# Tutorial
# ENTENDENDO O PROTOCOLO QUIC, O PROTOCOLO QUE ESTÁ MUDANDO A INTERNET!

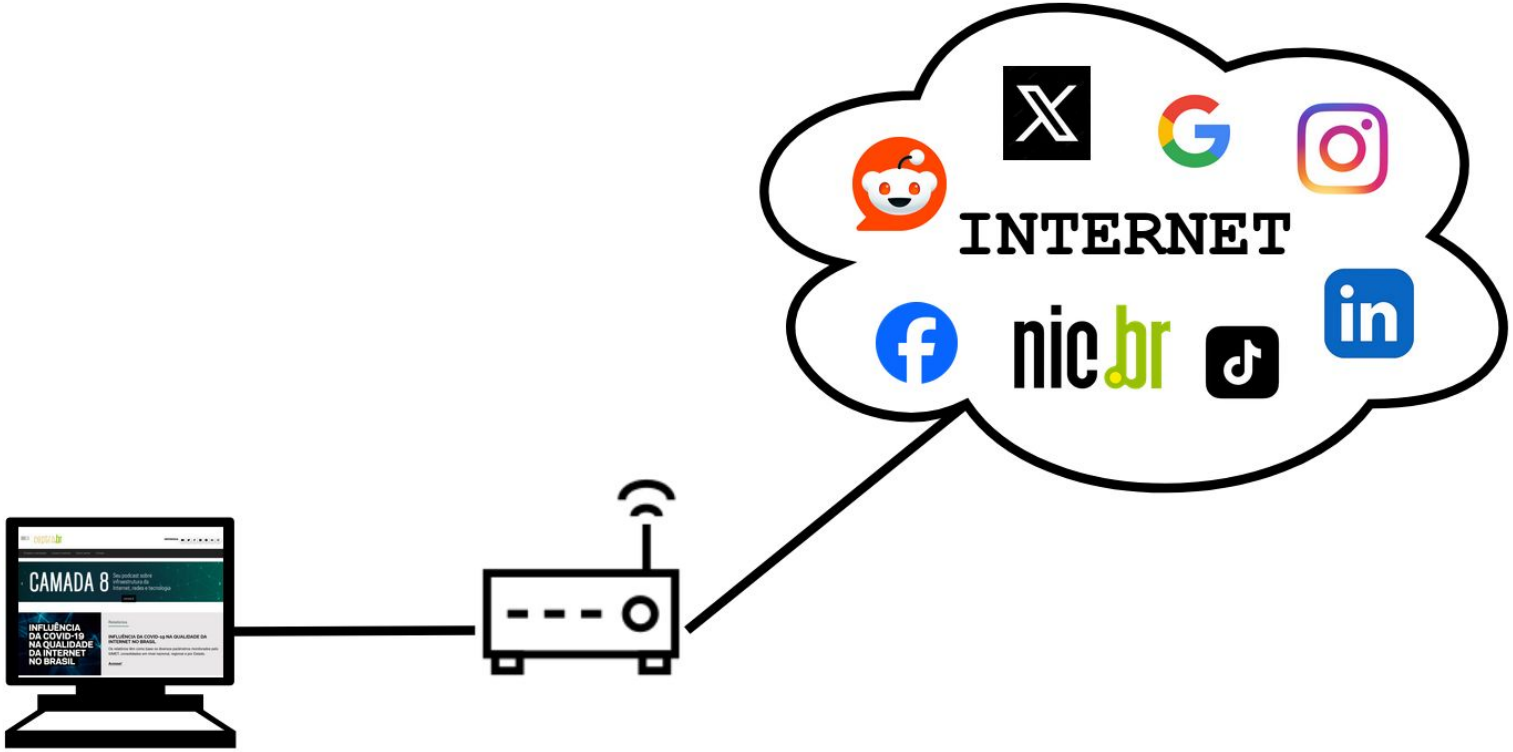
**Eduardo Barasal Morales**
**Lucas Jorge da Silva**
**Tiago Jun Nakamura**

ceptro.br nic.br cgi.br

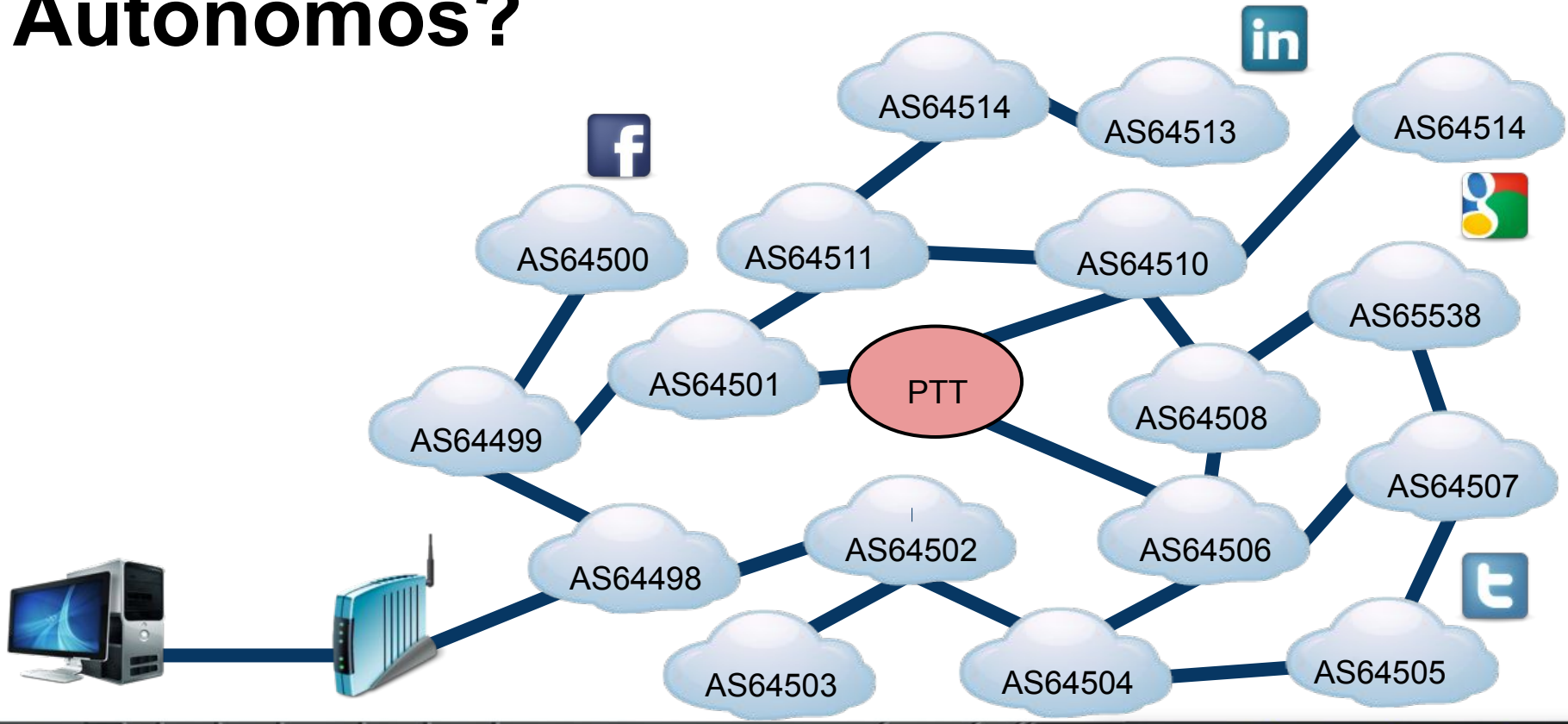
# Agenda

- Introdução a comunicação na Internet
- Camada de **Transporte**: TCP e UDP
- Camada de **Segurança**: TLS
- Camada de **Aplicação**: HTTP
- QUIC
- Utilização dos protocolos
- Laboratório

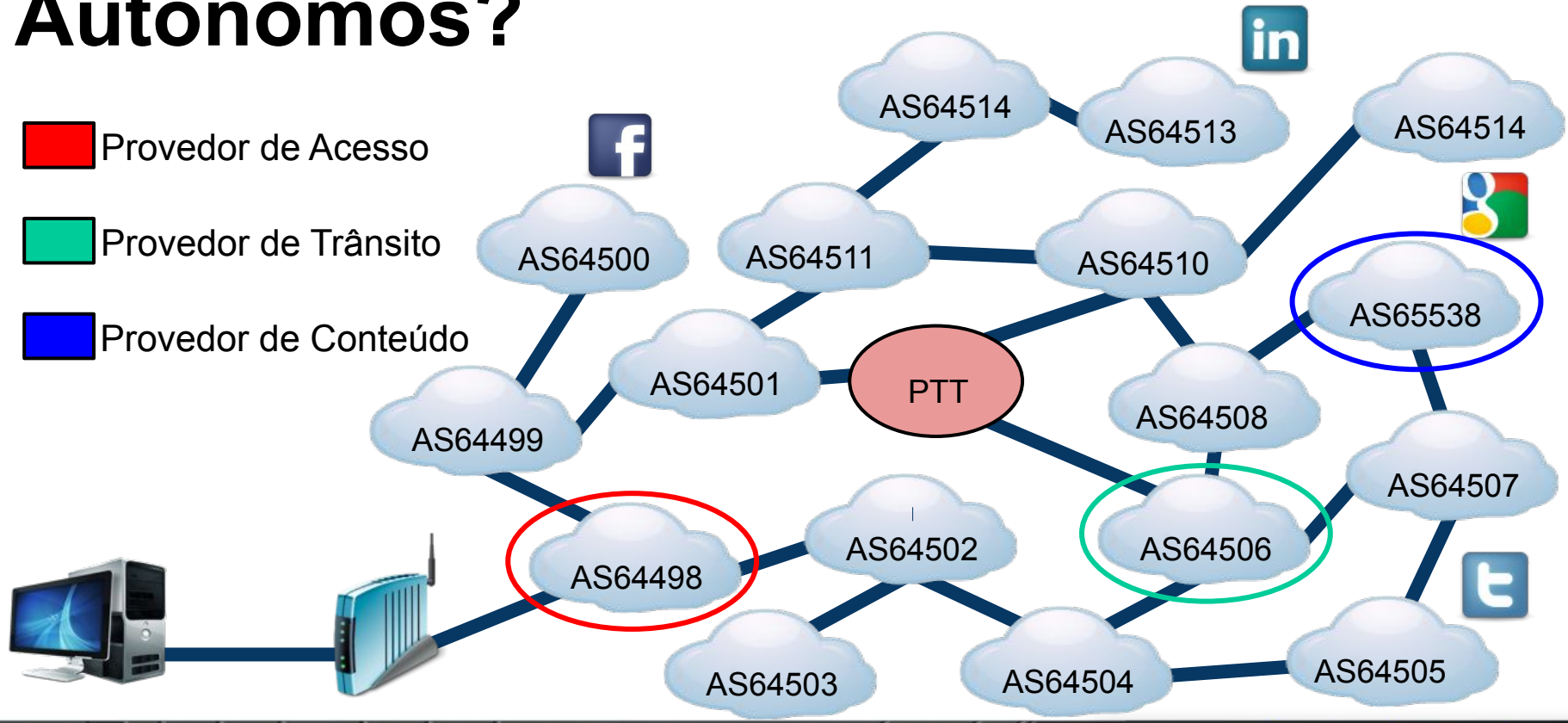
# Como a Internet funciona?



