS?

- TCP is not encrypted.
- If packets are snooped on along the route, such as via wireless LAN, the contents of the communication can be determined.
- TLS can build an encrypted communication channel over a TCP communication channel.

SSH ≠ TLS

- (As many people probably know, SSH and TLS use the same encrypted communication channel but are different things.)
- SSH was developed (mainly) to be used to log in to PCs.
  So in order to use it, you (generally) need account information on the server.
- TLS provides a universal cryptographic transport.

How key exchange works, even for high school students

- Diffie-Hellman (DH) key exchange

1. Assume that g and p are appropriately configured and publicly available.
2. a and b calculate each other and keep it secret (not even to the other person).
3. Client side: Calculate A = g^a mod p and send it to the server.
4. Server side: Calculate B = g^b mod p and send to the client.
5. Client side: Calculate B^a = g^(b * a) mod p and use this as the private key.
6. Server side: Calculate A^b = g^(a * b) mod p and use this as the private key.

How TLS enables encrypted communication

1. ClientHELO is sent from the client to the server.
2. ServerHELO is sent from the server to the client.
3. The server key is sent from the server to the client.
4. The client sends the client key to the server.
5. Both parties check whether they can decrypt the data correctly using each other's key information.
6. They send each other confirmation messages.
7. Encrypted communication begins.

Network Properties

This may seem sudden, but have you ever heard a story like this?

- The network cannot tell whether information is actually reaching the other party.
- Of course, you can't expect any reaction.
- The order of the information is also not guaranteed.
- Sometimes the information gets distorted.

Isn't this different from the nature of TCP?

TCP is awesome

- Assign a number to every piece of information you send.
- Check the information you receive, arrange it in the order it was sent, and receive it in order.
- A CRC check is performed every time to ensure that the information is correct.
- If there are any missing numbers or errors, a retransmission is requested.

3 handshake

- TCP requires a procedure before starting communication with the other party.
- Summary of procedure:

1. A communication start request is made from the client to the server.
2. The server side gives permission to the client side to start communication.
3. The client accepts the start of communication with the server.

- Following these steps ensures that the connection can be initiated.
- This is called a three-handshake.

Disadvantages of TCP & TLS

- Before communication can begin, three handshakes and key exchanges, as well as multiple round-trip packet exchanges, are required.
- If a packet is lost or data is garbled, the only way to restore it is to request a resend of the packet.
- (Although this wasn't mentioned) Since it is implemented at the OS kernel level, it is nearly impossible to customize it in general applications.

Introducing QUIC!

- It basically uses the same three-handshake system as TCP, but it also creates an encrypted communication channel.

If the conditions are met on the same server, even a three-handshake is not required.

- An error-correcting code is used. If a small number of packets are lost, they can be restored from the remaining packets.

We know for certain that they are using error-correcting codes, but we don't yet know the extent of their recovery capabilities.

- Since it communicates using the UDP protocol, it can be built from general applications (if you try hard).

It involves a lot of math, so it's not practical unless you use a library.

Disadvantages of QUIC

- The protocol is very difficult and it seems unlikely that an average programmer could put it together.
- In environments where communication quality is stable, it is much slower than TCP and requires more CPU power.
- There is still some inconsistency in the standards, and communication may not be possible depending on the libraries used.

summary

Use QUIC!