Ollscoil na Gaillimhe
University of Galway

CT 420 Real-Time Systems

Emerging Protocols-II

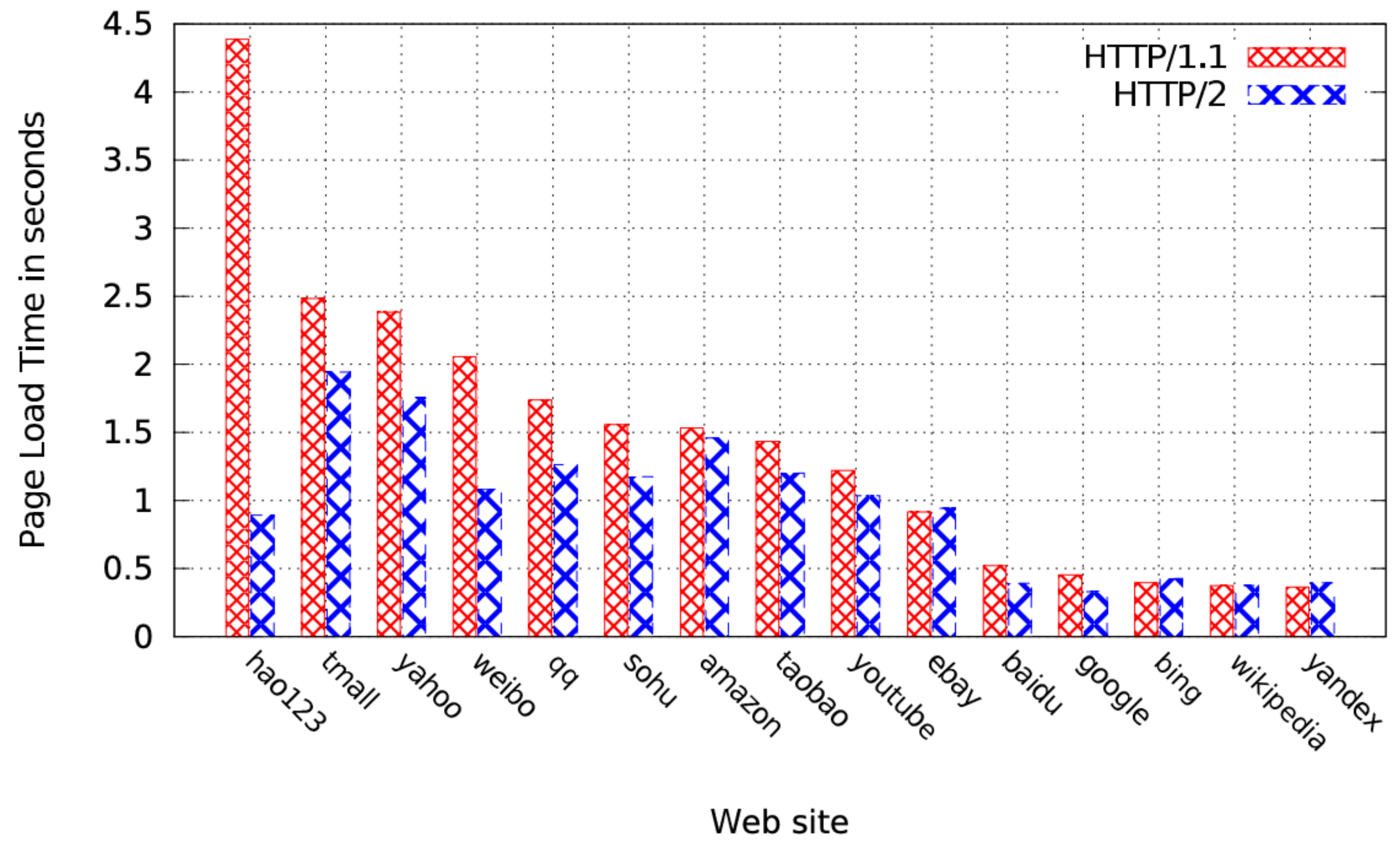
Dr. Jawad Manzoor
Assistant Professor
School of Computer Science

Contents

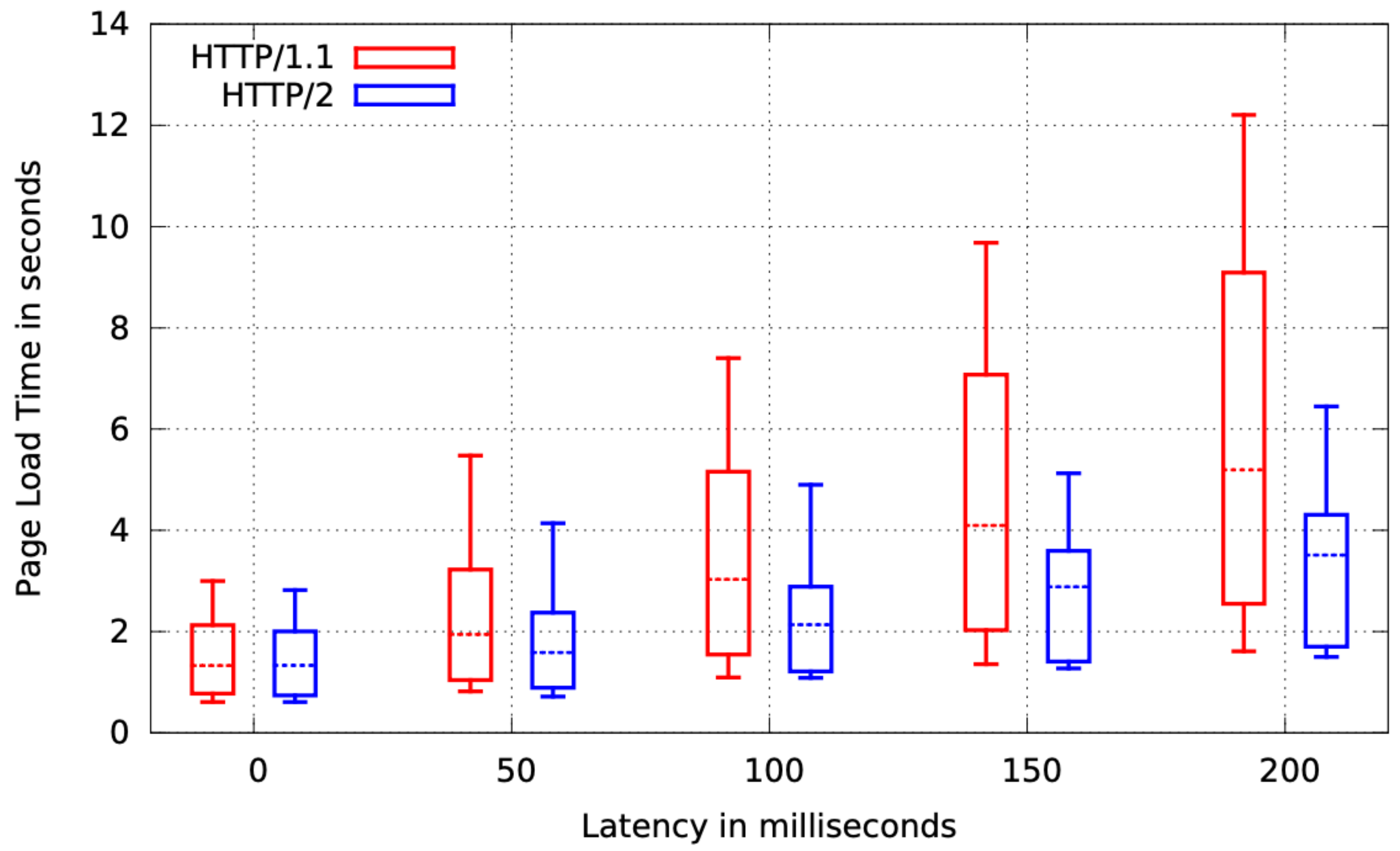
Ollscoil na Gaillimhe
University of Galway