# Mas qual é a função dos sistemas Autônomos?



# Mas qual é a função dos sistemas Autônomos?



# Mas como duas máquinas se comunicam?


**Modelo TCP/IP**

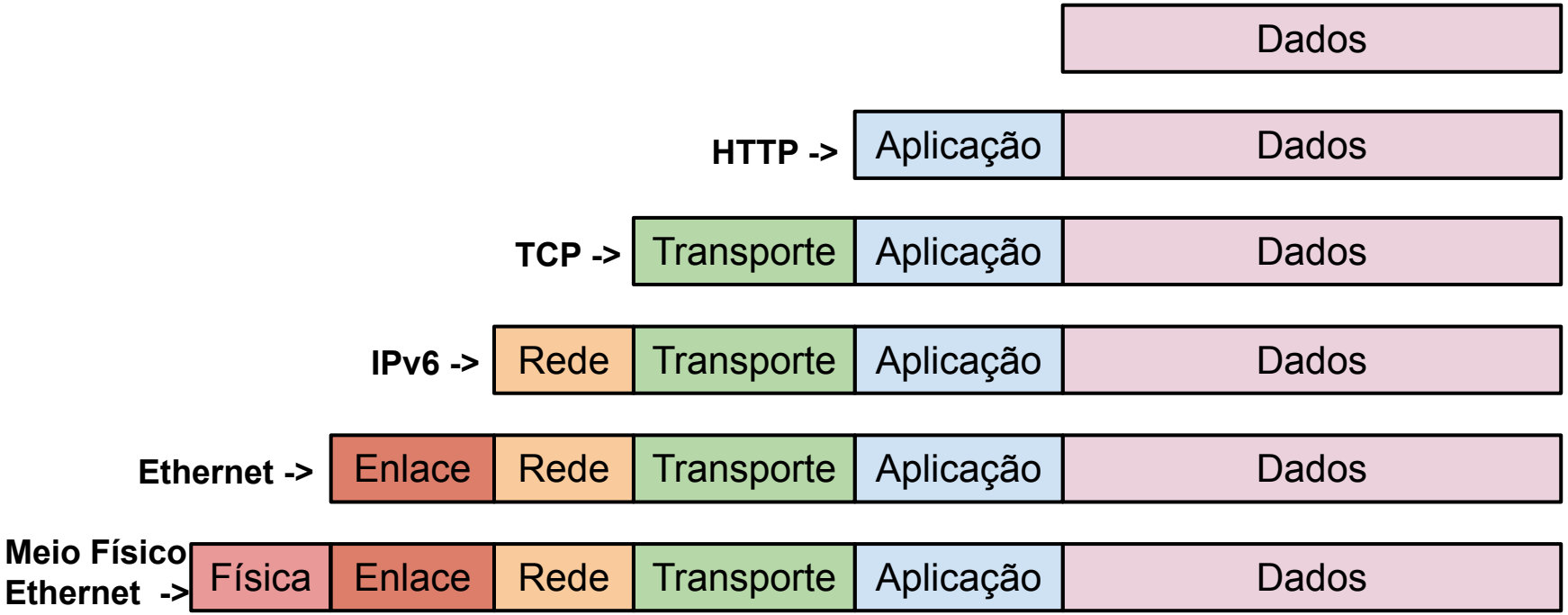
- 5 - Aplicação
- 4 - Transporte
- 3 - Internet
- 2 - Enlace
- 1 - Física

**1 - Acesso à Rede**

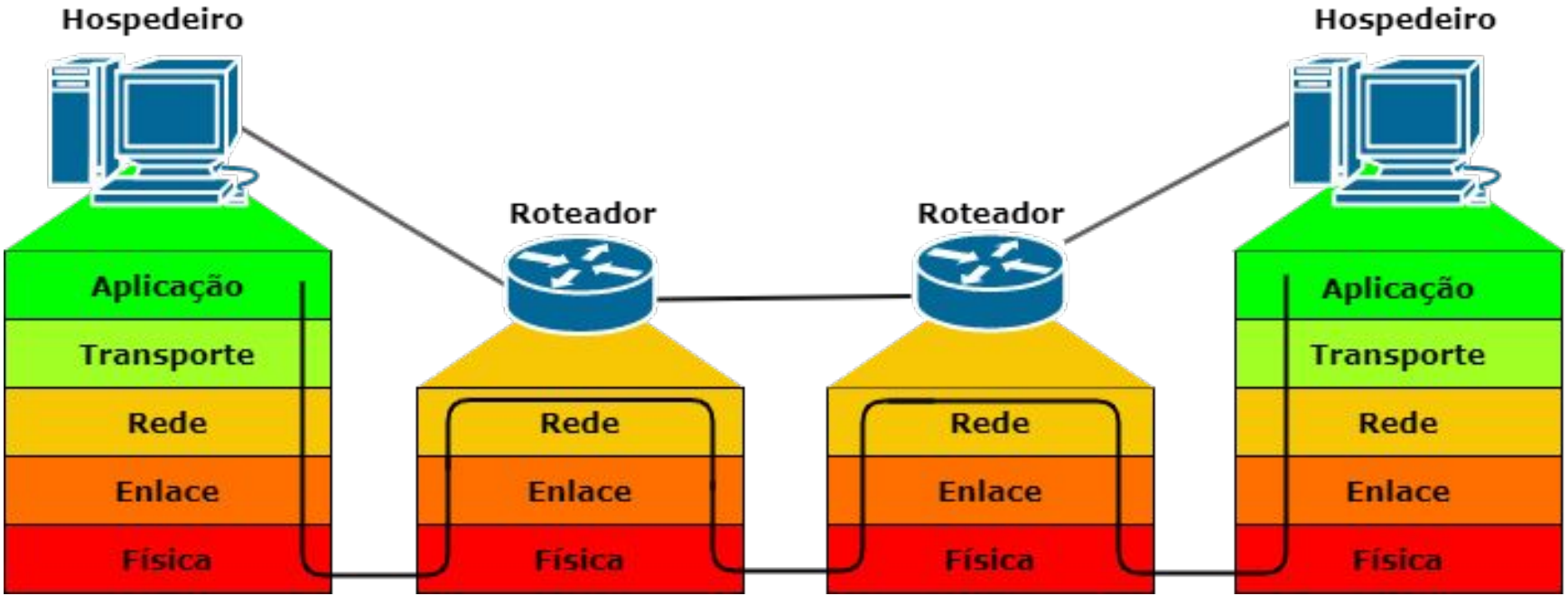
**Modelo OSI**

- 7 - Aplicação
- 6 - Apresentação
- 5 - Sessão
- 4 - Transporte
- 3 - Rede
- 2 - Enlace
- 1 - Física

# Mas como duas máquinas se comunicam?



# Mas como duas máquinas se comunicam?



# Camada de Transporte

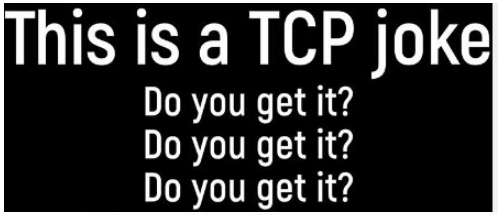
ceptro.br nic.br cgi.br

# Camada de Transporte: TCP

- Transmission Control Protocol (TCP) - 1981 - RFC 793
  - Orientado a conexão
  - Garantia de entrega
    - Livres de erro
    - Em seqüência
    - Sem perdas ou Duplicação
  - Three-way handshake

**Cabeçalho do protocolo TCP**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Endereço da porta de origem (16 bits) | | | | | | | | Endereço da porta de destino (16 bits) |
| Número de Sequência (32 bits) | | | | | | | | |
| Número de Confirmação (32 bits) | | | | | | | | |
| HLEN (4 bits) | Reservado (6 bits) | URG | ACK | PSH | RST | SYN | FIN | Tamanho da janela (16 bits) |
| Checksum (16 bits) | | | | | | | | Ponteiro de Urgência ou Urgent pointer (16 bits) |
| Opções e Preenchimento | | | | | | | | |



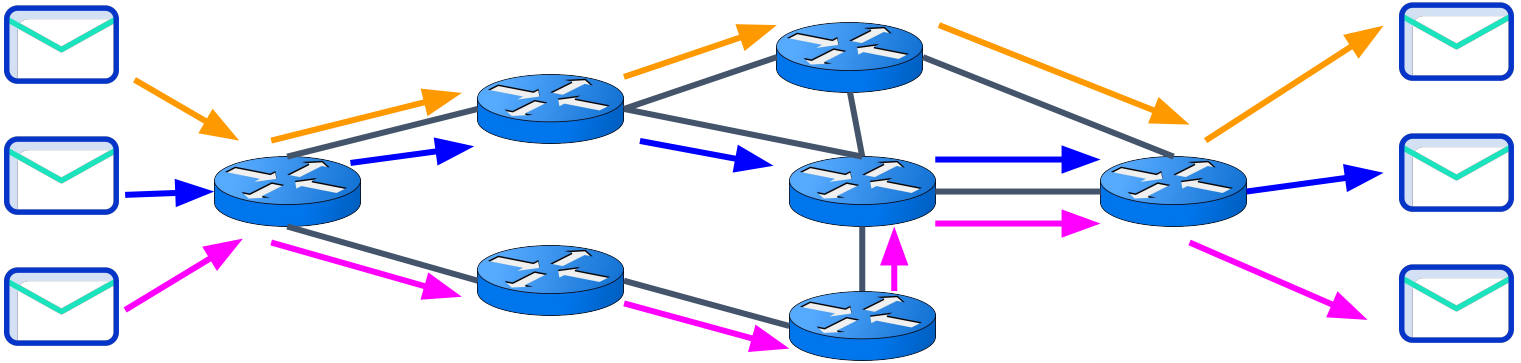
# TCP

Porta de Origem
Porta de Destino

**Internet vista pelo TCP**

Endereço IPv6 de Origem
Endereço IPv6 de Destino

**Internet vista pelo IPv6**



# Camada de Transporte: UDP

- User Datagram Protocol (UDP) - 1980 - RFC 768
  - Serviço do tipo “best-effort”
    - Entregas fora de ordem
    - Sem garantia de entrega
  - Não orientado à conexão
    - Cada segmento é independente
  - É simples e rápido!

