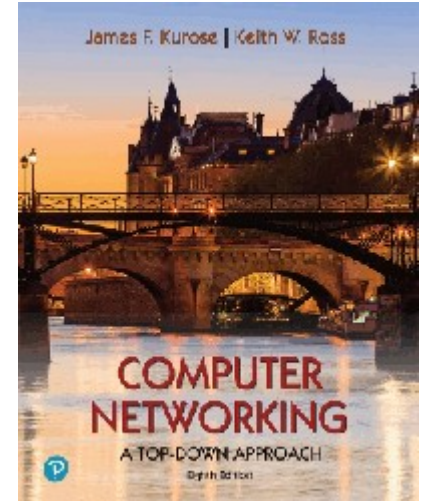


# Redes de Computadores

## Protocolos Transporte: novos desenvolvimentos



Material baseado nas apresentações (*slides*) disponibilizados junto com o livro referência a seguir.

A note on the use of these Powerpoint slides:  
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

All material copyright 1996-2020  
J.F Kurose and K.W. Ross, All Rights Reserved

*Bibliografia:*

*Computer Networking: A Top Down Approach*

*8th Edition, Global Edition  
Jim Kurose, Keith Ross  
Pearson 2020*

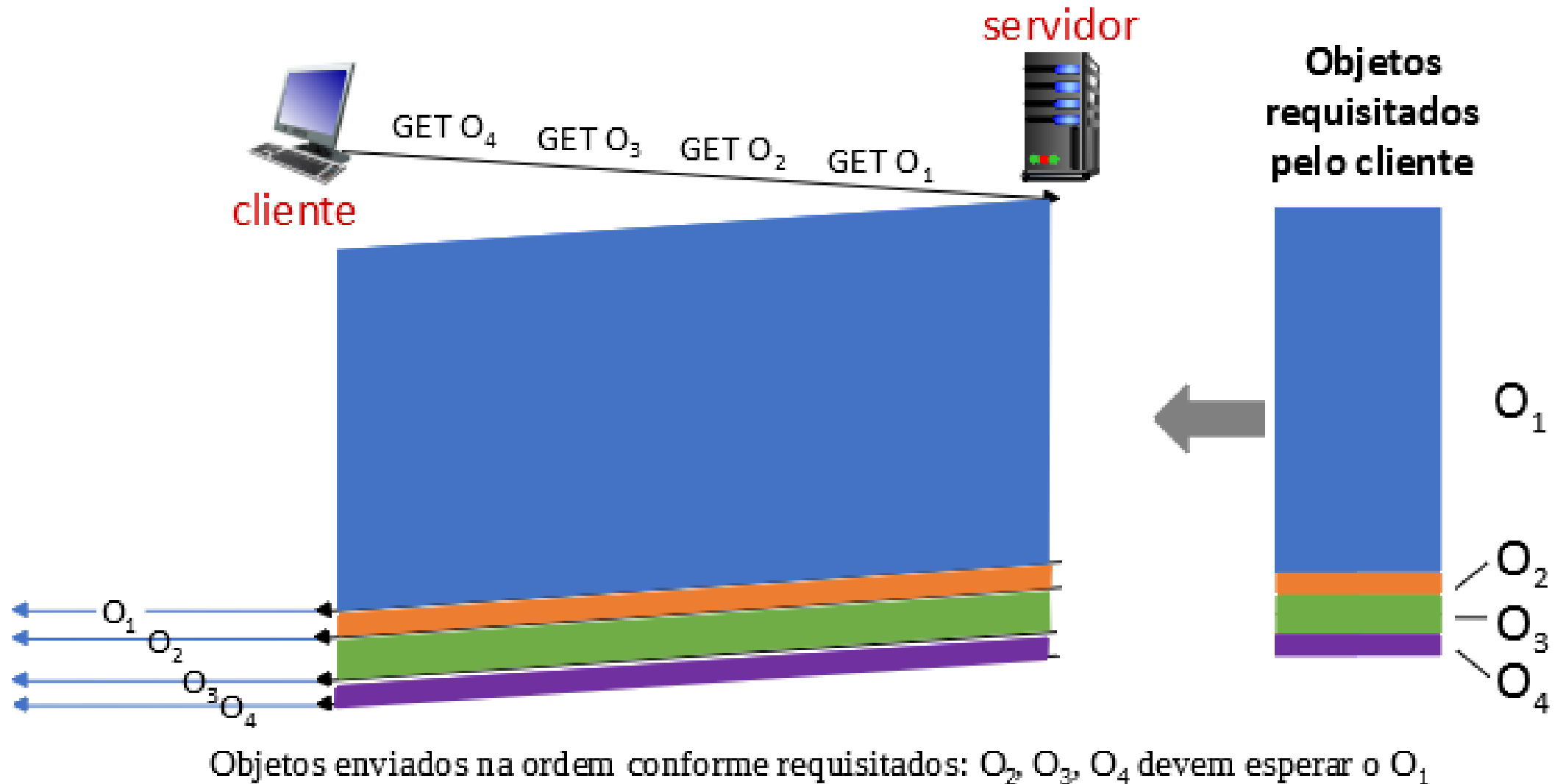
# Questões na funcionalidade de transporte da Internet

