CS 725/825 & IT 725

Lecture 11

Application Layer

October 14, 2026

QUIC motivation

- HTTP/2 is trying to match the performance characteristics of the underlying transport layer protocol (TCP) and the 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.
- Better 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 than 20 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: HTTP response header
- 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/2 301                                                    Moved Permanently
location: https://www.google.com/
cross-origin-resource-policy: cross-origin
x-content-type-options: nosniff
server: sffe
content-length: 220
x-xss-protection: 0
date: Mon, 12 Oct 2026 21:20:35 GMT
expires: Mon, 12 Oct 2026 21:50:35 GMT
cache-control: public, max-age=1800
content-type: text/html; charset=UTF-8
age: 1434
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```

- Subsequent connections
  - UDP packets sent to port 443

Electronic Mail - SMTP

- Asynchronous message delivery
  - delivers robustness and reliability
- Two types of agents:
  - User Agent (UA)
  - Message Transfer Agent (MTA)
- Two types of interactions (and protocols):
  - MTA to MTA
  - UA to MTA

UA to MTA Communication

- UA and MTA on the same host (the old days)
  - UA and MTA communicate using files
  - use of host's authentication methods
- UA and MTA communicate over a network (today)
  - SMTP was not designed for this
  - sending mail: SMPT with authentication
  - retrieving mail: IMAP (includes authentication) or "remote authenticated access via HTTP" (webmail)

MTA to MTA communication

- Simple Mail Transfer Protocol (SMTP)
  - covers single hop
  - no encryption
  - no authentication
  - there was supposed to be a “not so simple” mail transfer protocol
  - some problems were addressed by ESMTP (extended SMTP) and other procedural methods

MIME

- Problem: SMTP was designed to deliver limited length, English text
- Solution: MIME (Multipurpose Internet Mail Extensions)
  - encode content to look like text
  - mark it with content type so it can be unpacked and rendered on the receiving end
  - package components of the message