**Cabeçalho UDP**

| | |
|---|---|
| Número da porta de origem (16 bits) | Número da porta de destino (16 bits) |
| Comprimento total (16 bits) | Checksum (16 bits) |

# Camada de Segurança

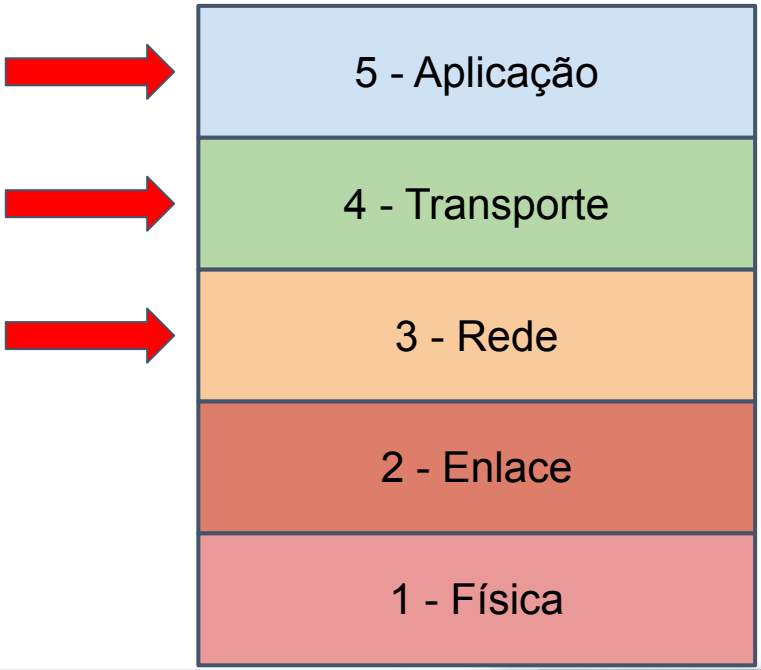
ceptro.br nic.br cgi.br

# Camada de Segurança

Modelo TCP/IP

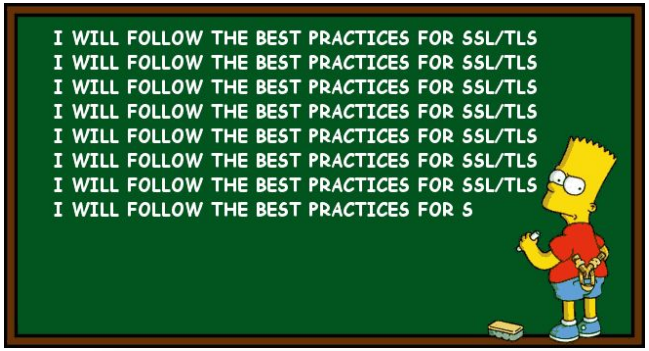
Onde encaixar a camada de Segurança?

Segurança

# TLS

- Transport Layer Security (TLS) - 1999 - RFC 2246 (versão 1.0)
- Evolução do protocolo SSL do Netscape
- Utiliza criptografia simétrica e assimétrica
- Garante confidencialidade, autenticidade, integridade e autenticação
- Possui versões 1.0, 1.1, **1.2 e 1.3**
- HTTP + TLS = HTTPS



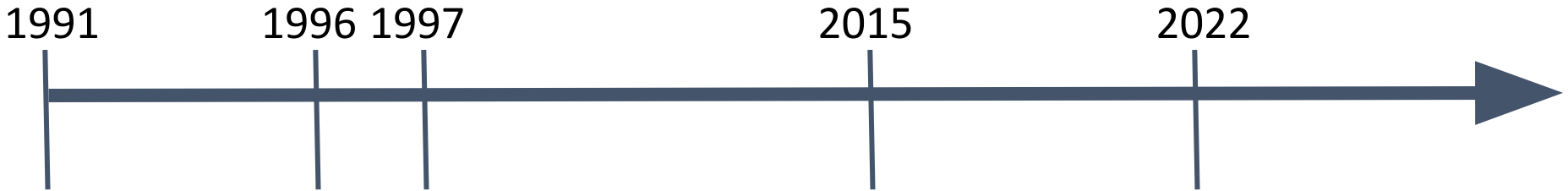
# TLS 1.3

- Melhora na latência e segurança
  - Handshake mais rápido
  - Remove características obsoletas e inseguras
    - Algoritmos  de criptografía - ex: MD5
    - Renegociação
  - Mais simples
  - Zero Round-Trip Time (0-RTT)

# Camada de Aplicação

ceptro.br nic.br cgi.br

# Breve história do HTTP




HTTP/0.9

HTTP/1.0

HTTP/1.1

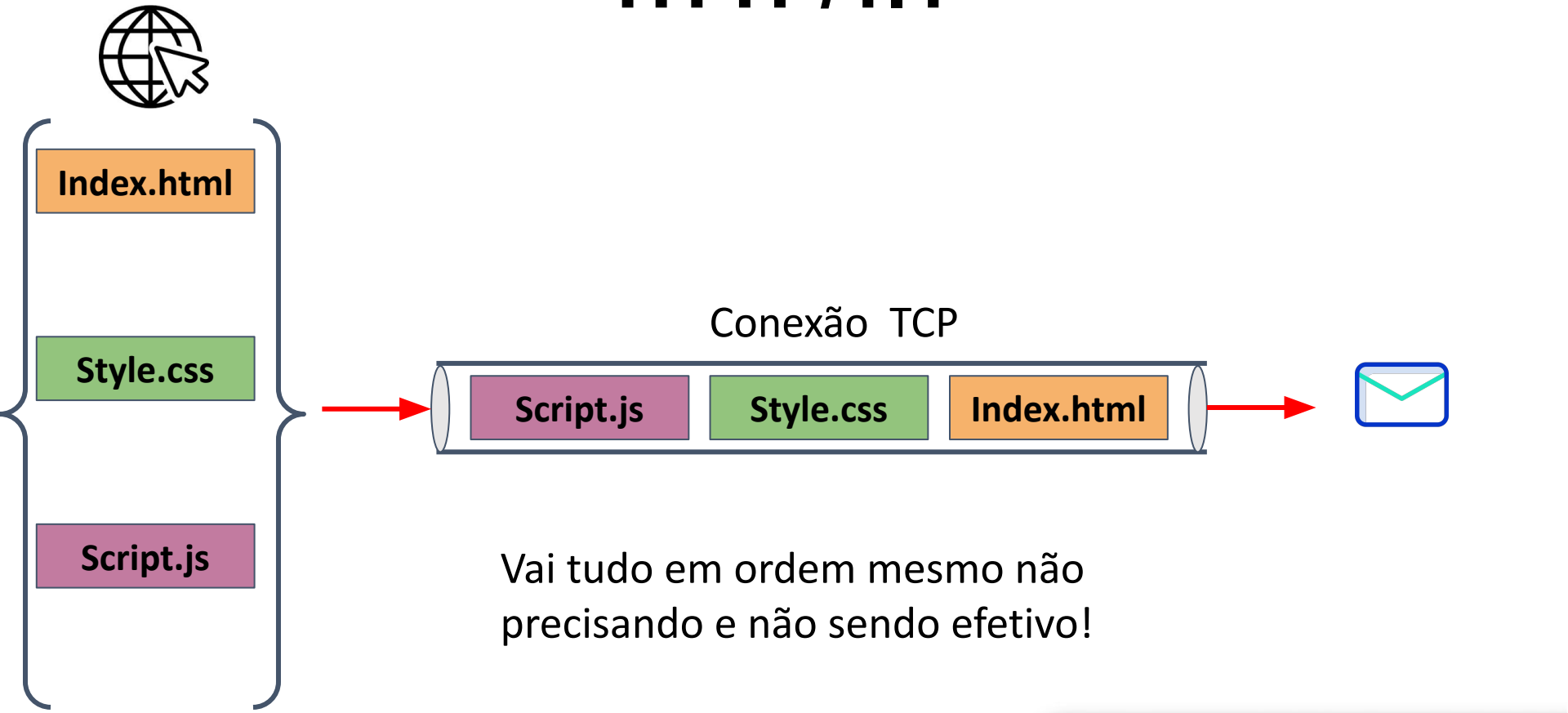
HTTP/2.0

HTTP/3.0

# Breve história do HTTP

| Versão | Lançamento | RFC | Principal Característica |
|---|---|---|---|
| HTTP/0. 9 | 1991 | Não tem | Criado no CERN, só tinha GET |
| HTTP/1.0 | 1996 | RFC 1945 | Cabeçalho, POST, Cookies, |
| HTTP/1. 1 | 1997 | RFC 9112 (Original RFC 2068) | Keep-alive |
| HTTP/2.0 | 2015 | RFC 9113 (Original RFC 7540) | Multiplexação, Server-side push, priorização, HPACK |
| HTTP/3.0 | 2022 | RFC 9114 | QUIC, TLS por default, QPACK |

# HTTP/1.1




Vai tudo em ordem mesmo não precisando e não sendo efetivo!

# HTTP/2

1.1

1.2

2.1

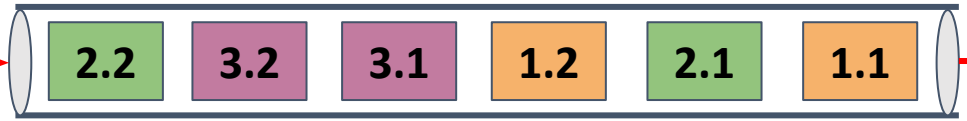
2.2

3.1

3.2

Melhoria implementada através da multiplexação do conteúdo!