- TCP, UDP: principais protocolos de transporte por 40 anos
- Diferentes versões do TCP foram implementados para atender diferentes cenários

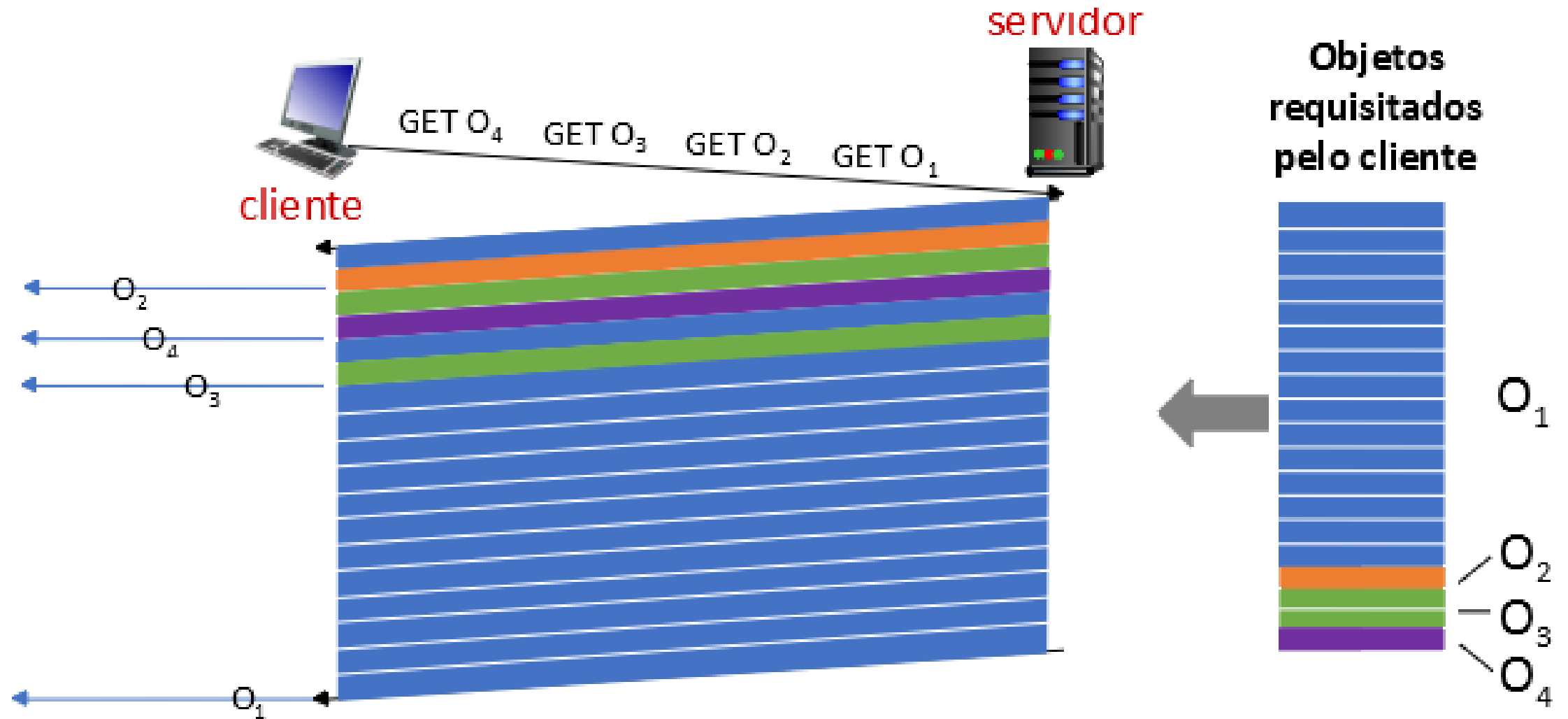
Cenário	Desafios
Pipes de redes longos e grandes (ou seja, transferências grandes de dados)	Muitos pacotes podem estar em trânsito, de forma que perdas em alguns podem ocasionar queda da conexão (pipeline)
Redes sem fio (wireless)	Perdas devido a interferências (ruídos) no canal; o TCP trata como perdas por congestionamento, o que leva a retransmissões
Enlaces com <i>delays</i> grandes (links transoceânicos, etc)	Valores de RTT extremamente longos (impacto nos timeouts e, ao final, nas taxas de transferência do TCP)
Redes de centros de dados (computação em nuvem)	Sensibilidade à latência