HTTP 2.0 performance

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY



Page load time for different websites using 50ms latency



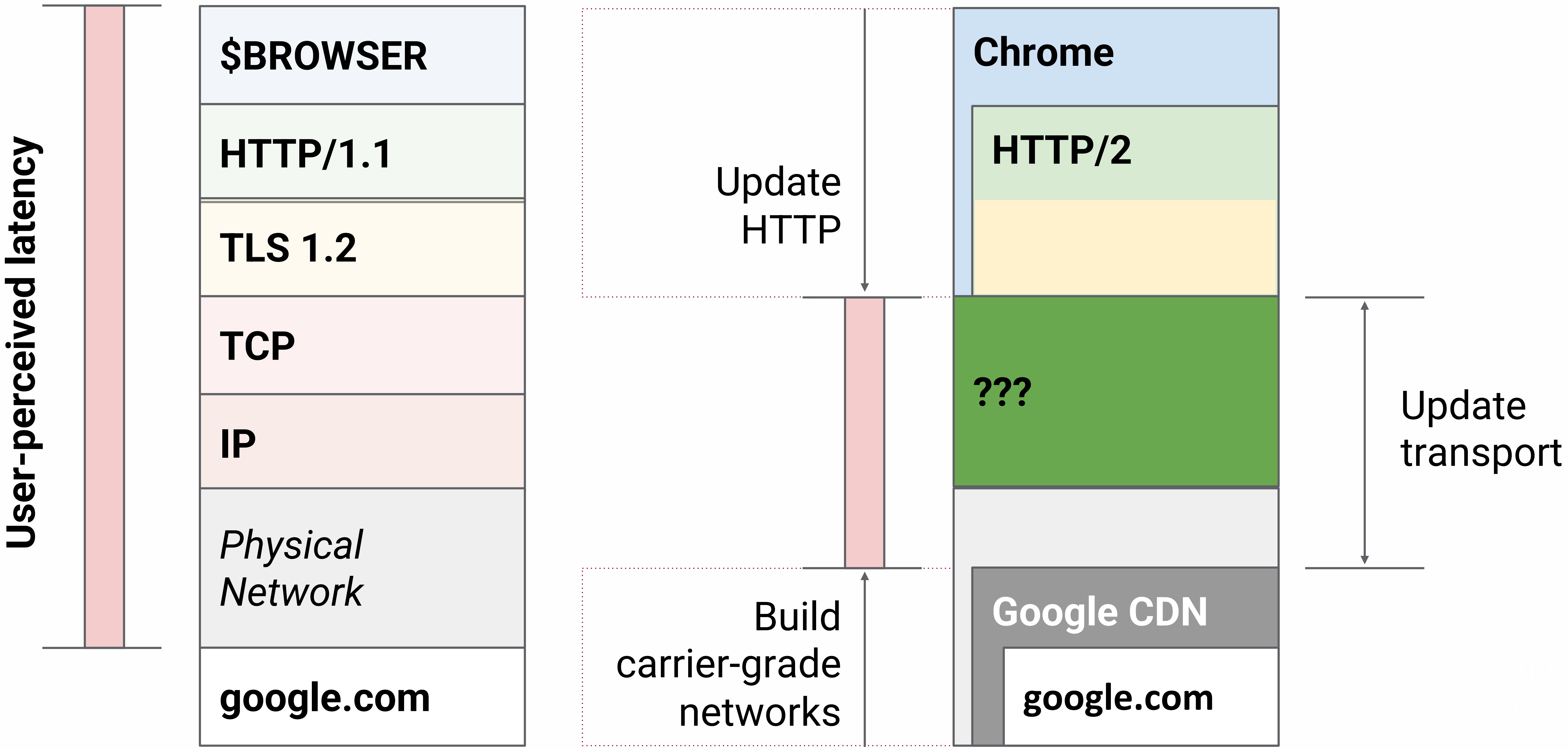
Impact of latency on page load time

"Is HTTP/2 really faster than HTTP/1.1?." *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015.

Can we do better?

- Improvements at the application layer have been implemented in HTTP 2.0
- To further improve the performance, fundamental changes to the underlying transport layer are required

Can we do better?



Transport Protocols

- Transmission Control Protocol (TCP)
  - Reliability and flow control
    - Ensure that data sent is delivered to the receiver application
    - Ensure that receiver buffer doesn’t overflow
  - Ordered delivery
    - Ensure bits pushed by sender arrive at receiver app in order
  - Congestion control
    - Ensure that data sent doesn’t overwhelm network resources

Transport Protocols

- User Datagram Protocol (UDP)
  - Abstraction of independent messages between endpoints
  - UDP has minimal overhead, making it the recommended transport to meet the strict latency bounds of real-time applications, but provides limited support to applications.
  - No guarantee of delivery

TCP vs UDP

Ollscoil na Gaillimhe
University of Galway

TCP

- Slower but more reliable transfers
- Typical Applications:
  - File Transfer Protocol (FTP)
  - Web Browsing
  - Email