Conexão TCP




Multiplexação em stream do HTTP/2

1

2

3

Index.html

Style.css

Script.js

# Problema: Head of line blocking

TCP 1 1.1

TCP 2 2.1

TCP 3 1.2

TCP 4 3.1

TCP 5 3.2

TCP 6 2.2

Conexão TCP

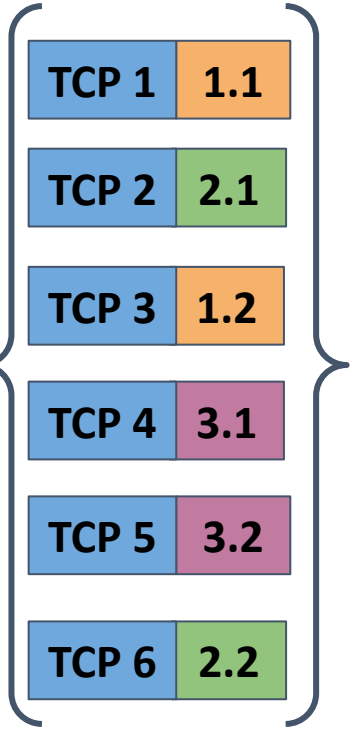
TCP 1 1.1

Multiplexação em stream do HTTP/2

1 2 3

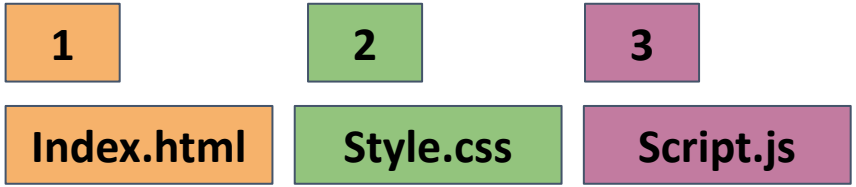
Index.html Style.css Script.js

# Problema: Head of line blocking

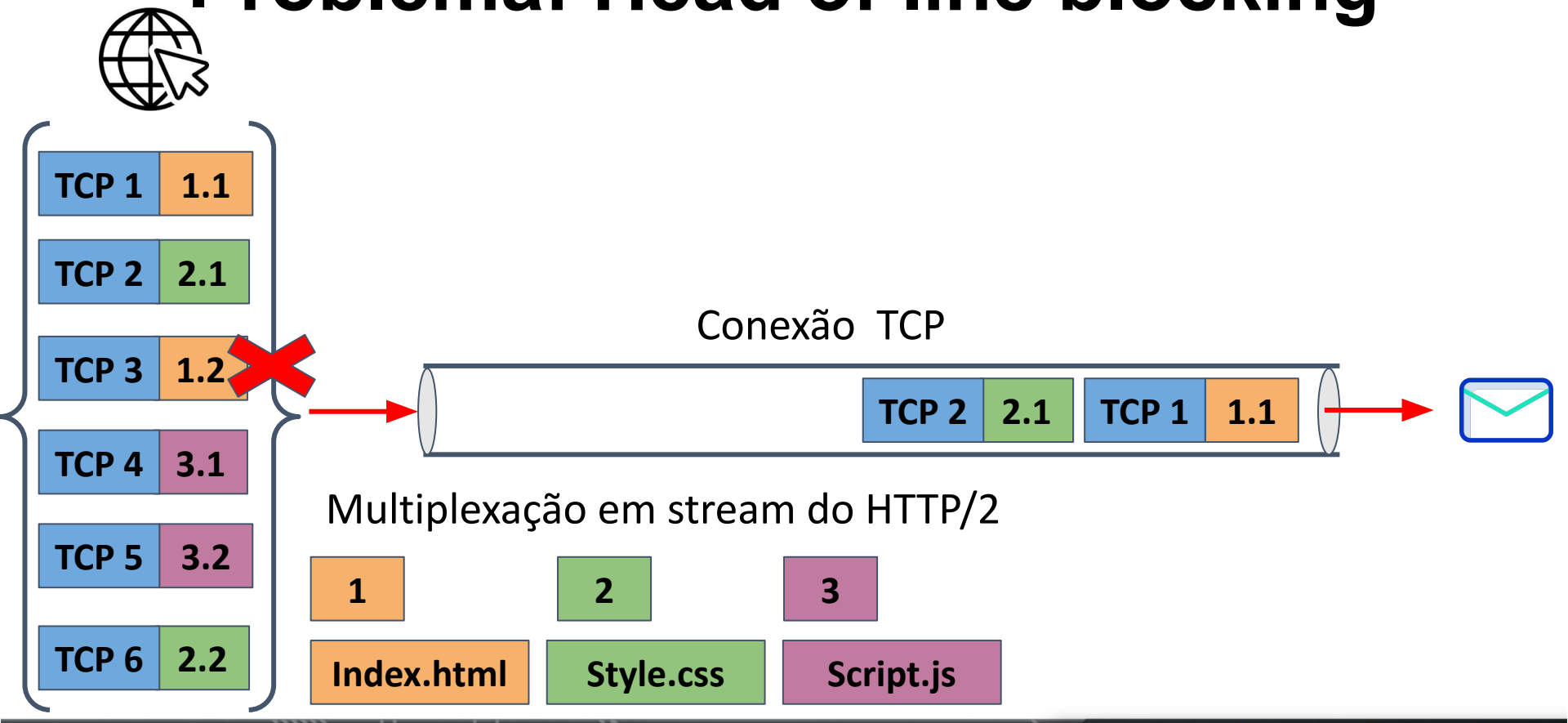



Conexão TCP

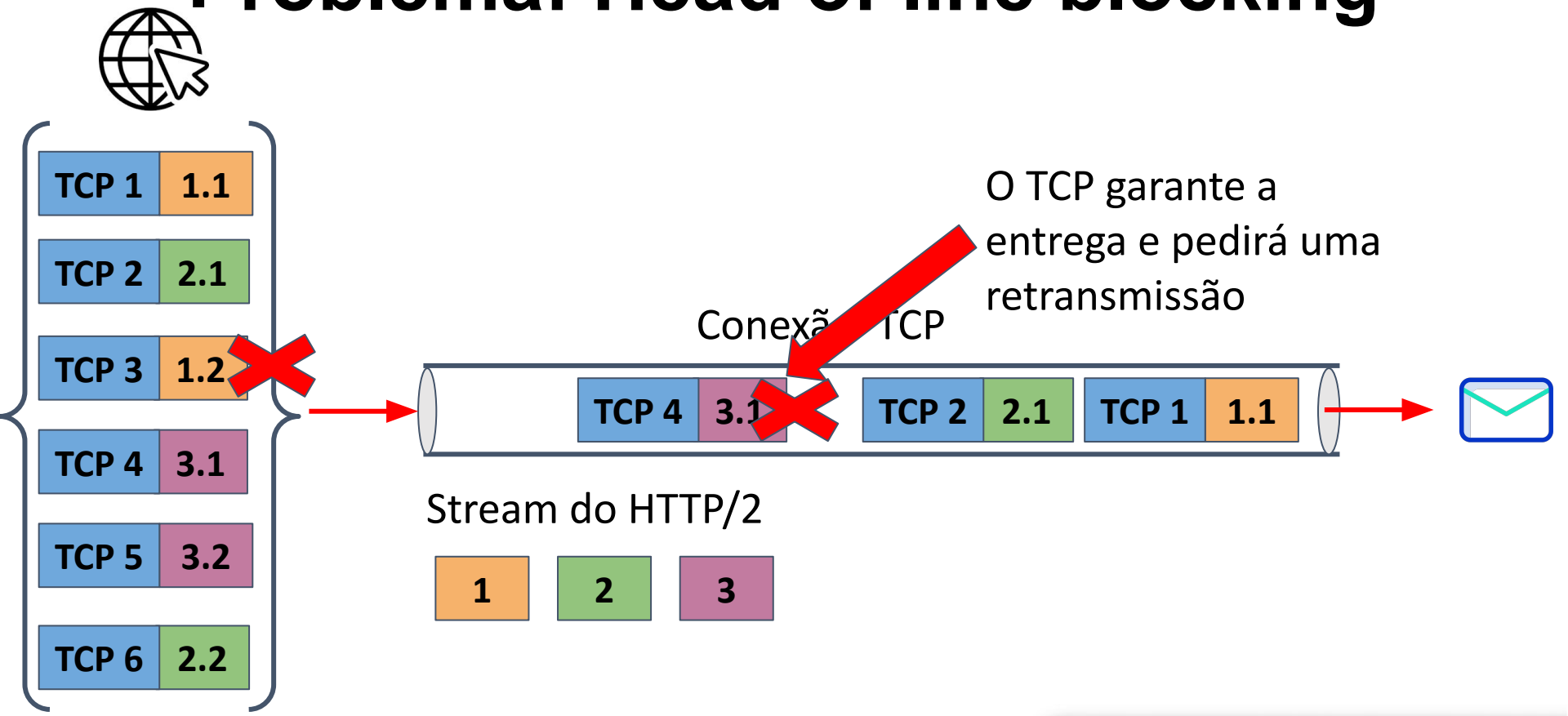

Multiplexação em stream do HTTP/2



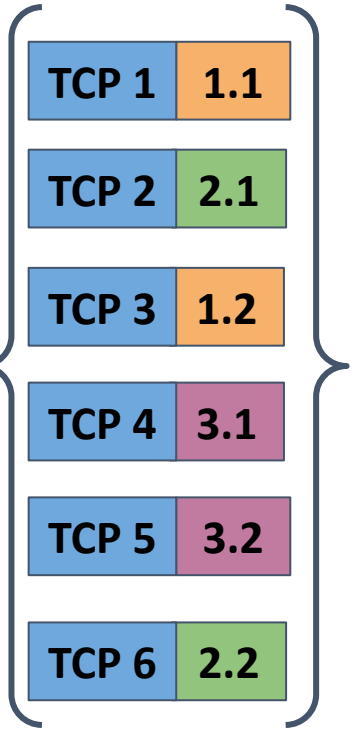
# Problema: Head of line blocking



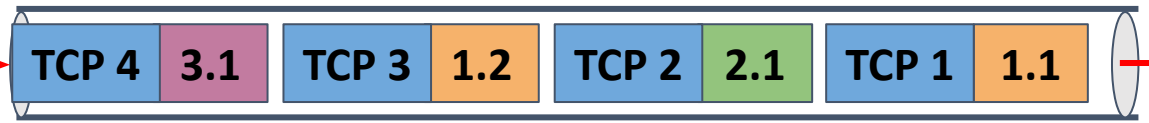
# Problema: Head of line blocking



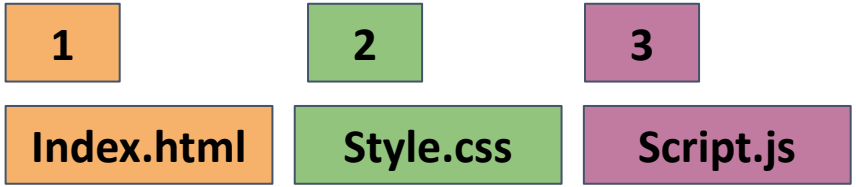
# Problema: Head of line blocking



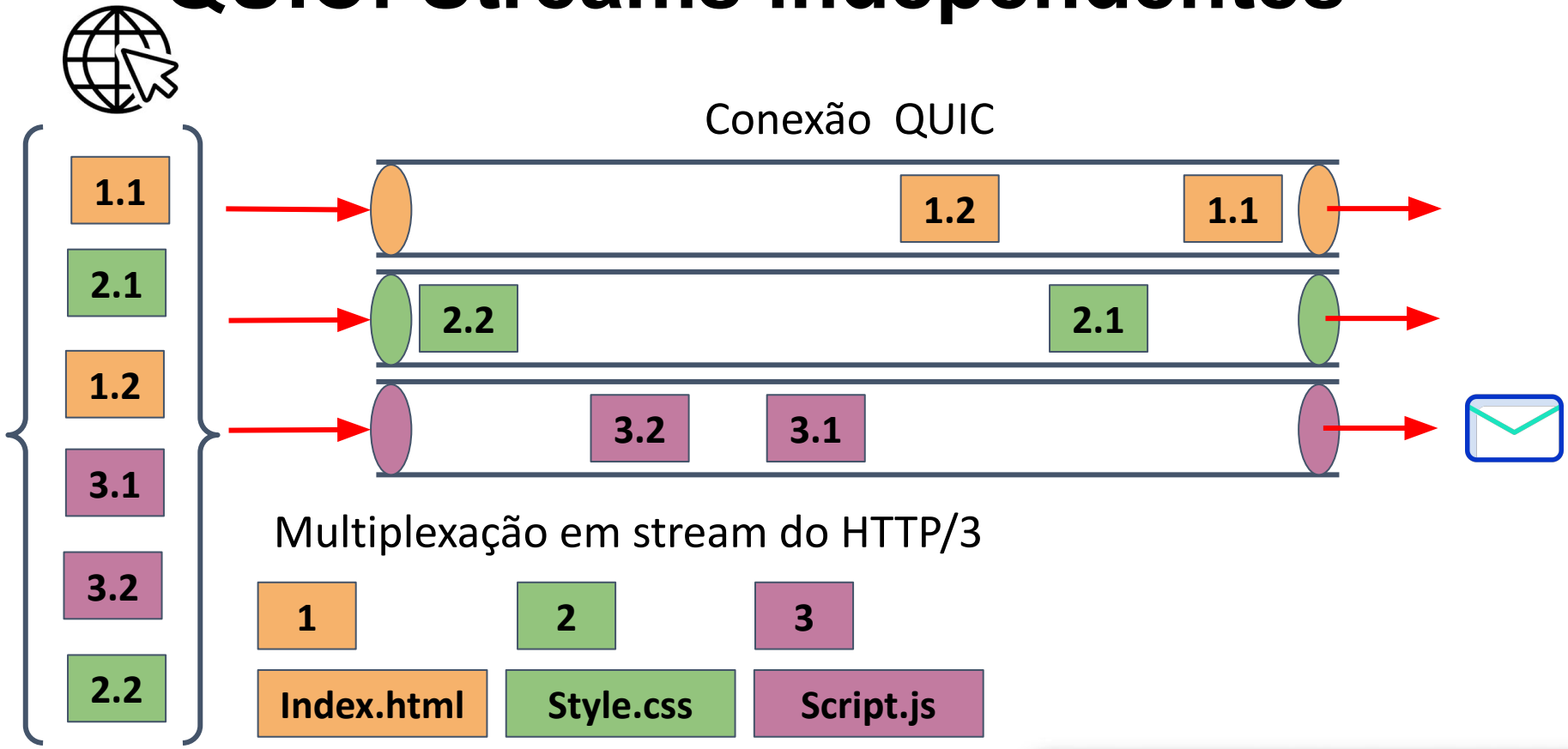