HTTP/3: QUIC : mover funções de nível de camada de transporte para a camada de aplicação

# HTTP/1 e problema do HOL (*Head-of-line blocking*)

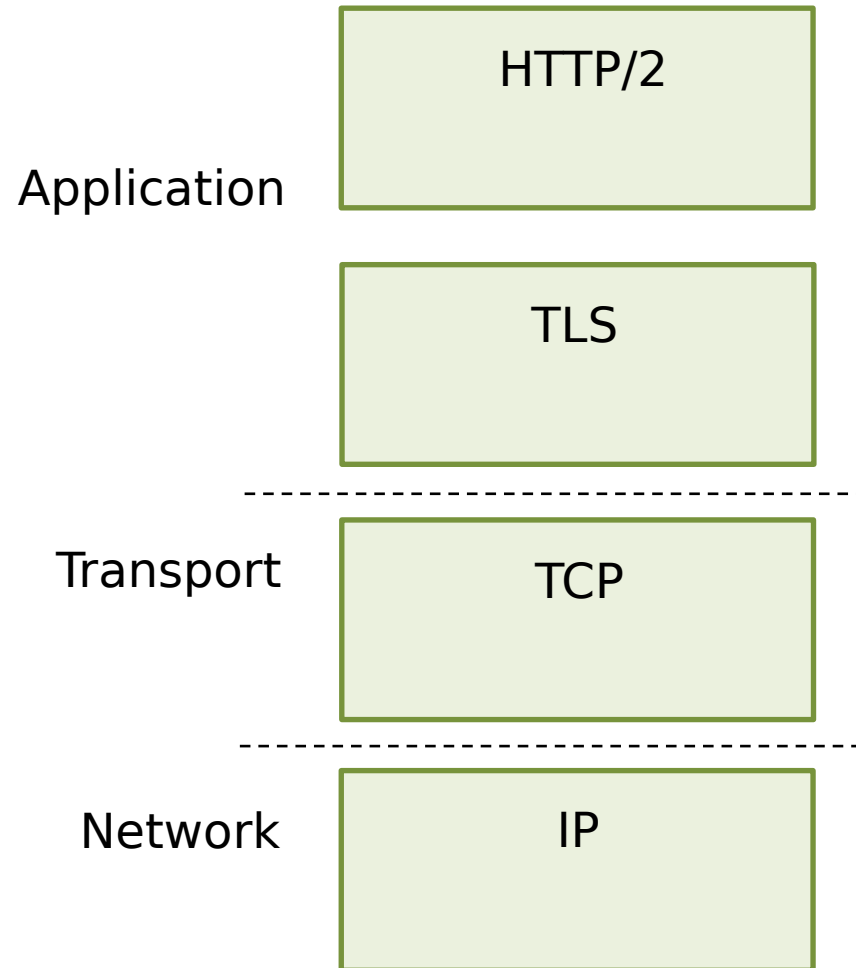


# HTTP/2 e solução para H0L (*framing*)



O<sub>2</sub>, O<sub>3</sub>, O<sub>4</sub> enviados rapidamente enquanto que O<sub>1</sub> é enviado ligeiramente atrasado

