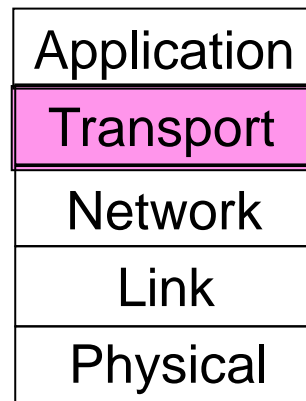
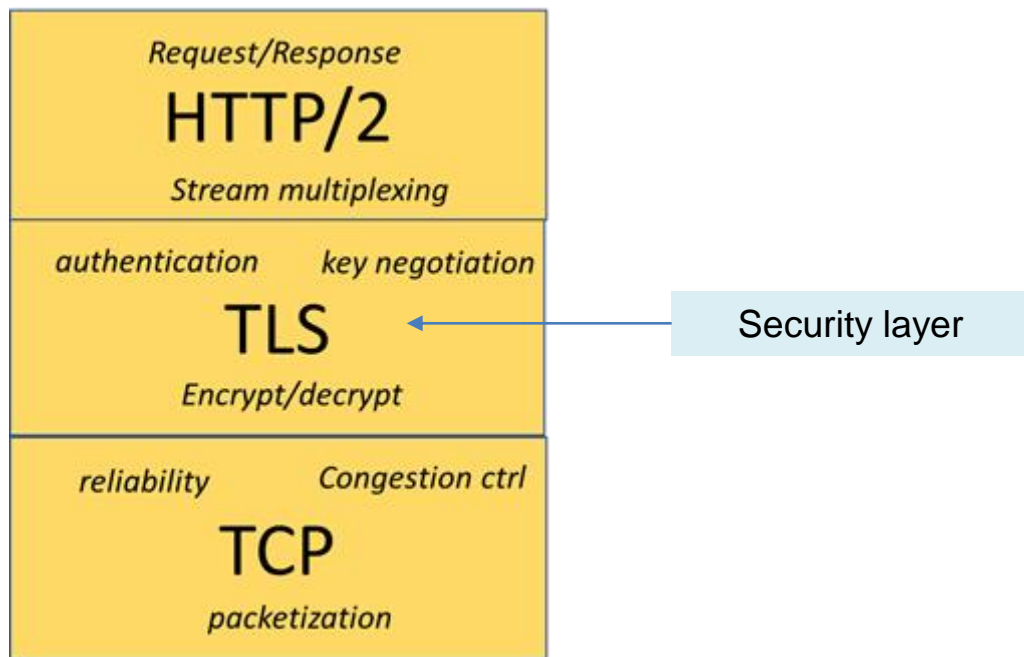


Capítulo 3

QUIC



Introducción



Introducción

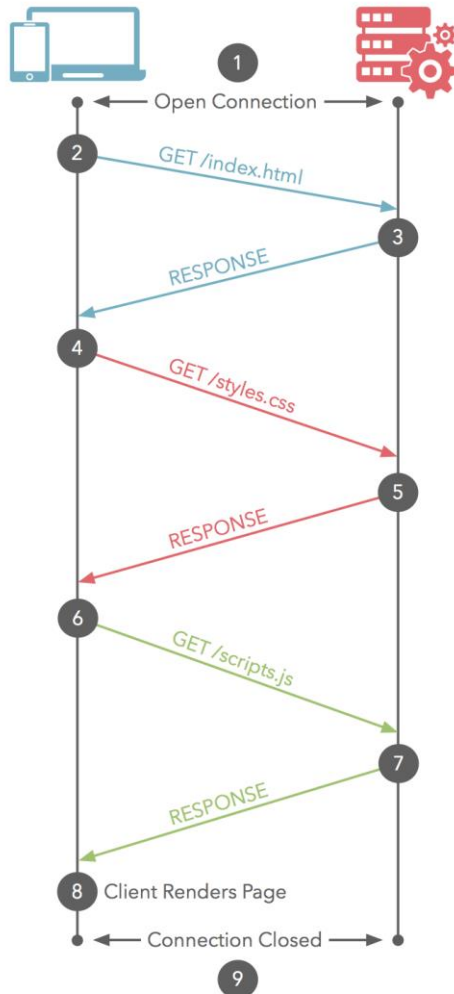
- **Que es QUIC?**
 - Un nuevo protocolo de transporte para **web (HTTP)**
 - Resuelve algunas limitaciones de TCP
- **Basado en UDP**
 - Recuperación de errores, control de congestion y control de flujo contruídos desde cero
- **Seguridad embebida**
 - Combina capa de transporte y seguridad con TLS/1.3