unicast

UDP

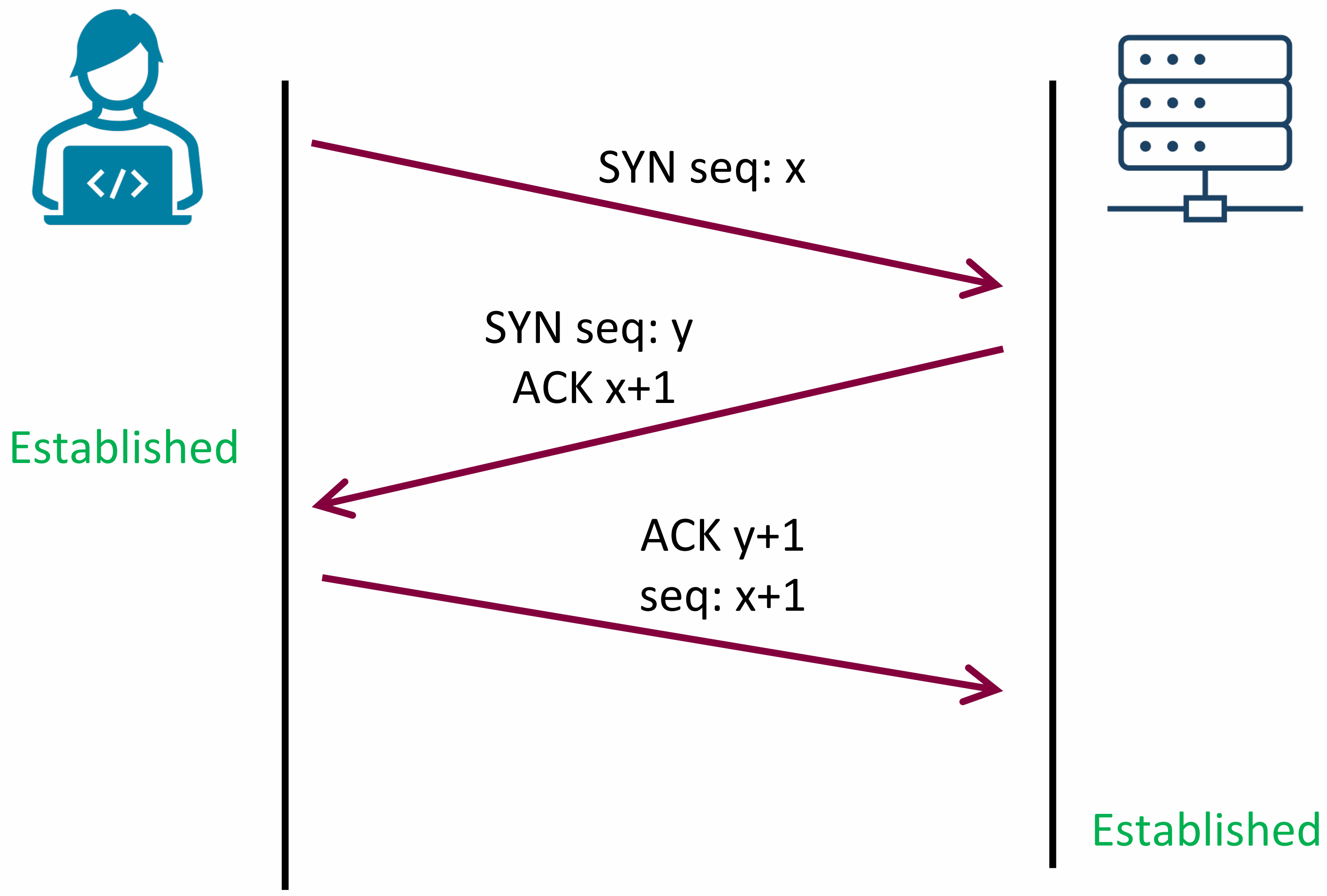
- Faster but not guaranteed transfers ("best effort")
- Typical Applications:
  - Live Streaming
  - Online Games
  - VoIP

unicast **multicast** **broadcast**

https://www.freecodecamp.org/news/tcp-vs-udp/

Things we'd like to change about TCP

- Slow Connection Establishment

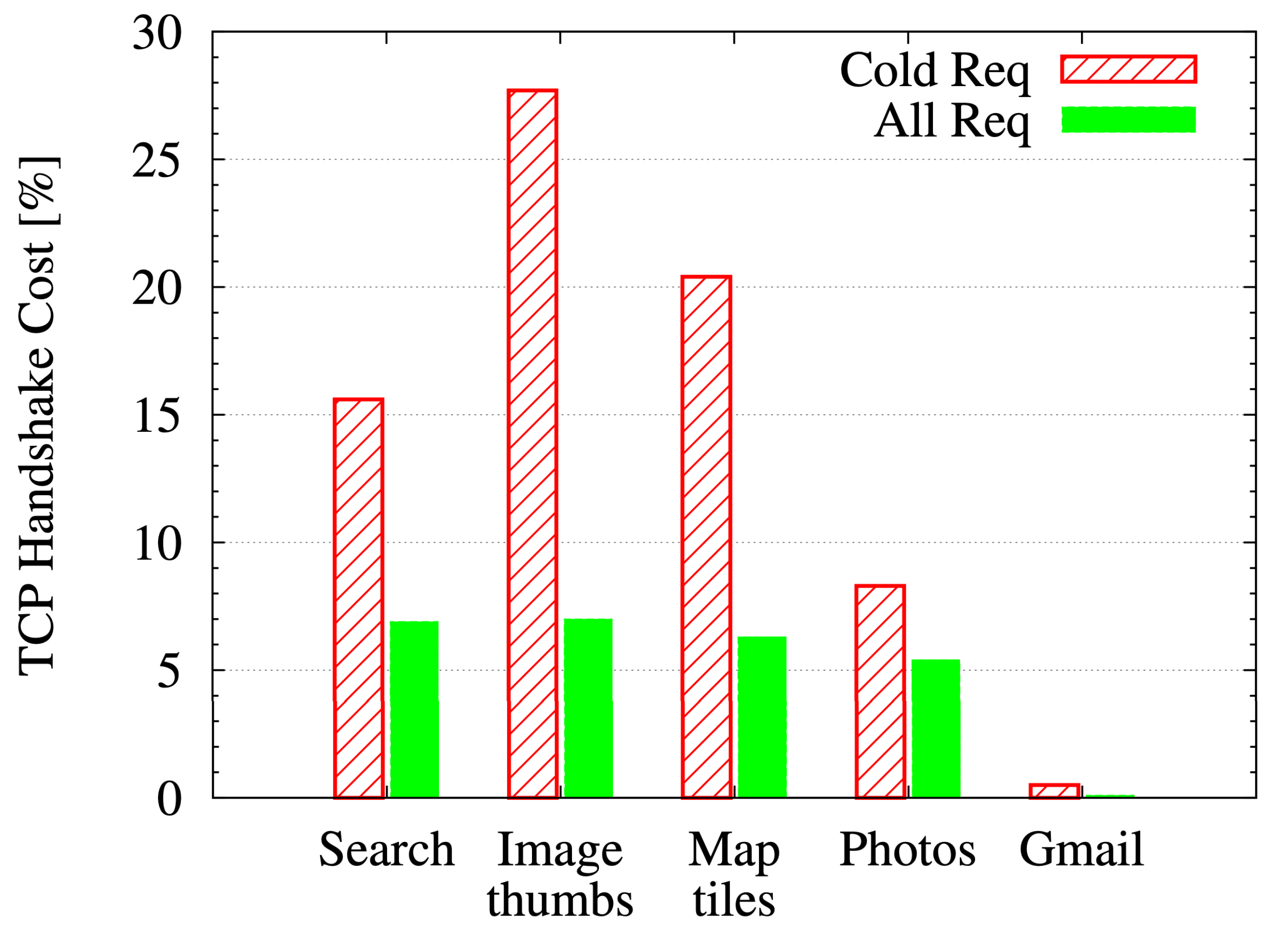


Things we'd like to change about TCP

Ollscoil na Gaillimhe
University of Galway

- TCP 3-way handshake has very high latency, particularly for small web requests
- Problem made worse by adding security (HTTPS / TLS) over TCP.
  - TLS handshake adds more round trips.
- Experiment: study the impact of TCP's handshake on user perceived HTTP request latency.
  - Sampled a few billion HTTP requests (on port 80) to Google servers world-wide to multiple Google services such as search, email, and photos
  - For each sampled request, we measured the latency
  - Requests sent on new TCP connections are defined as cold requests and those that reuse TCP connections as warm requests