# HTTP/2 sobre TCP

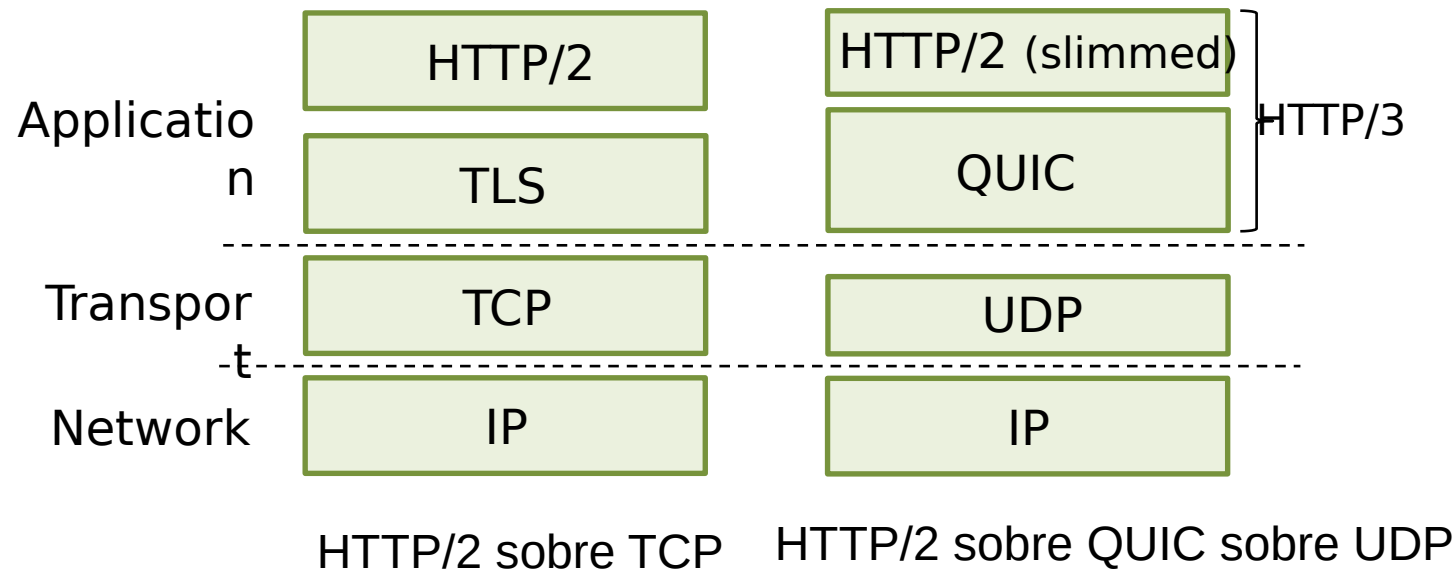


HTTP/2 over TCP

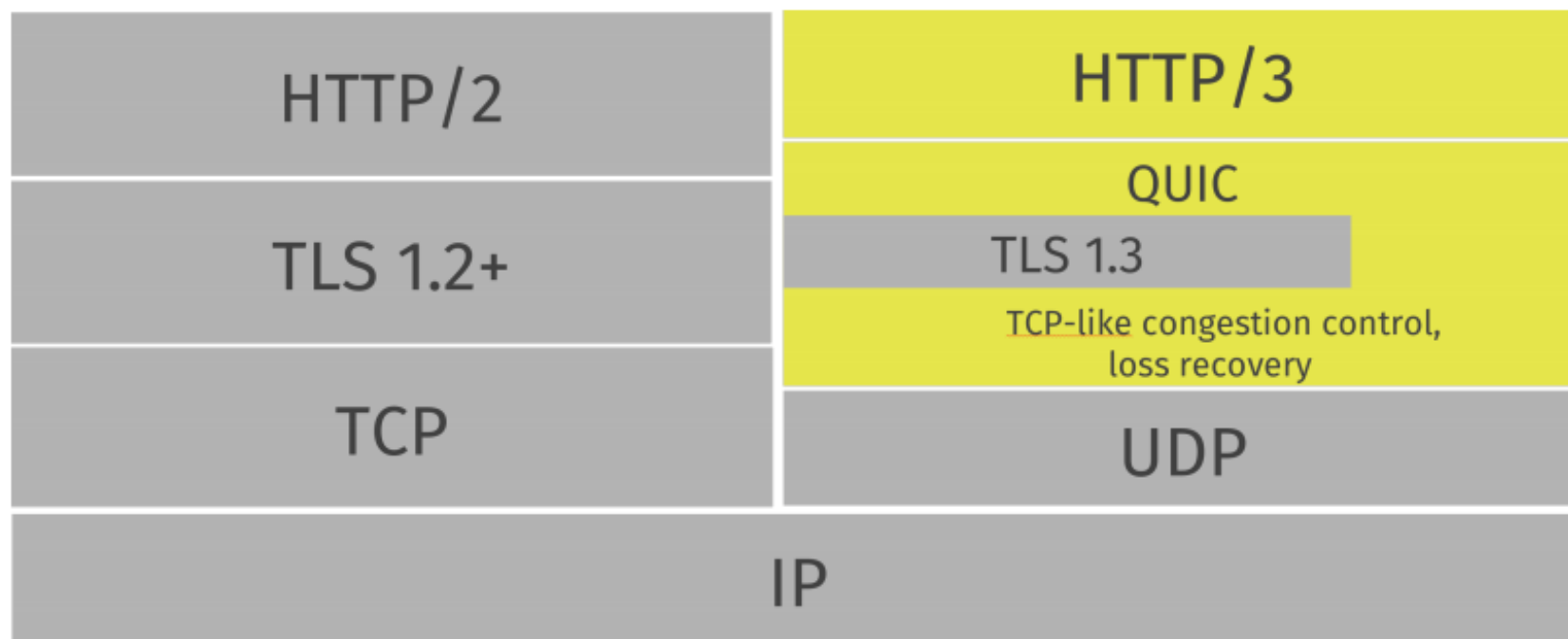
# Protocolo **QUIC** (Quick UDP Internet Connections)

Protocolo de camada de aplicação no topo do UDP para aumentar performance de transporte de rede

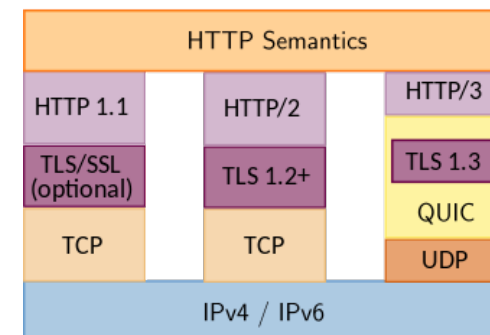
- Melhora da performance para o HTTP
- Especificado pela Google e usado em seus servidores e aplicações/apps (Chrome, mobile YouTube)



# QUIC (Quick UDP Internet Connections)



O **HTTP/3** é um *Proposed Standard* (RFC 9114 de Junho/2022) da IETF e especifica o QUIC como mecanismo de transporte.



**TLS 1.3** (2018): especificado pela RFC 8446 (<https://datatracker.ietf.org/doc/html/rfc8446>) e é a versão mais recente e atual e possui muitos recursos de segurança aprimorados (resolvendo algumas deficiências da versão TLS 1.2)

TLS 1.1 (2006): Deprecated in 2021 (RFC 8996)

# QUIC (Quick UDP Internet Connections)

O protocolo precisa adotar medidas para: estabelecimento de conexão, controle de erro, controle de fluxo e controle de congestionamento

**Controle de Erro e de Congestionamento:** *“Readers familiar with TCP’s loss detection and congestion control will find algorithms here that parallel well-known TCP ones.”* [from QUIC specification]

**Estabelecimento de Conexão:** confiabilidade, controle de congestionamento, autenticação, criptografia, estado da conexão estabelecido em uma mensagem com RTT