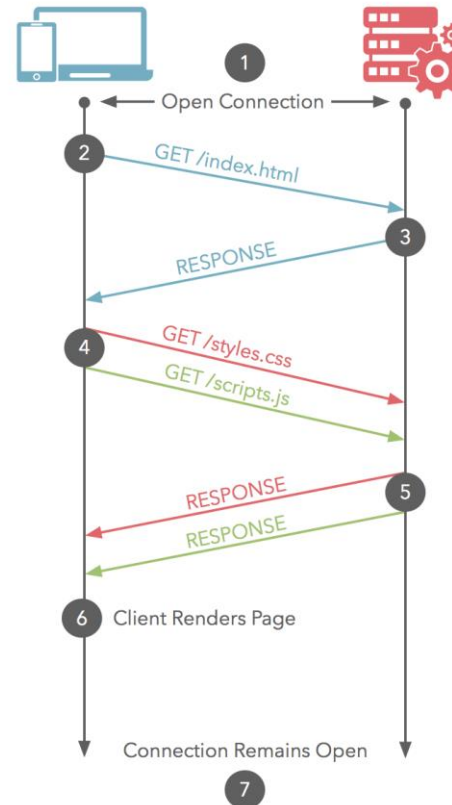
HTTP

Solicitudes independientes
(open, write, close)

HTTP/1.1 Baseline



HTTP/2 Multiplexing



Multiplexación
(Smart buffer management)

Problema de head of line blocking por culpa de TCP...



TCP

- **Especificaciones**

- RFC 793 (1981), y mejoras desde ese entonces
 - (T/TCP 1994, TCP session 1997, Congestión manager 1998, SCTP 2000...)

- **Características**

- Control de flujo y congestión
- Confiable
- Entrega de datos en orden
- Usado para la mayoría de los datos en Internet

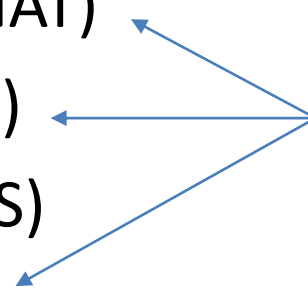


TCP

- Internet hacia finales de los 90s

- Network Address Translators (NAT)
- Firewalls (protección y políticas)
- Intrusion detection systems (IDS)
- Protocol Accelerators (PEP)

Muchos modifican campos de la cabecera de transporte (checksum, numero de puerto, etc.)
“Middleboxes”



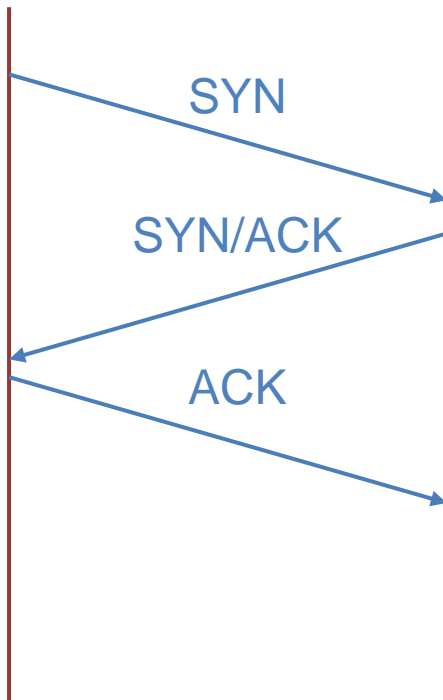
- Además:

- TCP y UDP ampliamente optimizados en Kernels
- Responsable de transportar HTTP, que creció dramáticamente (video, DNS, APIs, etc.)