https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37517.pdf

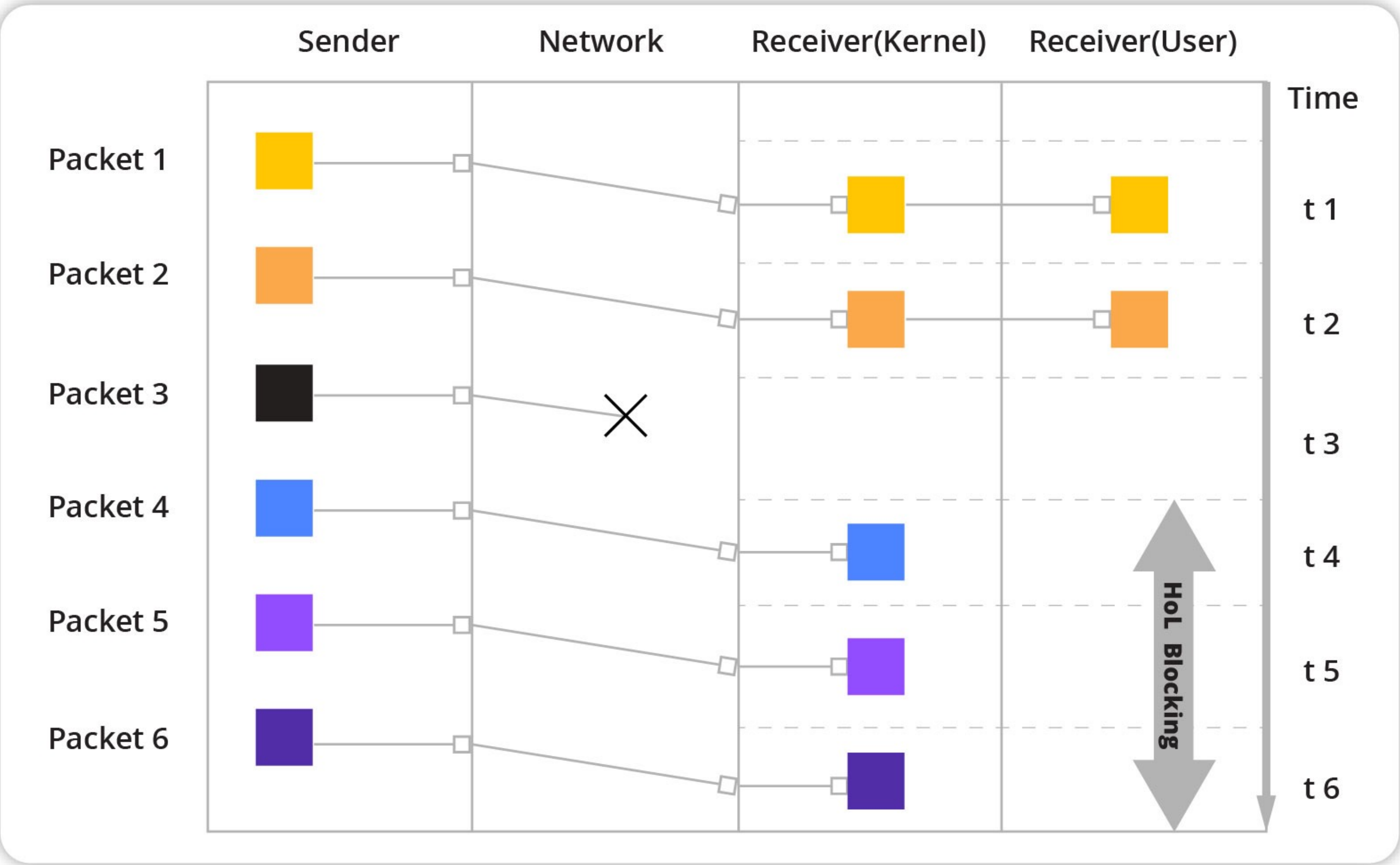


"TCP Fast Open" CoNext 2011

Things we'd like to change about TCP

- Head-of-line blocking

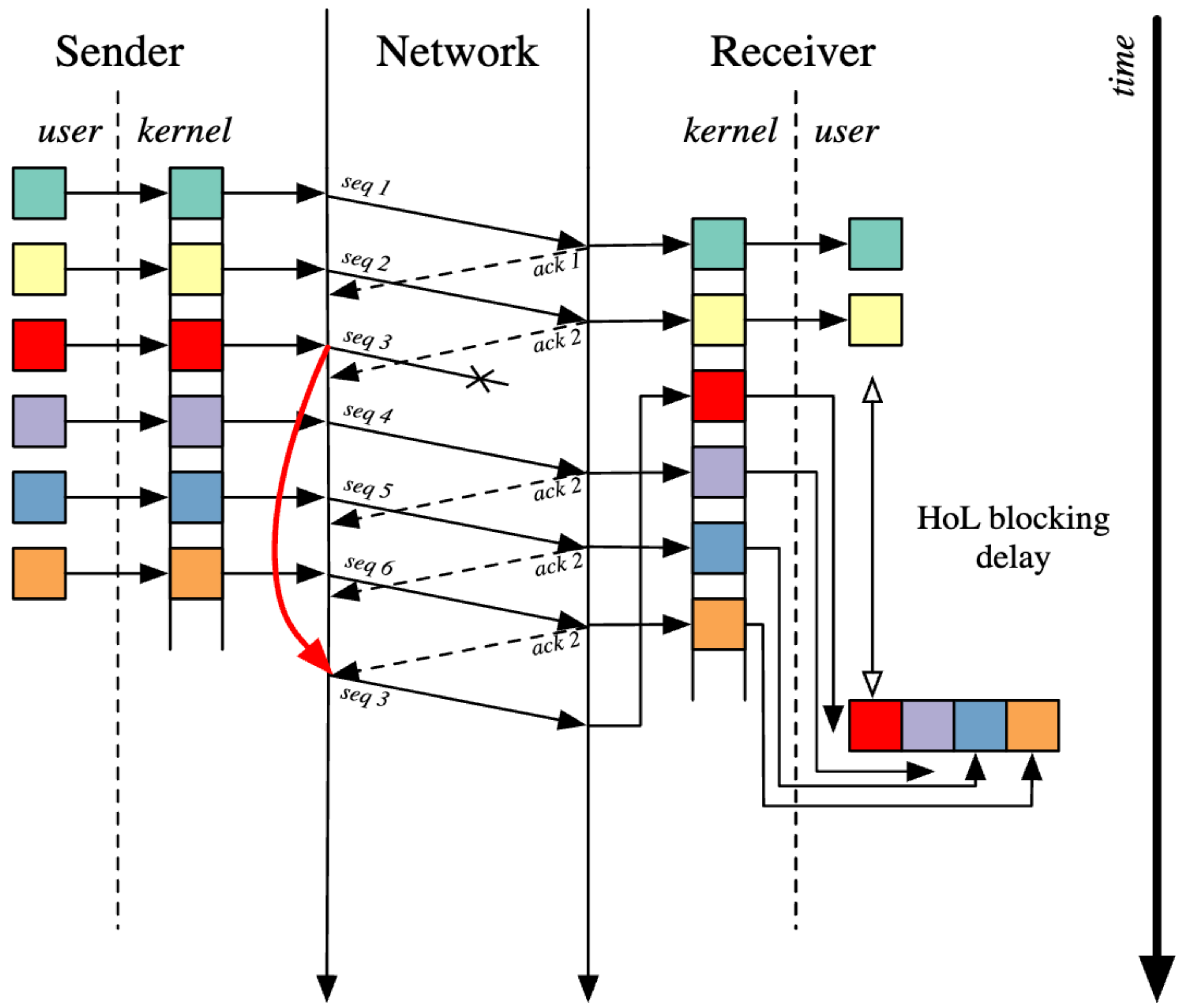
TCP Head of Line Blocking



https://www.cdnetworks.com/media-delivery-blog/what-is-quic/

Things we'd like to change about TCP

- Head-of-line blocking



"TCP Hollywood: An unordered, time-lined, TCP for networked multimedia applications." *IFIP networking conference and workshops*.

Why is it so hard to change TCP?