Múltiplos streams a nível de aplicação multiplexados sobre uma única conexão QUIC

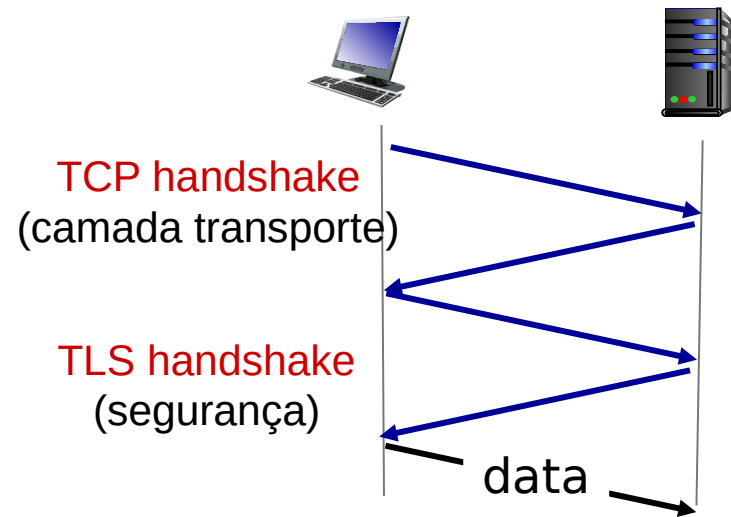
Transferência segura e confiável de dados

Controle de congestionamento: propostas já usadas no TCP



# Protocolo QUIC

## Comparativo no estabelecimento da conexão entre TCP+TLS e QUIC

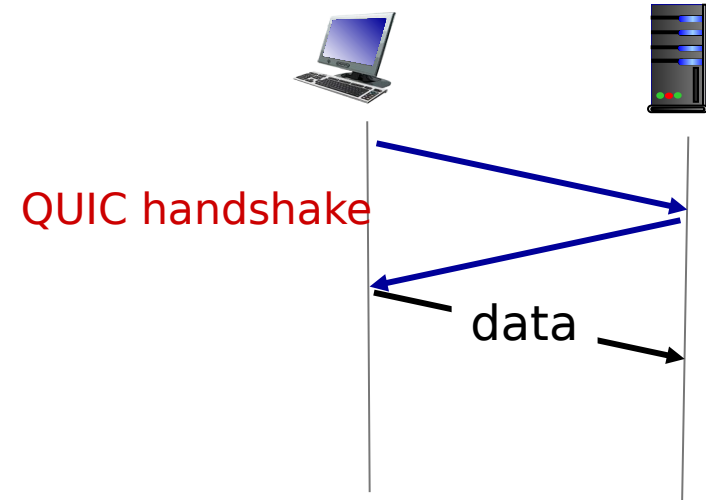


TCP (confiança, controle de congestionamento)

+

TLS (autenticação, estado da conexão cifrada)

