

CS 925

Lecture 14

HTTP2, QUIC & HTTP/3

Wednesday, March 7, 2024

HTTP/2 design goals

- ▶ Improve utilization
- ▶ Reduce latency
- ▶ Improve security
- ▶ Enable fine-grained control over resources

HTTP/2 approach (1)

- ▶ **Multiplexed connections**

- limits Head Of Line (HOL) blocking and eliminates the need for concurrent TCP connections

- ▶ **Resource push**

- reduces latency of waiting for page rendering and subsequent resource request

HTTP/2 approach (2)

- ▶ Support for **low-latency secure connection establishment**
 - utilizes low-latency methods to open secure connections
 - while secure connection is not mandated, many current implementations do not support insecure communication
- ▶ Explicit **bandwidth allocation** for streams within a connection
 - information received concurrently on all streams with bandwidth shared according to the set ratios
 - (still needs some work, see *RFC 9218 Extensible Prioritization Scheme for HTTP* from June 2022)

HTTP/2 steps

- ▶ Secure connection is **established**
- ▶ Individual **streams** are set up
- ▶ **Requests** dispatched
- ▶ Information **received concurrently** on all streams with **bandwidth shared**

QUIC motivation

- ▶ HTTP/2 is trying to match the **performance characteristics** of the underlying transport layer protocol (TCP) and **needs of the application** protocol (HTTP)
 - for example, consider the interaction between TCP Slow Start and typically short HTTP data.
 - HTTP attempts to address this by various methods, such as persistent “Keep-Alive:” connections, reducing the number of RTTs required to open a secure connection, or opening multiple simultaneous connections. While these solutions improve performance, they do not address the core issues with TCP.
- ▶ **Solution:** design an alternative transport protocol

QUIC deployment

- ▶ Changing a widely-used protocol is a complex task!
 - Many lessons were learned from the transition to IPv6 that started more almost 30 years ago and is still far from being done.
- ▶ At least, we do not have to worry about the network itself (network layer), only the end points...
- ▶ ... and, it turns out that Google (at least in the US) controls the most popular browser (Chrome) and provides some of the most significant web applications (search, maps, video, email, storage, web application infrastructure)

QUIC - a silent revolution

- ▶ **Requirement 1**: a way to **negotiate an alternative protocol** that will not break existing protocols and allows a clean fallback on the traditional protocols.
 - Alt-Svc: - Alternative Service in HTTP/3
- ▶ **Requirement 2**: must be **based on an existing transport layer protocol** so that no changes to the protocol stack of the operating systems is required.
 - standard UDP
- ▶ These allow **seamless incremental deployment** that improves performance but does not disrupt

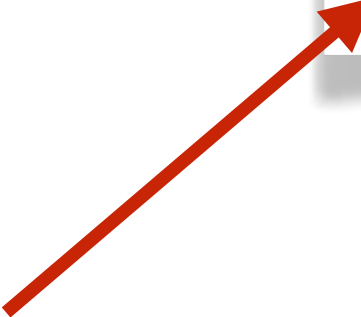
HTTP/3

- ▶ A protocol formerly known as **Hypertext Transfer Protocol (HTTP) over QUIC**
- ▶ The latest major revision of HTTP
 - HTTP/1.1 → HTTP/2 → **HTTP/3**
- ▶ Standardization:
 - QUIC: [RFC 9000](#) (May 2021)
 - HTTP/3: [RFC 9114](#) (June 2022)

HTTP/3 deployment

- ▶ First connection over TCP to port 443:

```
$ curl -I https://google.com
HTTP/1.1 301 Moved Permanently
Location: https://www.google.com/
Content-Type: text/html; charset=UTF-8
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-
src 'nonce-RyGz6SAXxRFuRw4GjadGUg' 'strict-dynamic' 'report-sample' 'unsafe-
eval' 'unsafe-inline' https: http;;report-uri https://csp.withgoogle.com/csp/
gws/other-hp
Date: Tue, 05 Mar 2024 16:58:04 GMT
Expires: Thu, 04 Apr 2024 16:58:04 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 220
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```



HTTP/3 deployment

► Subsequent connections

- UDP packets sent to port 443:

```
$ curl --http3 -I https://www.google.com
HTTP/3 200
content-type: text/html; charset=ISO-8859-1
content-security-policy-report-only: object-src 'none';base-uri 'self';script-src 'nonce-
bmeiVEXqkc08XIFnfqbfqw' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https:
http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Tue, 05 Mar 2024 17:00:42 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Tue, 05 Mar 2024 17:00:42 GMT
cache-control: private
set-cookie: 1P_JAR=2024-03-05-17; expires=Thu, 04-Apr-2024 17:00:42 GMT; path=/; domain=.google.com;
Secure
set-cookie: AEC=Ae3NU9P7lLd41PICgrFA3EGgK44Mjrxvtv593SJcEakkGQZKUfwaf3waLK9k; expires=Sun, 01-Sep-2024
17:00:42 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
set-cookie: NID=512=I3y4lT60oR7bl-f4kYqqmAK1GSqYZam0q5o-mFw-Esjv4meAo-
QWkmjD054Ru335VYGsCn-5jLYKSgGdhyWd4vI3QsH6qyHR-
eVixqmWZ2uvy21PlBVJAG56xokgFYmnCdhDdDw8C2vMKP6L9fhgatUWdVhioiCxKcY0-qoU-M; expires=Wed, 04-Sep-2024
17:00:42 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```

HTTP/3 deployment

► alt-svc cache:

```
$ curl --alt-svc alt-svc.log -I https://google.com
HTTP/1.1 301 Moved Permanently
Location: https://www.google.com/
Content-Type: text/html; charset=UTF-8
...
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

$ cat alt-svc.log
# Your alt-svc cache. https://curl.se/docs/alt-svc.html
# This file was generated by libcurl! Edit at your own risk.
h1 google.com 443 h3 google.com 443 "20240404 17:06:09" 0 0

$ curl --alt-svc alt-svc.log -I https://google.com
HTTP/3 301
location: https://www.google.com/
content-type: text/html; charset=UTF-8
...
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```