TCP

- Limitaciones

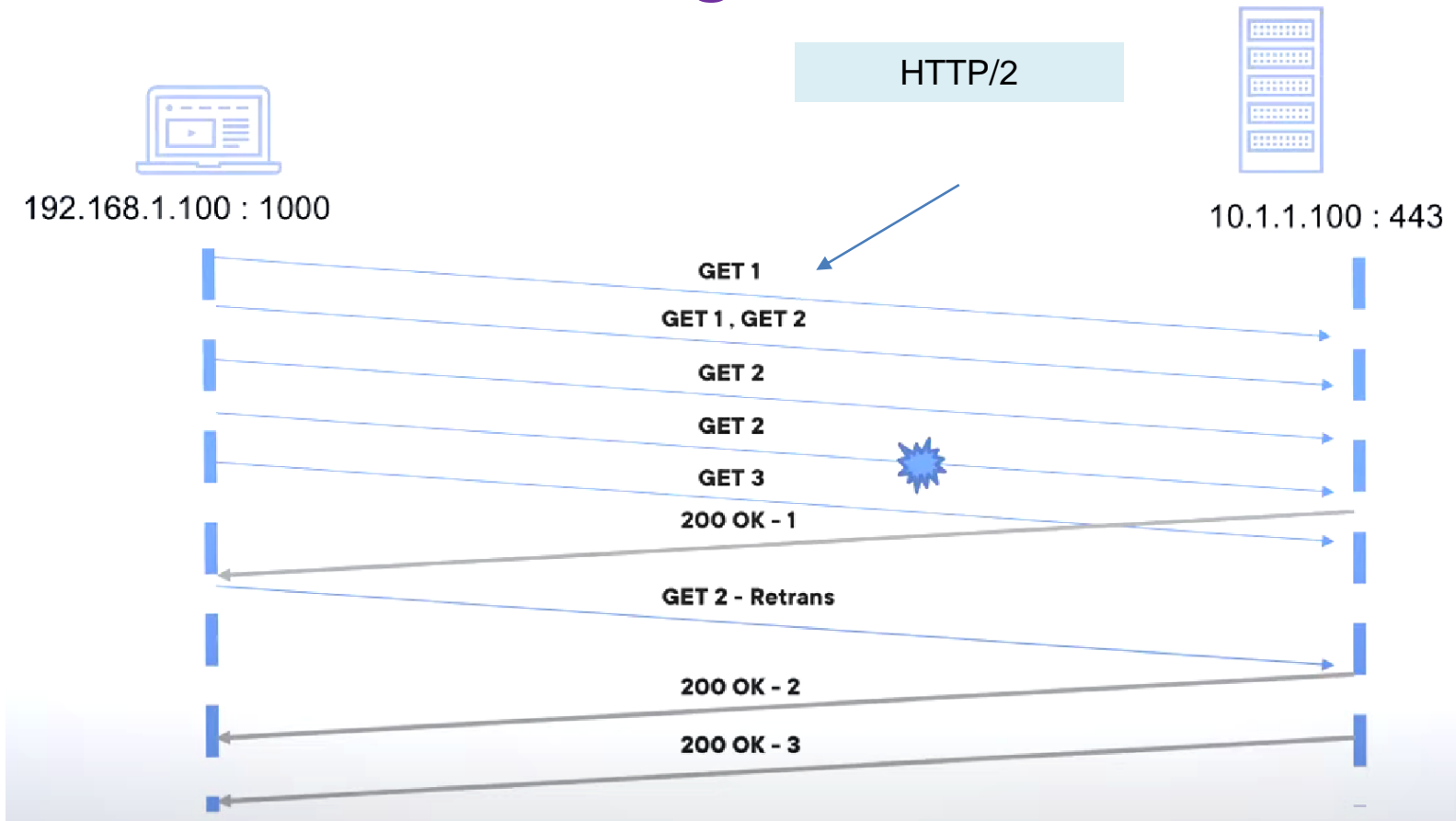


- Nuevo *handshake* para conexiones incluso si fueron recientes
- Cabeceras no encriptadas
- No hay más espacio en el campo de opciones: estancamiento
- *Head of line blocking*: streams puedes generar cuello de botella
- Cuando se combina con TLS los round trip times aumenta