Ollscoil na Gaillimhe
University of Galway

- TCP is implemented in Operating System (OS) kernel
- You may need to change the OS kernel
  - On all servers and clients around the world!

- You may need to change the entire network!
  - Middleboxes may drop packets if they don’t understand something on the packet

QUIC

- QUIC protocol was developed to overcome the performance issues in TCP.
- It is a reliable transport over UDP
- Initially designed by Jim Roskind at Google, implemented, and deployed in 2012.
- In May 2021, the IETF standardized QUIC in RFC 9000
- In June 2022, IETF standardized HTTP/3 as RFC 9114 which uses QUIC by default

Browser support

Ollscoil na Gaillimhe
University of Galway

HTTP/3 protocol - OTHER

Usage | % of all users | ?
Global | 94.47% + 1.24% = 95.7%

Baseline 2024 Newly available across major browsers

Third version of the HTTP networking protocol which uses QUIC as transport protocol. Previously known as HTTP-over-QUIC, now standardized as HTTP/3.

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
		3.1 - 13.1					3.2 - 13.7									
4 - 78	12 - 18	[4 5] 14 - 15.6					[5] 14 - 15.8									
[2] 79 - 84	[2] 79 - 84	[5] 16.0 - 16.3	2 - 71	10 - 72			[5] 16.0 - 16.3									
[3] 85 - 86	[3] 85 - 86	[5] 16.4 - 17.6	[1] 72 - 87	[3] 73			[5] 16.4 - 17.7	4 - 13.0								
87 - 132	87 - 132	18.0 - 18.2	88 - 134	74 - 113	6 - 10		18.0 - 18.2	14.0 - 26		12 - 12.1		2.1 - 4.4.4				2.5
133	133	18.3	135	114	11	133	18.3	27	all	80	15.5	133	135	14.9	13.52	3.1
134 - 136		18.4 - TP	136 - 138				18.4									

https://caniuse.com/http3

Client support

- Google apps
  - Several mobile apps from Google support QUIC including YouTube, Gmail, Google Drive, Google Search etc.
- Social Media apps
  - Facebook, Instagram, WhatsApp, Snapchat
- Enterprise & Cloud Services
  - Microsoft 365, Uber
- Streaming
  - Netflix, Disney+, Amazon Prime, Spotify

Server support

- LiteSpeed, NGINX, Apache HTTP Server, Caddy
- Microsoft Windows Server 2022
- CDSs: Akamai Technologies, Cloudflare, CDNetworks
- As of early 2025, around 27-30% of all websites support HTTP/3

- QUIC implementations:

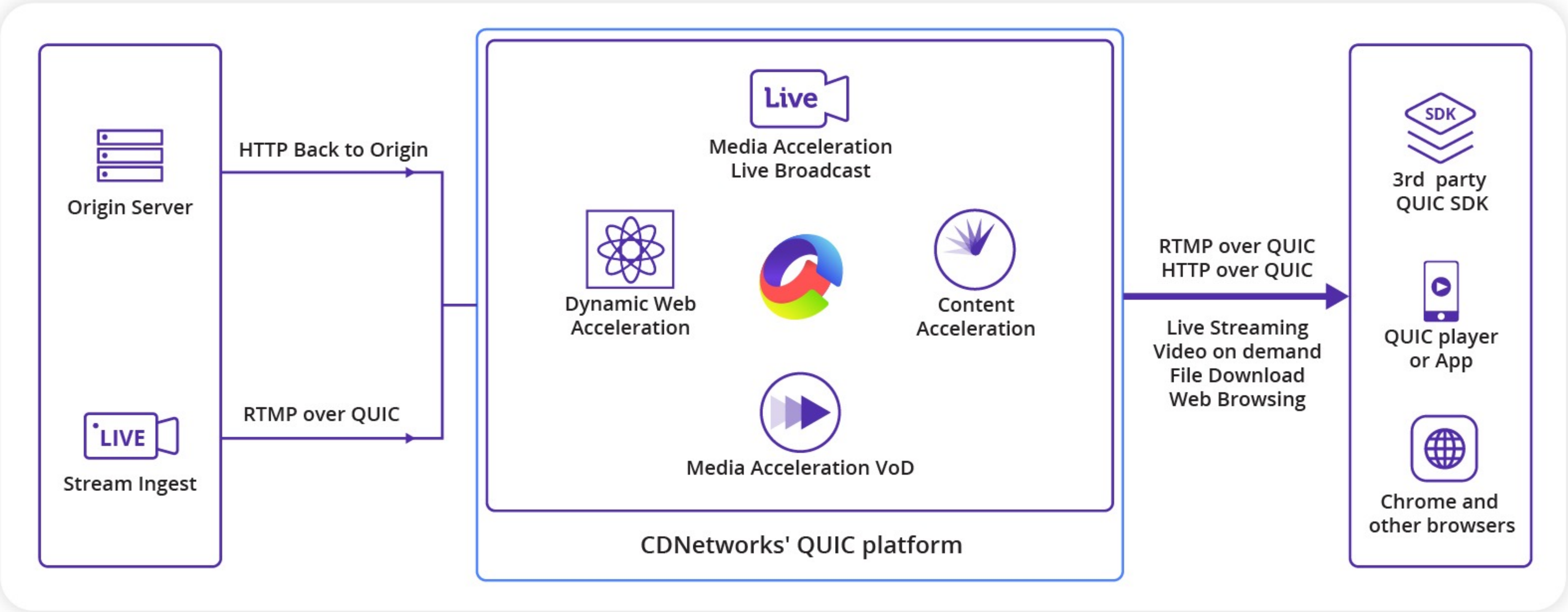
https://github.com/quicwg/base-drafts/wiki/Implementations