Conexão TCP



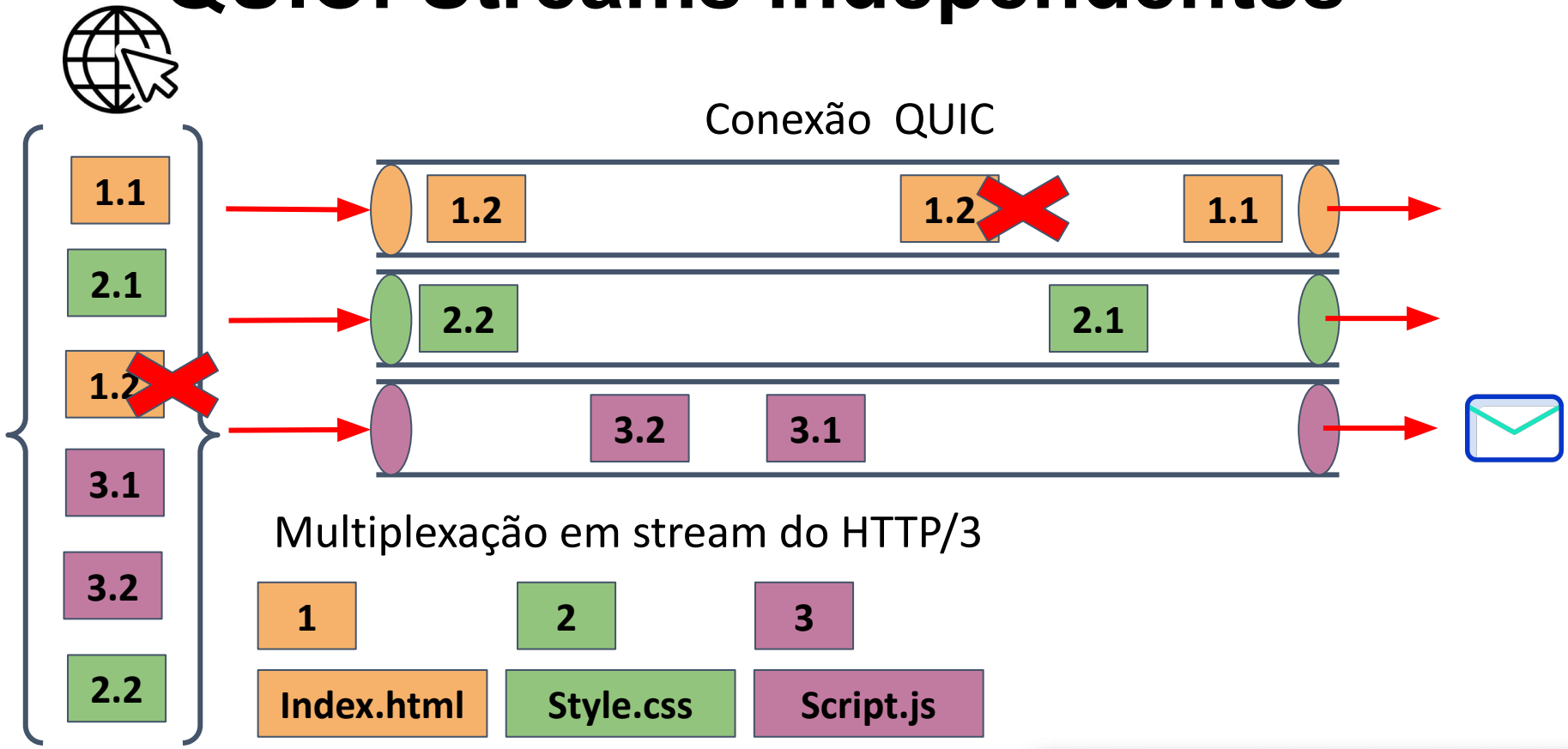

Multiplexação em stream do HTTP/2



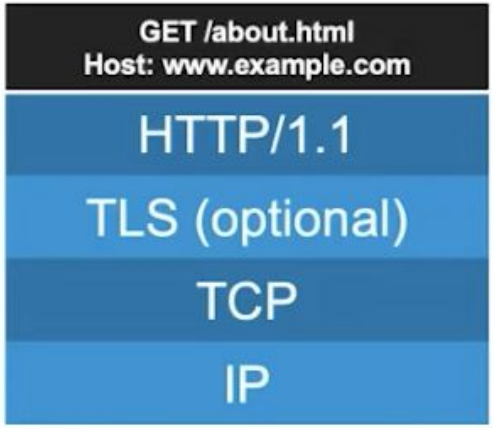
# QUIC: Streams independentes



# QUIC: Streams independentes

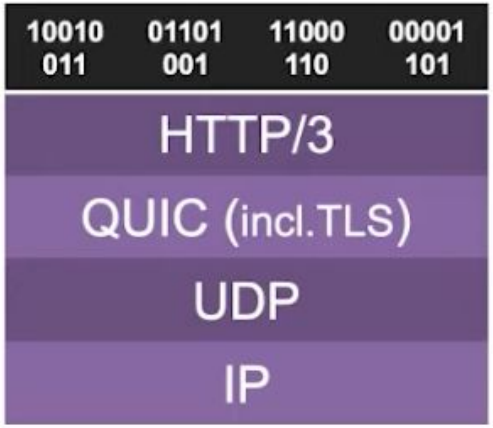


# Versões do HTTP




Browser faz várias conexões TCPs

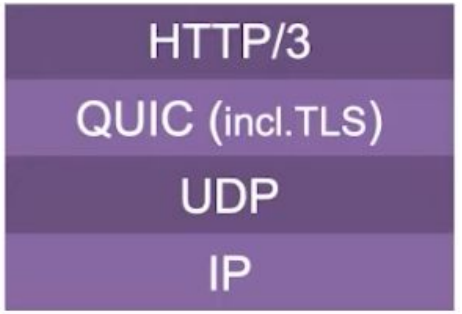

Browser faz uma requisição TCP, mas divide o conteúdo em Streams




Browser usa o QUIC e o HTTP3, para dividir o conteúdo em Streams

# HTTP/3

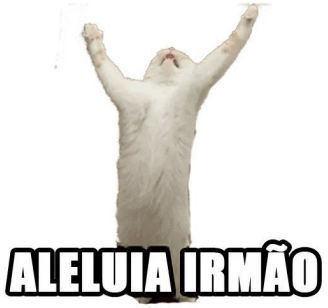
- Conhecido anteriormente como HTTP-over-QUIC
- Criado para resolver alguns problemas do HTTP/2:
  - Desempenho lento em situações em que o dispositivo muda de uma rede Wi-Fi para uma rede móvel (4G/5G)
  - Diminuir o impacto na conexão quando há perda de pacotes