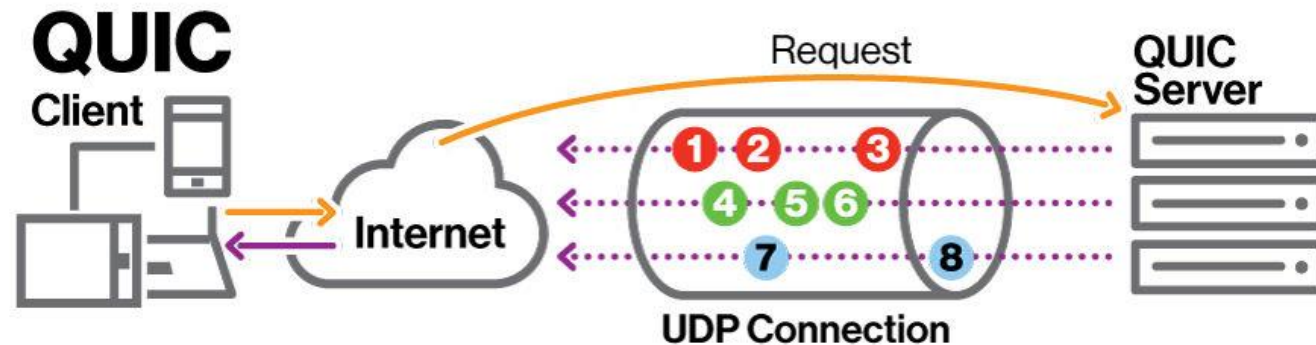
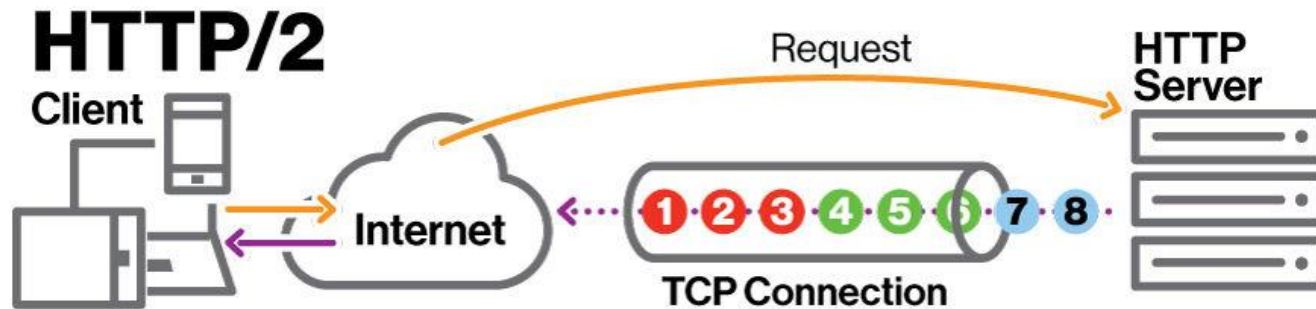
TCP

- Head of Line Blocking



TCP

- Head of Line Blocking



QUIC

- **Historia**

- 2012: Quick UDP Internet Connection (QUIC)
 - Google Services and Chrome (internal)
 - 2014: 35% del tráfico de Google sobre QUIC (7% de Internet)
- 2016: Opened and introduced into IETF
 - No usan el acronimo QUIC (al prototipo de Google se lo conoce como GQUIC.
 - Se mejora notablemente el protocolo (TLS 1.3, formato de paquetes, etc.)
 - 2021: QUIC Estandarizado



QUIC

- **Estándares**

- RFC 9000 QUIC transport
- RFC 9001 TLS for QUIC
- RFC 9002 Loss Detection and Congestión Control