QUIC Performance

Ollscoil na Gaillimhe
University of Galway

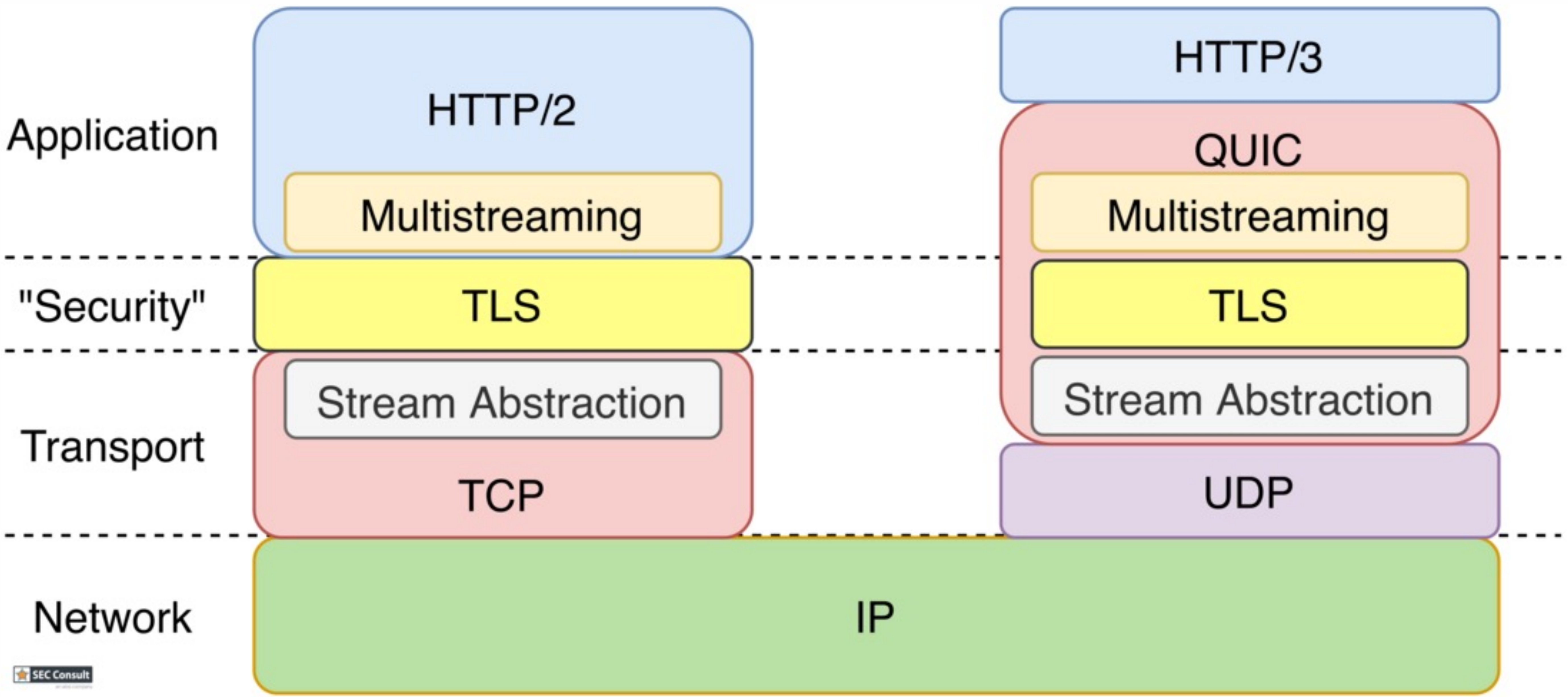
The internal testing data at CDNetworks showed that, during QUIC stream pull scenario with a 1Mbps bitrate, the new platform can improve bandwidth performance by 41% under the same business concurrency conditions

CDNetworks QUIC Overview



https://www.cdnetworks.com/media-delivery-blog/what-is-quic/

Where does QUIC fit?



https://sec-consult.com/

HTTP 3.0

Ollscoil na Gaillimhe
University of Galway

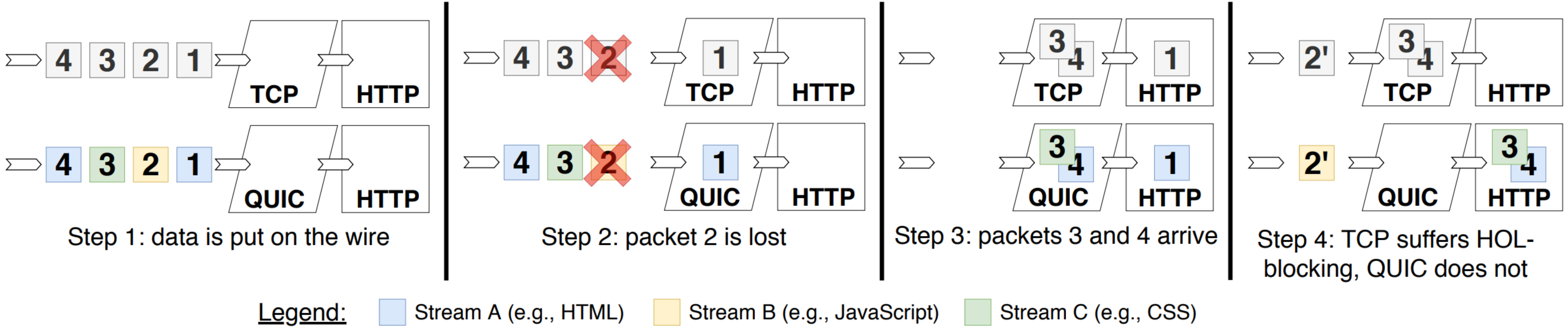
- In June 2022, IETF standardized HTTP/3 as RFC 9114
- The main difference between HTTP 2.0 and HTTP 3.0 is the underlying transport layer protocol.
  - In HTTP 2.0, we have TCP connections
  - HTTP 3.0 uses QUIC (UDP-based protocol)

QUIC Features

- Multiplexing without head-of-line blocking
- Low latency connections
- Connection Migration
- Linkability Prevention
- Encryption
- Resistance to protocol ossification
- Field compression with QPACK

No head-of-line blocking

- In QUIC each byte stream is transported independently over the network.
- QUIC ensures the in-order delivery of packets within the same byte stream.
- If a packet gets lost, only the affected stream gets blocked and waits for its re-transmission.

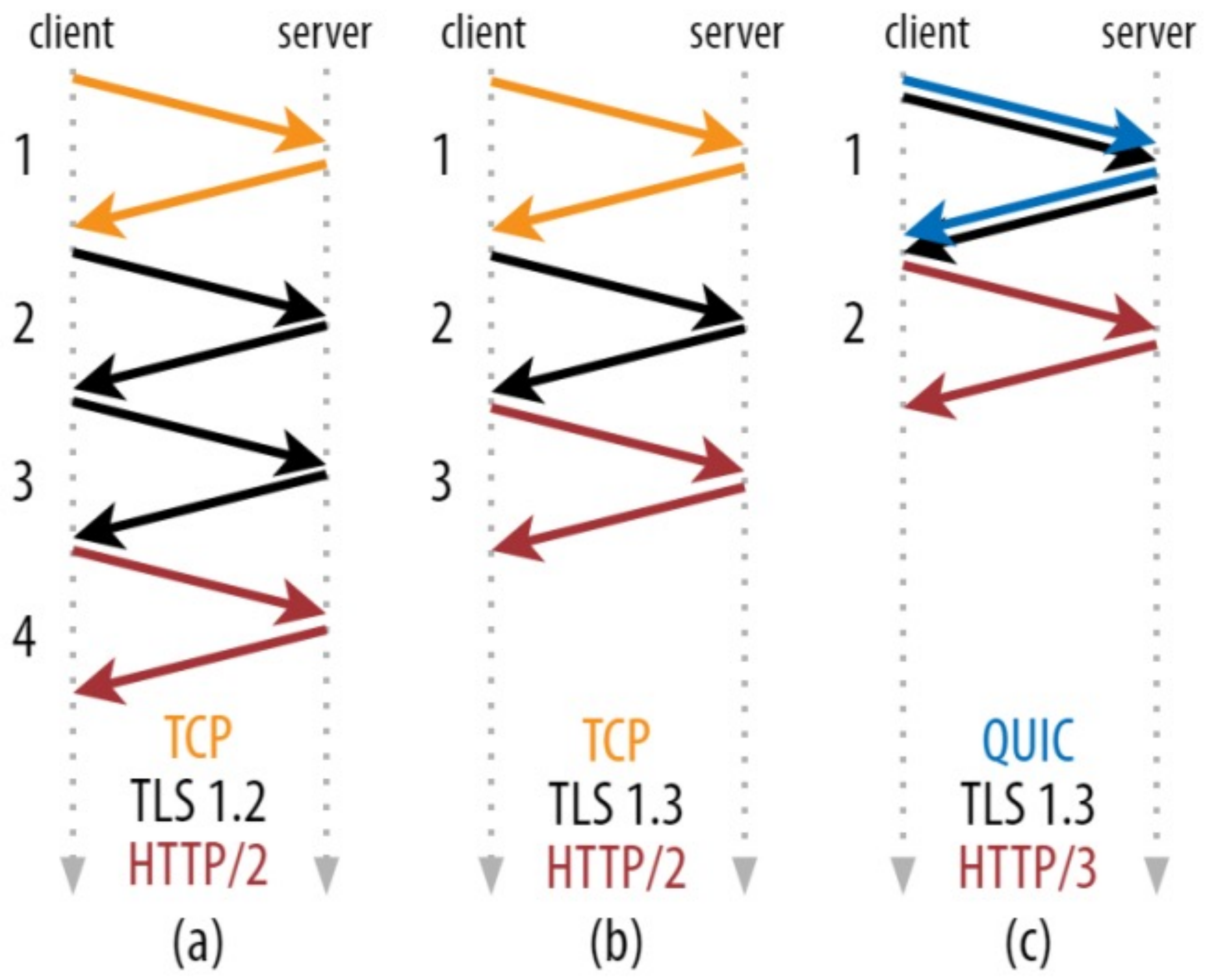


Source: Robin Marx et al., "Resource Multiplexing and Prioritization in HTTP/2 over TCP Versus HTTP/3 over QUIC"