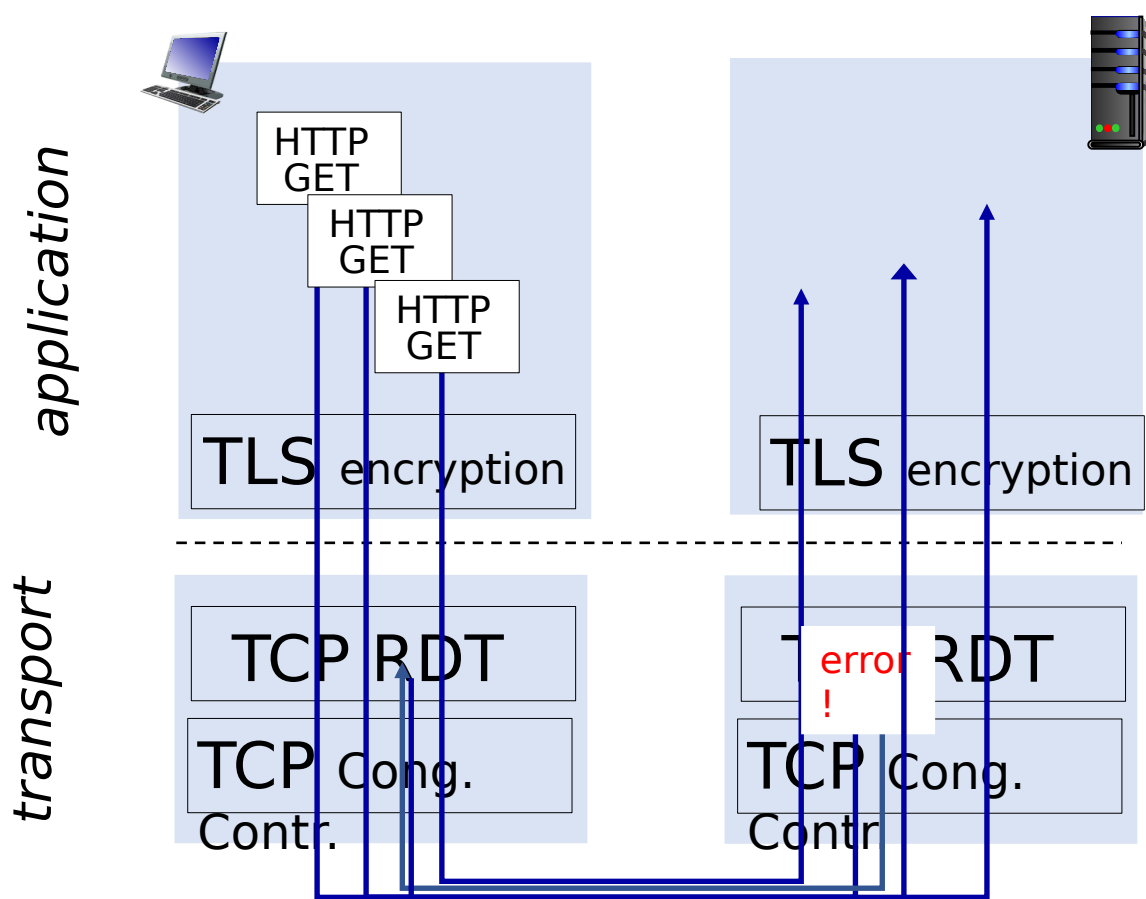
**2 handshakes em série**



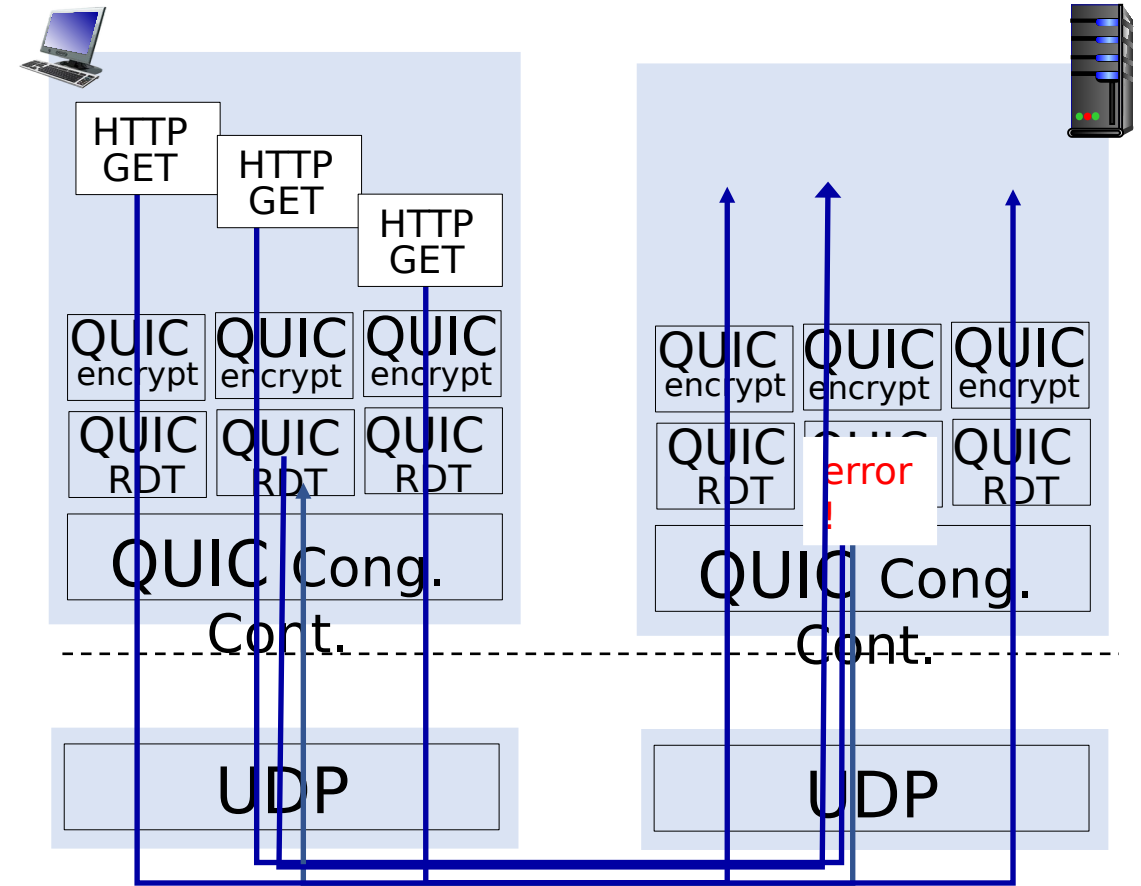
QUIC (confiança, controle de congestionamento, autenticação, estado de conexão cifrada)