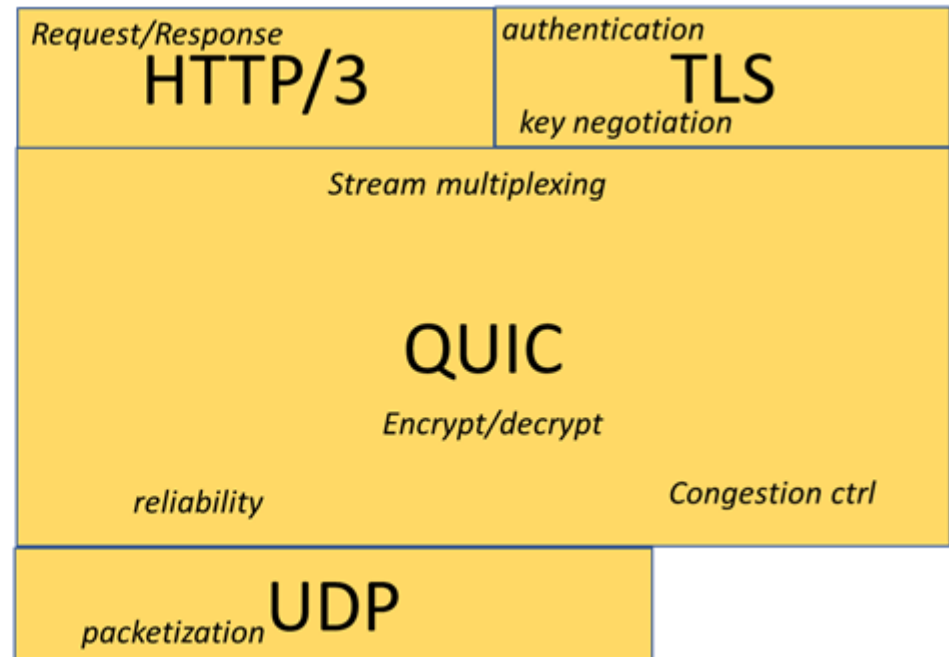
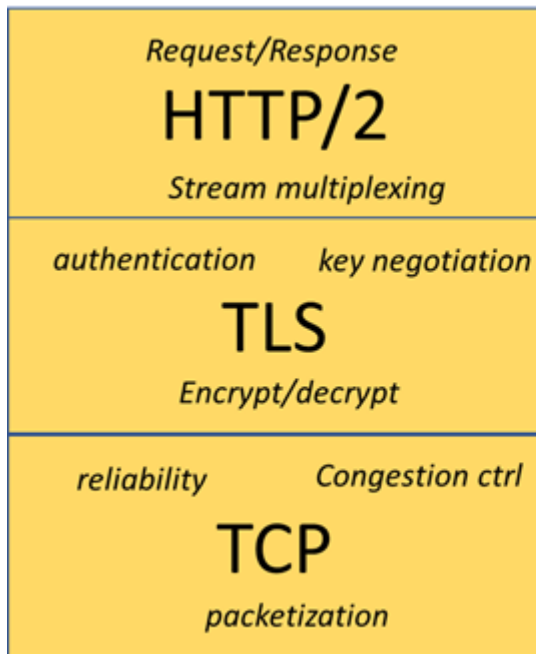
- **Aplicaciones**

- Servidores: Google, Youtube, Facebook,
- Clientes: Chrome, Firefox, Edge



QUIC

- *Stack de Protocolos*



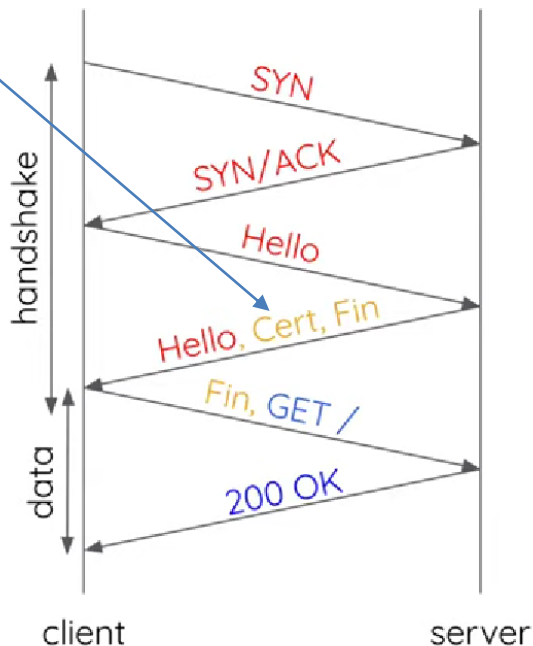
QUIC

- 1) Handshake Reducido

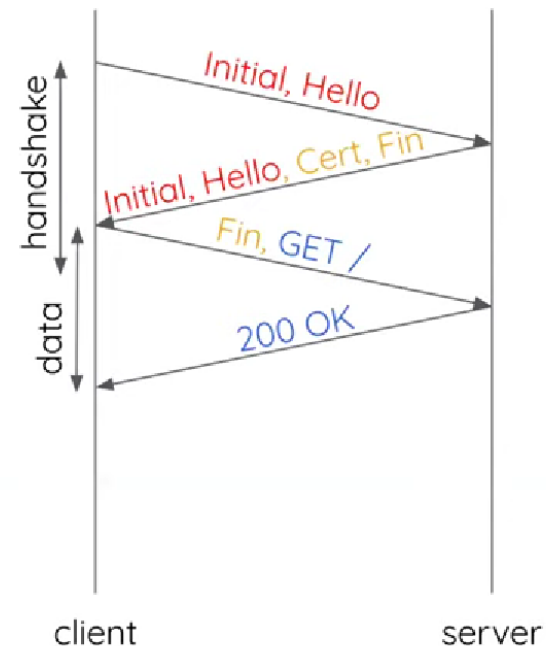
First connection to server

TCP + TLS/1.3

TLS/1.3
1) Client hello, key share, 2) Server hello, key share, certificate validation