# QUIC

# Por que criar o Quic

- Desejo de criar uma alternativa ao TCP tão utilizado mas mantendo algumas semelhanças
  - Menor latência
  - Menor handshake
  - Incorporando segurança
  - "Quick UDP Internet Connections" nome inicial
    - Mas que hoje não é mais utilizado

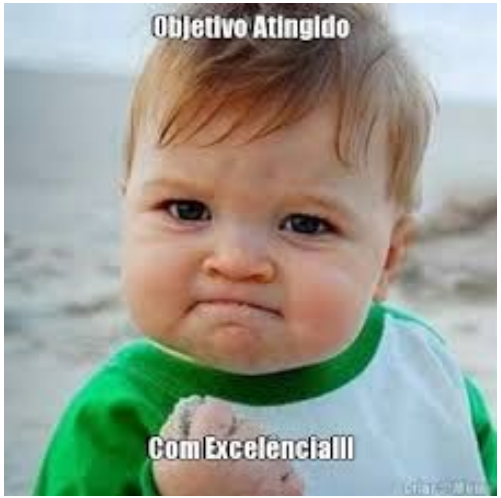


# Protocolo QUIC


- Foi desenvolvido **2012** pelo **Google**
  - Public release no Chromium version 29 em Agosto de **2013**
- **Lançado pelo IETF na RFC 9000 em Maio de 2021 a versão 1**
  - O nome do protocolo QUIC é definido
- **Lançada a versão 2 do QUIC RFC 9369 em Maio de 2023**
  - Parecido com o primeiro mas com algumas coisas extras
- **Novas RFCs e drafts estão sendo lançados no Working Group**

# Protocolo QUIC

- Seus objetivos são
  - Controlar de fluxos para envio de dados
  - Fornecer baixa latência no estabelecimento de conexões
  - Permitir migração de caminho de rede de forma transparente
    - Reconexão de NAT
    - Mudança de rede
  - Incluir medidas de segurança.