**1 handshake**

# QUIC: streams: paralelismo e sem bloqueio HOL



(a) HTTP 1.1



(b) HTTP/2 com QUIC: sem bloqueio HOL

# Outros Protocolos de Transporte

**SCTP** (*Stream Control Transmission Protocol*) **RFC 4960**

- Transporte de dados de telefonia sobre IP
- Pedacos (chunks) de mensagens (mesma concepção do UDP)
- Vários caminhos (várias conexões em uma única associação SCTP): vários IPs para uma única associação de porta
- Pode ou não manter a ordem das mensagens

**DTLS** (Datagram Transport Layer Security) **RFC 4347**

- Datagramas com criptografia por TLS

# Outros Protocolos de Transporte

## **DCTCP** (*Data Center TCP*) **RFC 8257**

- Transporte de dados em centros de dados (datacenters), especificamente, em computação em nuvem
- Procura resolver questões de latência do TCP
- Requisitos de workloads: mensagens de query (sensíveis à latência); mensagens para coordenação e controle de estado – mensagens curtas (sensíveis à delay); atualização de dados – fluxos de dados grandes (sensíveis à desempenho)
- Como: usa os bits de ECN (Explicit Congestion Notification) no IP (logo, switches e roteadores precisam ser reconfigurados); alterações na definição do tamanho de janela (window); altera a forma como trata pacotes SYN, SYN-ACK e RST

# Problemas no TCP/IP

## **Security Problems in the TCP/IP Protocol Suite**

(BELLOVIN, 19–) Reprinted from Computer Communication Review, Vol. 19, No. 2, pp. 32-48, April 1989.

### **Initial Sequence Number - ISN**

Números de sequência (aleatoriedade): ataques de MITM (man-in-the-middle); spoofing ; connection spoofing ou connection forgery

Mitigação: RFC 1948 – Defending Against Sequence Number Attack

# Problemas no TCP/IP

## **Estabelecimento da conexão**

Ataques de SYN flood -> DOS e DDOS; também referenciado como ataque Naptha

Mitigação: firewalls, IDS, BGP (bloqueio de fluxos com origem estranha à rede)

## **Outros problemas de segurança**

Mecanismo de controle de congestionamento: ataque Low-Rate TCP-Targeted Denial of Service Attacks

Portas de origem: aleatorização

Injeção de RST junto com SYN

Opções incomuns e checksums errados

FIN-WAIT-2 flooding attack

# Problemas no TCP/IP

## **Vulnerabilidade no DTLS**

Quebra do cifragem

Plaintext-Recovery Attacks Against Datagram TLS  
(ALFARDAN; PETERSON, 2009)

## **Outros problemas de segurança**

Servidor processa conexões de entrada

UDP flood attack