Connection Establishment

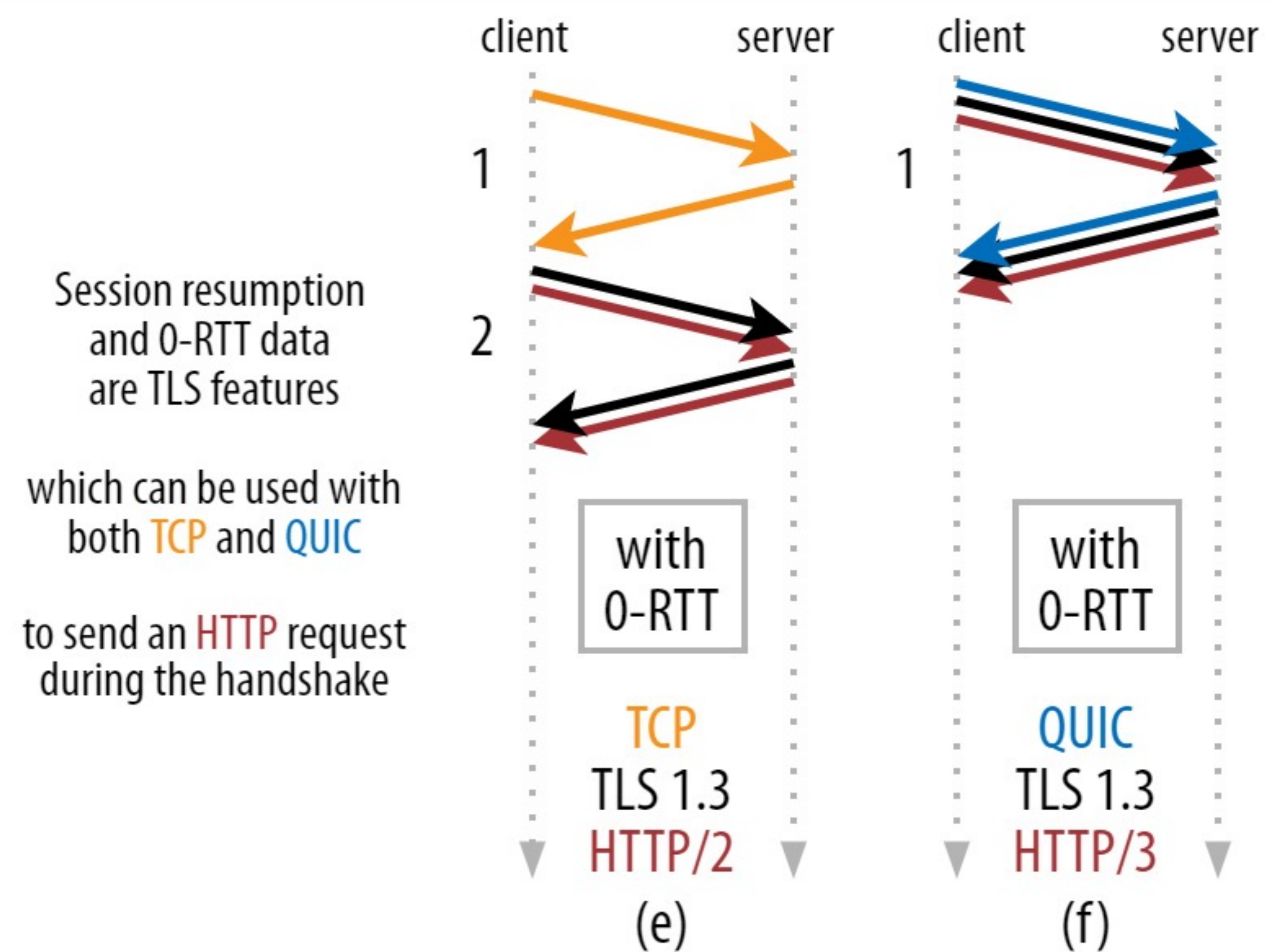
Ollscoil na Gaillimhe
University of Galway

- QUIC has a faster connection setup, as it combines the 'transport' handshake with the TLS cryptographic session establishment, which in TCP+TLS are two separate processes.



0-RTT Connection

- To reduce the time required to establish a new connection, a client that has previously connected to a server may cache certain parameters from that connection and subsequently set up a 0-RTT connection with the server.
- With 0-RTT feature, an HTTP request can be sent, and a (partial) response can be received during the very first handshake!