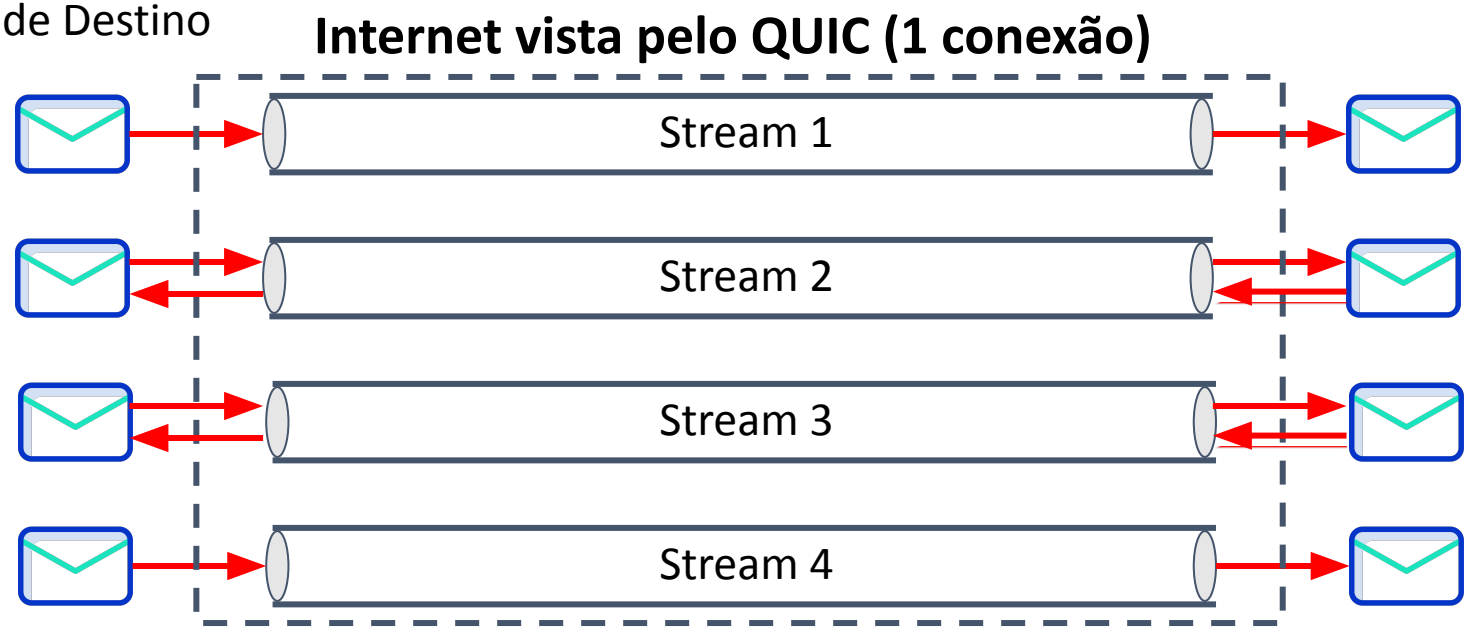


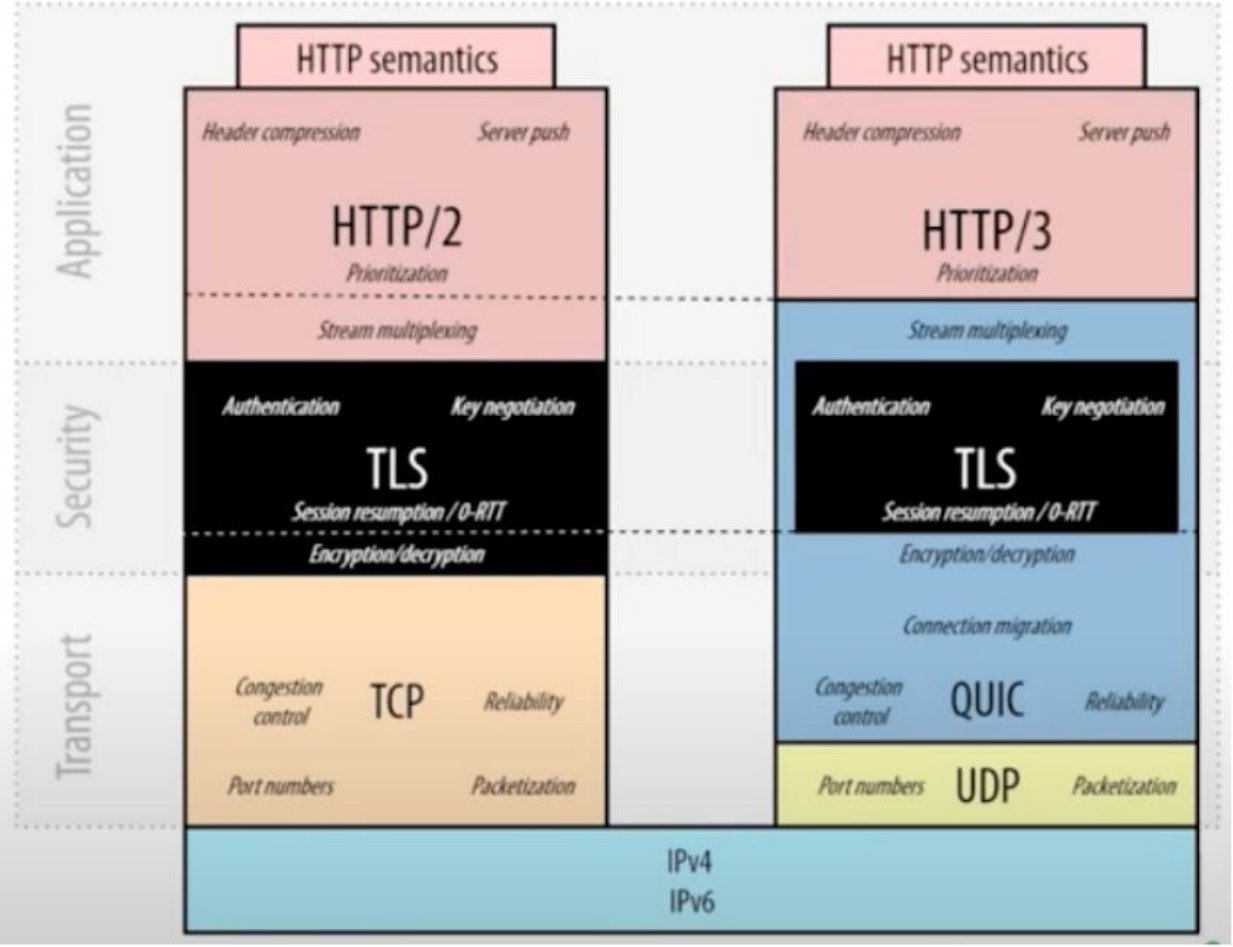
# Protocolo QUIC

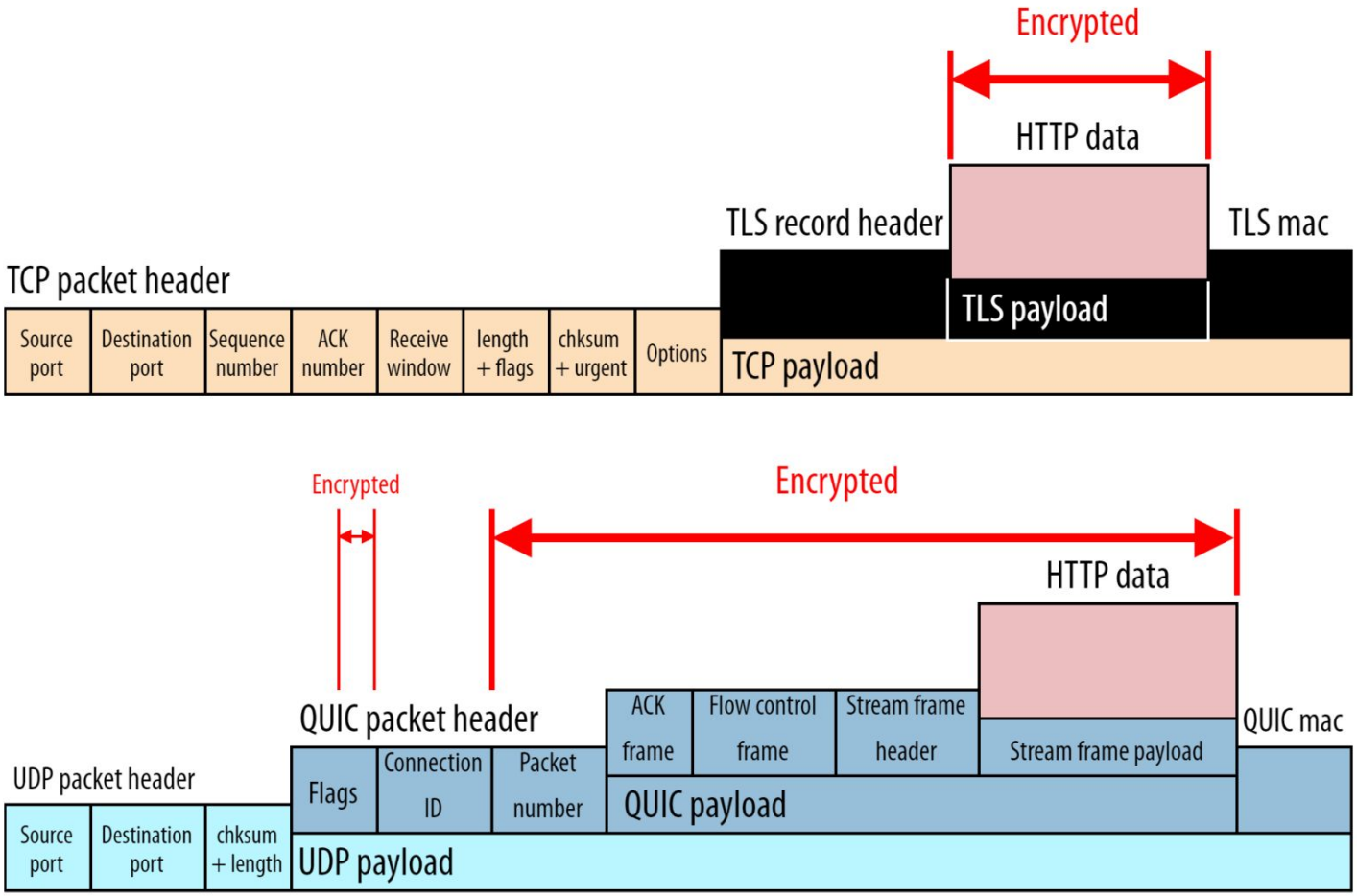
- É um protocolo seguro de camada de Transporte
  - Alguns podem dizer que ele é de camada "4,5"
    - Funciona em cima de UDP
- Opera de maneira stateful a comunicação entre um cliente e um servidor
- Possui um handshake estruturado para permitir a troca de dados o mais rápido possível

# QUIC

Porta de Origem
Porta de Destino




Application
Security
Transport
HTTP semantics
Header compression
Server push
HTTP/2
Prioritization
Stream multiplexing
Authentication
Key negotiation
TLS
Session resumption / 0-RTT
Encryption/decryption
Congestion control
TCP
Reliability
Port numbers
Packetization
HTTP semantics
Header compression
Server push
HTTP/3
Prioritization
Stream multiplexing
Authentication
Key negotiation
TLS
Session resumption / 0-RTT
Encryption/decryption
Connection migration
Congestion control
QUIC
Reliability
Port numbers
UDP
Packetization
IPv4
IPv6


Encrypted
HTTP data
TLS record header
TLS mac
TCP packet header
Source port
Destination port
Sequence number
ACK number
Receive window
length + flags
chksum + urgent
Options
TLS payload
TCP payload
Encrypted
Encrypted
HTTP data
QUIC packet header
ACK frame
Flow control frame
Stream frame header
Stream frame payload
QUIC mac
UDP packet header
Flags
Connection ID
Packet number
QUIC payload
Source port
Destination port
chksum + length
UDP payload

# Pontos de atenção (QUIC+TLS1.3+HTTP/3)

- Complexidade de implementação.
- Criptografa mais dados:
  - Dificulta o processo de *troubleshooting*.
  - Falta de informação dificulta a identificação de ameaças suspeito.
- Maior uso de recursos do servidor.
  - Alguns equipamentos ainda não estão otimizados para o tráfego

# QUIC: Por que esta estrutura?

ceptro.br nic.br cgi.br

# Ossificação

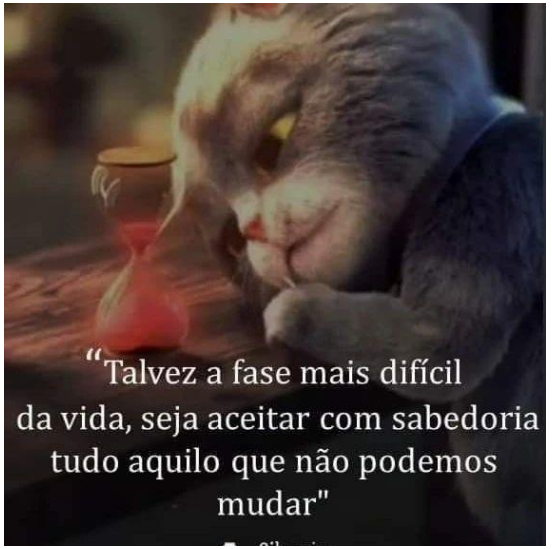
- Internet
  - É composta de várias caixas intermediárias - Roteadores, firewall, balanceadores de carga, NATs …
  - É difícil que todos implantem um novo protocolo
    - São mudanças de Configuração e de Software
    - Exemplo: IPv6 temos mais de 25 anos e só agora passamos de 50% no Brasil

# Princípios Básicos da Internet

- Núcleo da rede só cuida do encaminhamento dos pacotes!
- A inteligência, os protocolos complexos , as novas funcionalidades, são implementadas nas extremidades, por qualquer um, sem precisar pedir permissão a terceiros. Não é preciso pedir permissão para a inovação, para criar novas aplicações.

# Por que o QUIC foi desenvolvido usando UDP

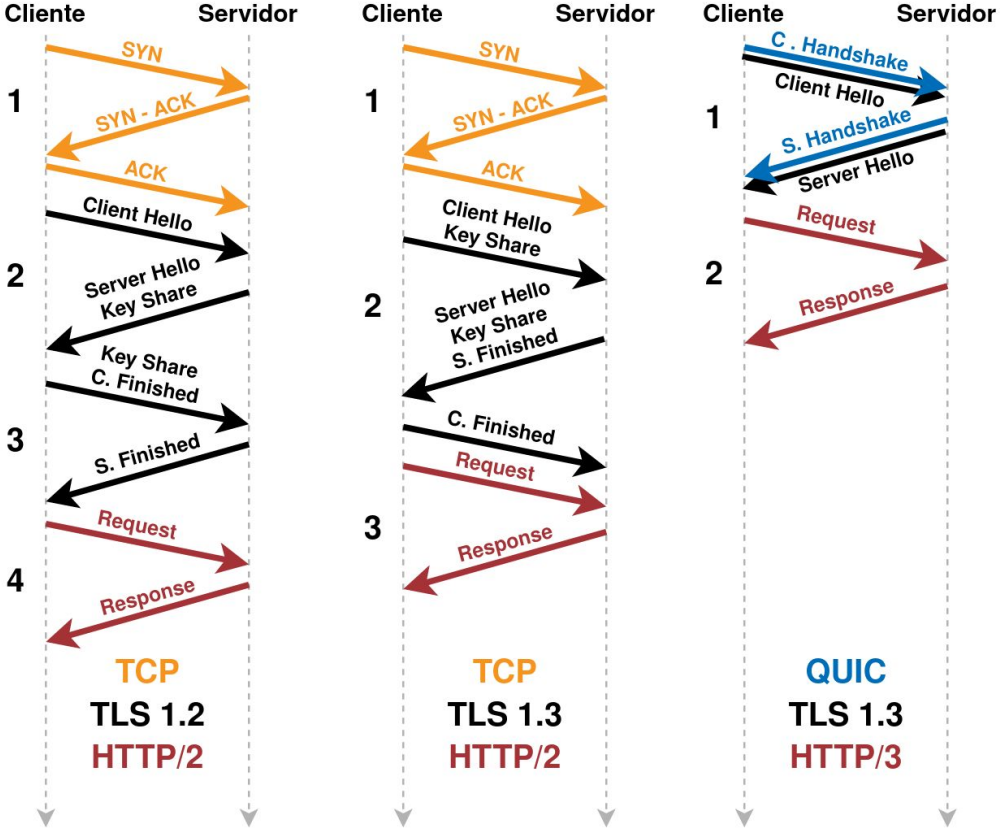
- Mas o QUIC fica na camada de transporte não precisa que todos implantem
  - IPv6 tem o campo "Próximo cabeçalho" e IPv4 tem o campo "Protocolo" - que precisavam indicar o novo protocolo
  - E as máquinas das pontas precisam implantar nos sistemas Operacionais - não mexe no kernel e sim no User Space

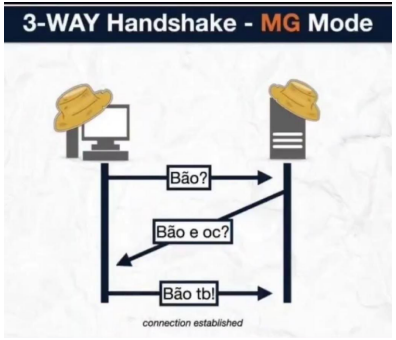


# QUIC: Handshake

ceptro.br nic.br cgi.br

# Comparativo de HandShake



# QUIC: Migração de redes

ceptro.br nic.br cgi.br

# Migração de Redes

IPv6 de origem
**2001:db8:1234::1**

# Migração de Redes

Saindo do WiFi e
indo para o 5G

# Migração de Redes

IPv6 de origem
**2001:db8:5678::1**

# Migração de Redes

- Identificação da conexão TCP
  - Endereço IPv6 de Origem e Destino
  - Porta Origem e Destino
- Identificação da Conexão QUIC
  - ID da Origem e do Destino
  - Se mantém a privacidade