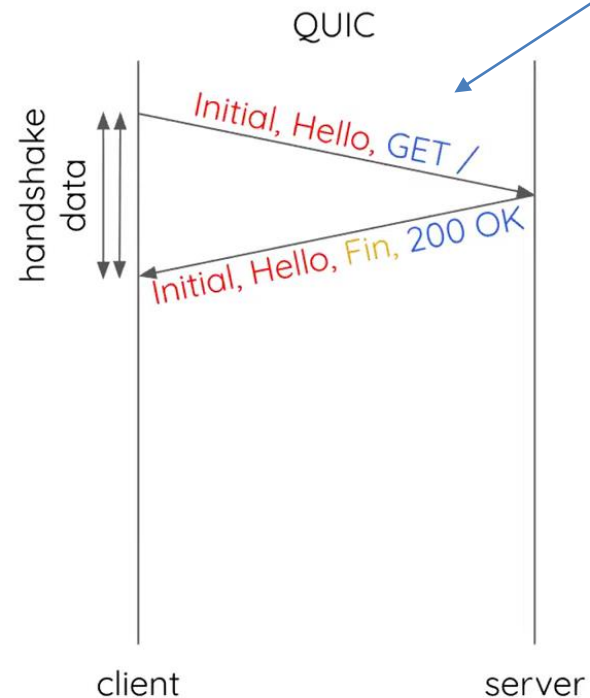
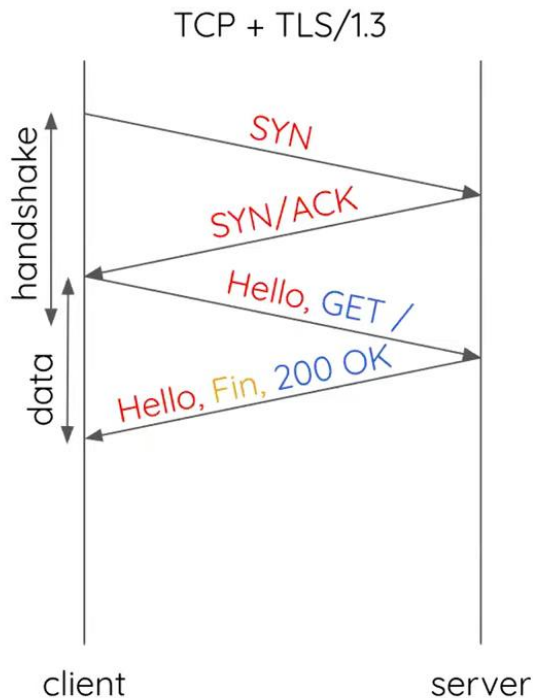
QUIC



QUIC

- 1) Handshake Reducido

Subsequent connection to the same server



Memoria para un Connection ID (18 Bytes)



QUIC

- **2) Opera sobre UDP**
 - Compatible y aceptado por los “middleboxes”
- **3) Multiplexa a nivel transporte**
 - Soluciona el problema de *head of line blocking*
- **4) Encripta incluso la cabecera**
 - La cabecera está encriptada (opciones, secuencias)
- **5) Facilita migración de conexiones**
 - Memoria de un *connection ID* no asociado a IP/puerto
- **6) Mecanismos de control de congestión/flujo**
 - Los mismos que TCP (levemente modificados)



QUIC

- **Reemplazará TCP?**

- Reemplazará varios servicios web
 - Ejemplo: UBER, otros van a seguir
- Pero su beneficio es mínimo o nulo para transferencias de e.g., archivos de gran tamaño
- En el corto plazo:
 - No es soportado por firewalls aún
 - No está optimizado en los Kernels aún
 - Detección de Intrusos (IDS), y load balancers no lo soportan aún
 - En muchos casos puede no justificar el refactoring de apps



Capítulo 3

QUIC