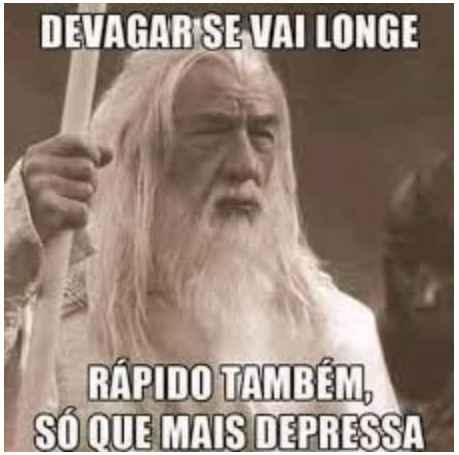

**Logo a mudança de rede no QUIC, não ocasiona em perda de comunicação**

# QUIC: 0-RTT

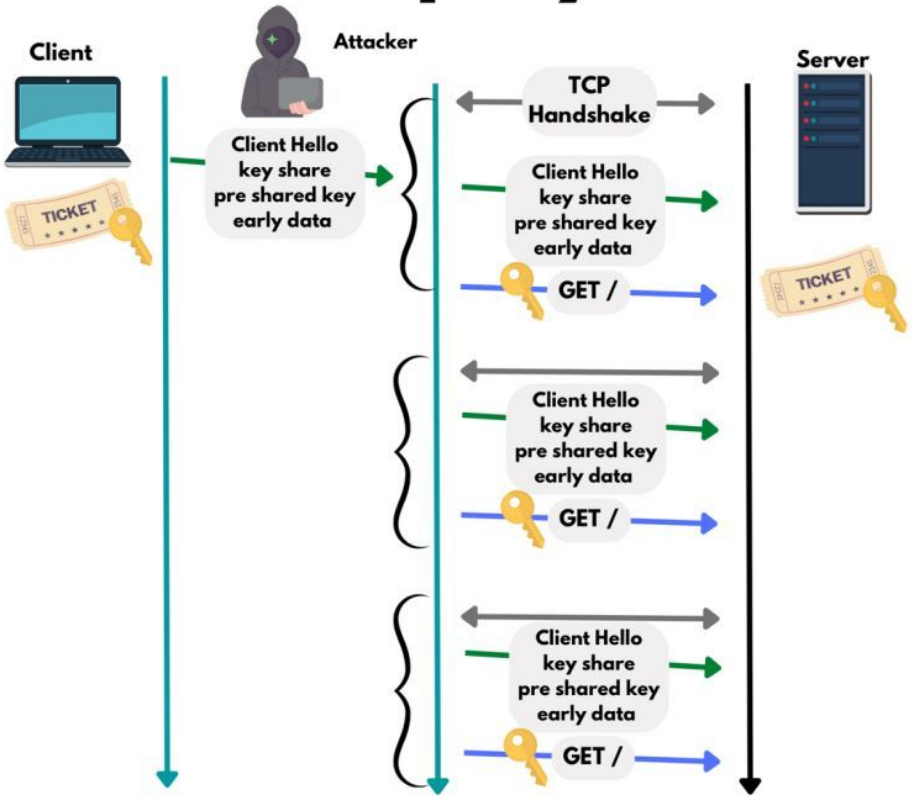
ceptro.br nic.br cgi.br

# 0-RTT

- Funcionalidade para diminuir a latência
  - Se uma conexão já foi realizada com um servidor anteriormente, o processo de handshake não é realizado novamente.
  - Procura na primeira conexão já enviar dados.
  - Pode ser explorado para fazer ataques.



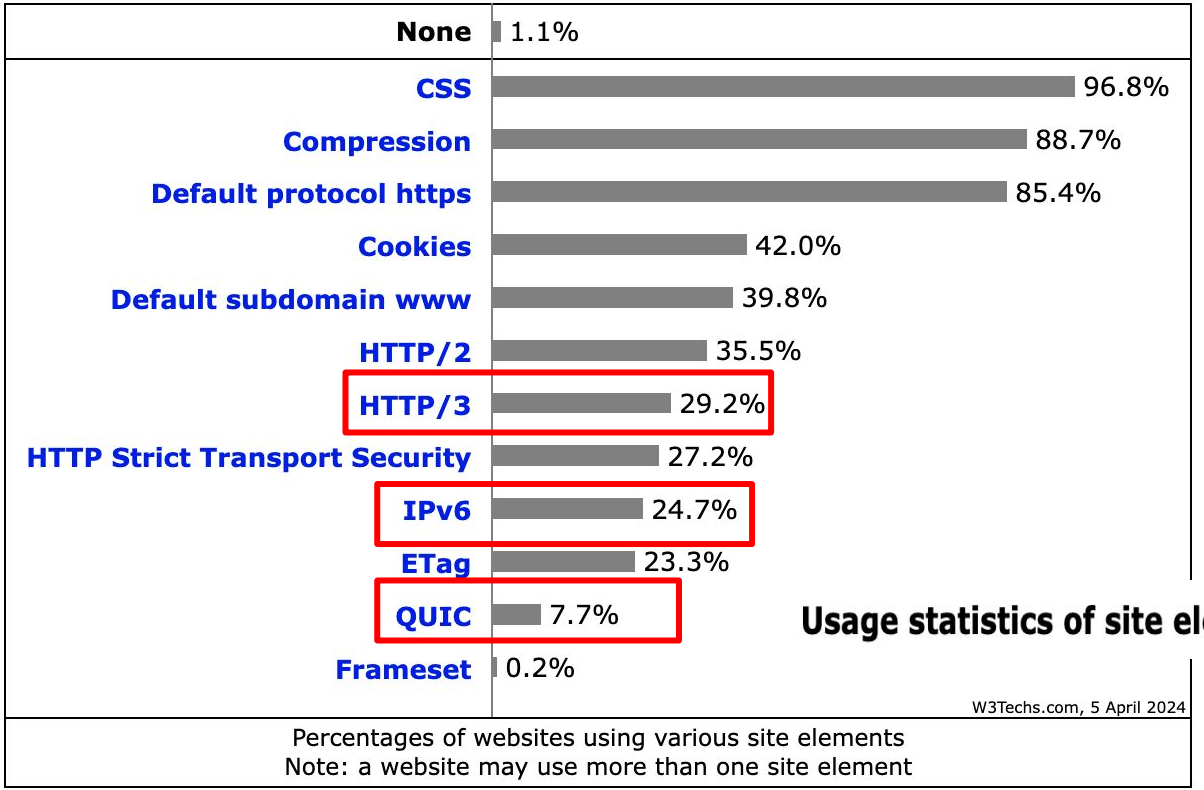
# 0-RTT Replay attack



# Utilização dos Protocolos

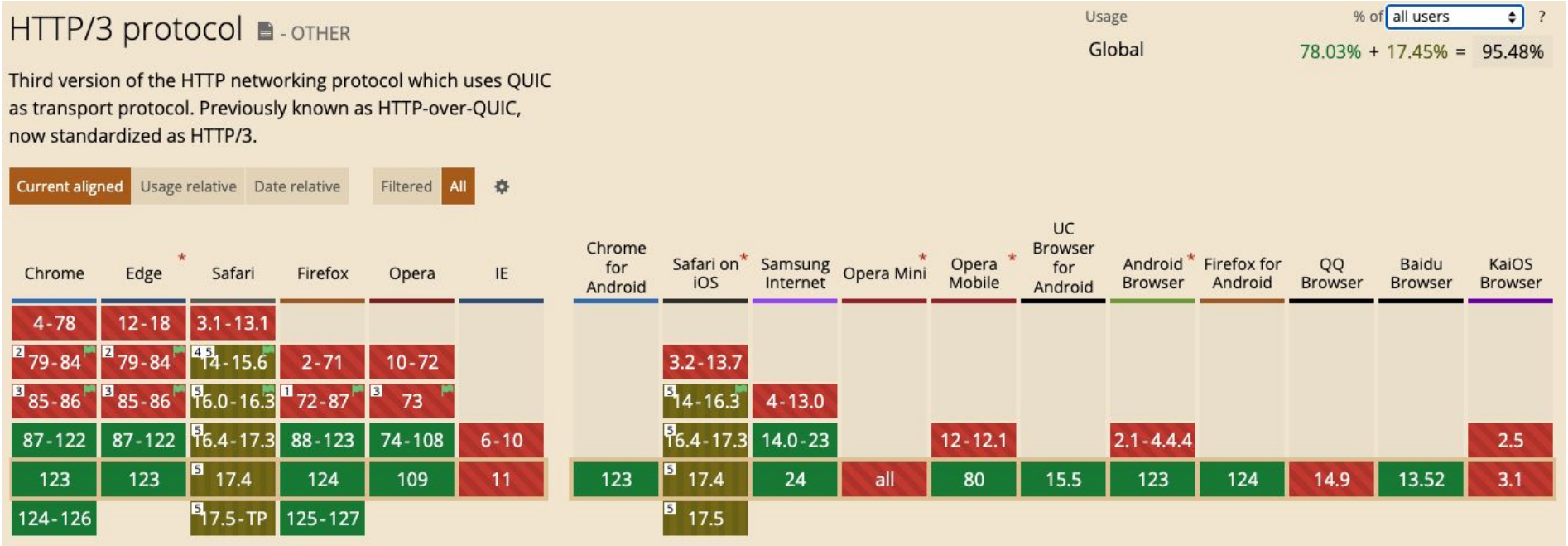
ceptro.br nic.br cgi.br

# Protocolos a nível mundial




https://w3techs.com/technologies/overview/site_element

# Implantação nos Navegadores




https://caniuse.com/http3

# Laboratório

ceptro.br nic.br cgi.br

# Obrigado!

CEPTRO.br Cursos: cursosceptro@nic.br
CEPTRO.br IPv6: ipv6@nic.br

@comunicbr
@nicbr
@NICbrvideos

nic.br cgi.br

www.nic.br | www.cgi.br