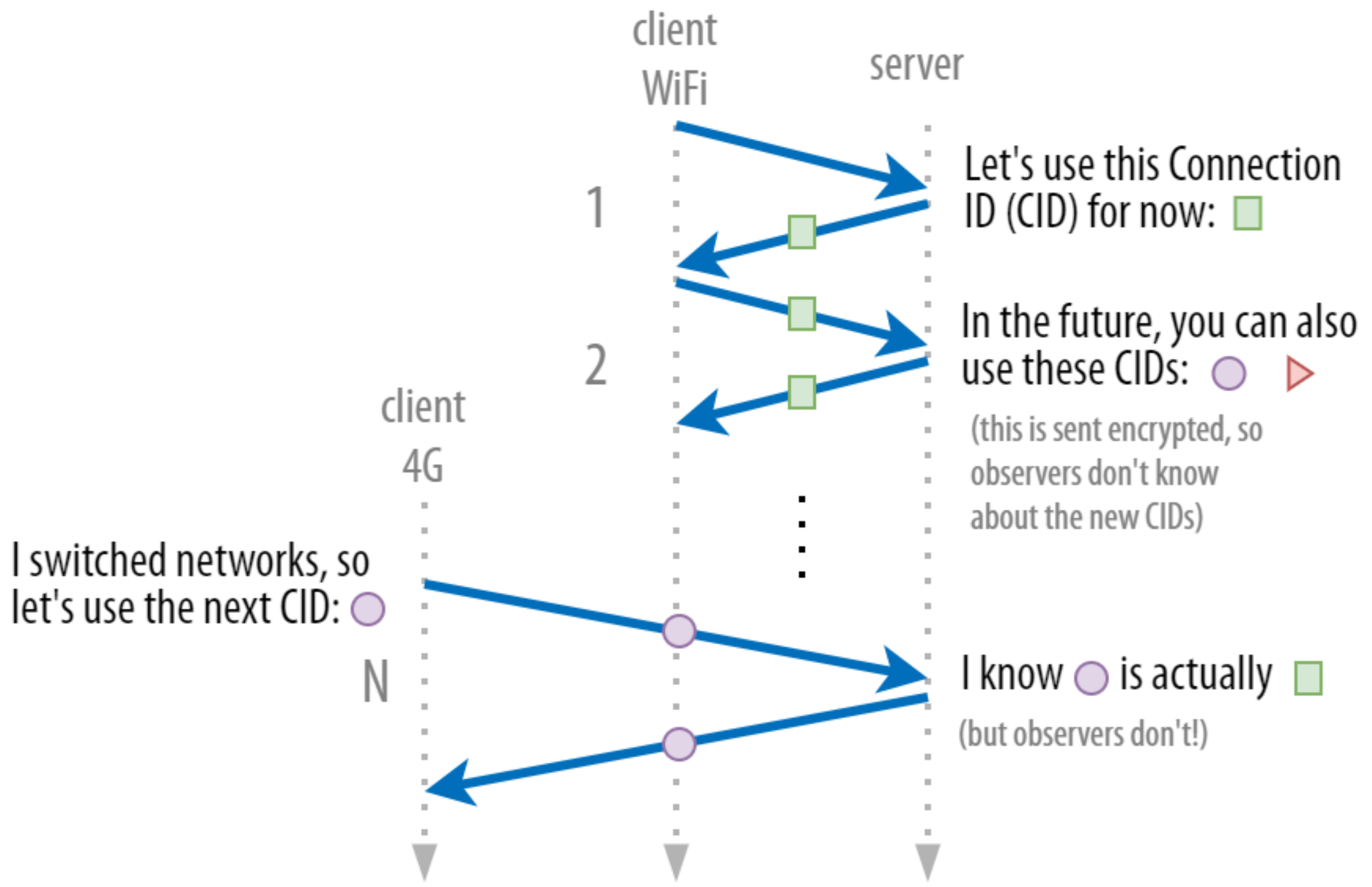
Connection Migration

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

- QUIC improves performance during network-switching events, like what happens when a user of a mobile device moves from a local WiFi hotspot to a mobile network.
- When this occurs on TCP, a lengthy process starts where every existing connection times out one-by-one and is then re-established on demand.
- QUIC includes a connection identifier to uniquely identify the connection to the server regardless of source.
- This allows the connection to be re-established simply by sending a packet, which always contains this ID, as the original connection ID will still be valid even if the user's IP address changes.

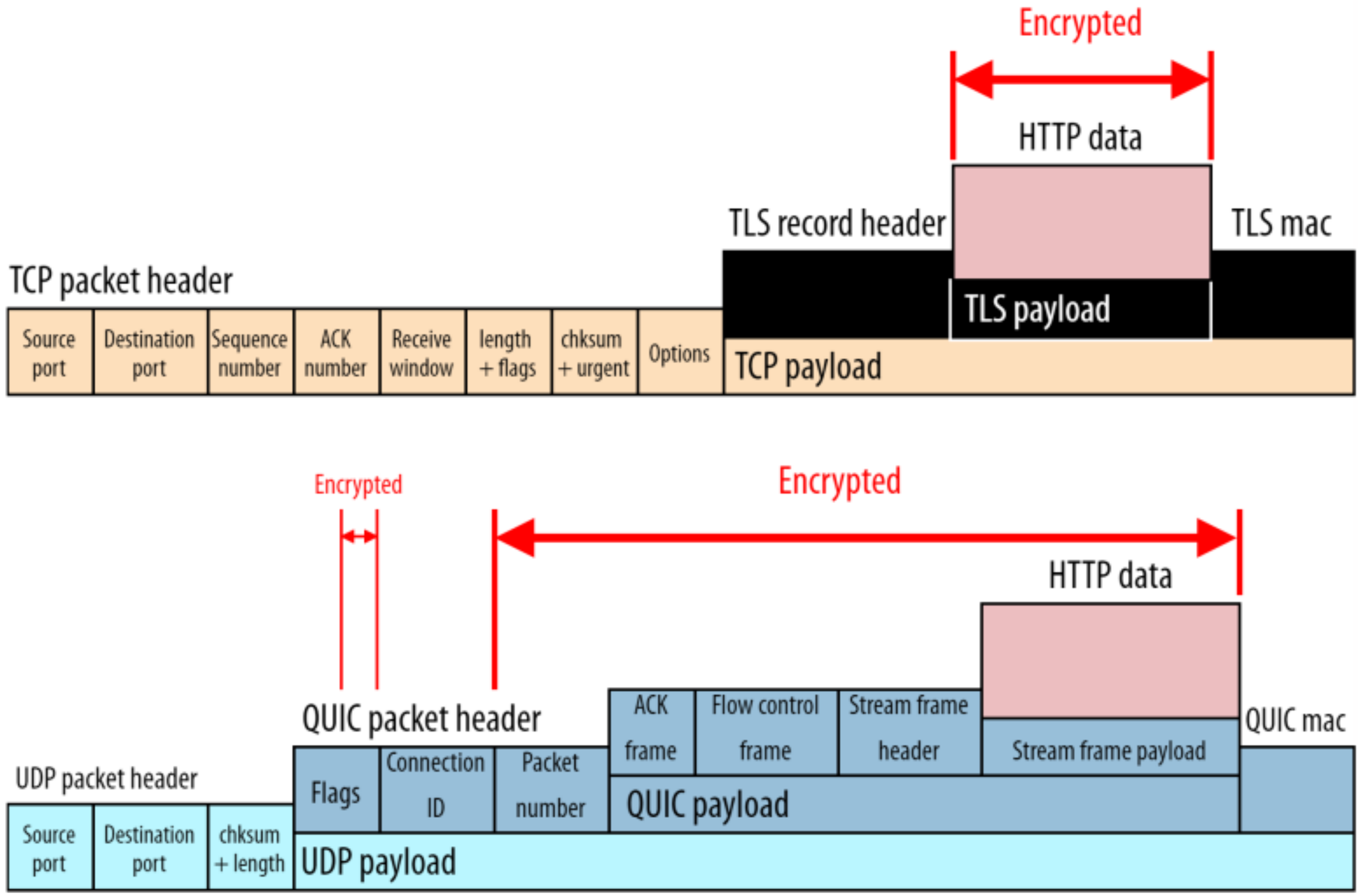
Linkability prevention

- To avoid privacy issues, e.g. to prevent hackers from following the physical movement of a smartphone user by tracking the unencrypted CID across networks, QUIC uses a list of connection identifiers instead of just one.



Encryption

- TCP + TLS only encrypts the actual HTTP data.
- QUIC also encrypts large parts of the protocol header.
  - Metadata, such as packet numbers and connection-close signals, which were visible to all middleboxes (and attackers) in TCP, are now only available to the client and server in QUIC.



Source: https://pulse.internetsociety.org/

Resistance to protocol ossification

- Protocol ossification, is an inherent characteristic of protocols implemented in the operating system (OS) kernel, such as TCP.
  - OSs are rarely updated, which applies even more to the operating systems of middleboxes, such as firewalls and load balancers.
  - It is a problem because it makes it hard to introduce new features, as middleboxes with an older version of the protocol don't recognize the new feature and drop the packets.
- QUIC aims to solve this issue.
  - QUIC runs in the user space instead of the kernel, so it's easier to deploy new implementations.
  - QUIC is heavily encrypted. If middlebox can't read a piece of info, it can't make any decisions based on that info.

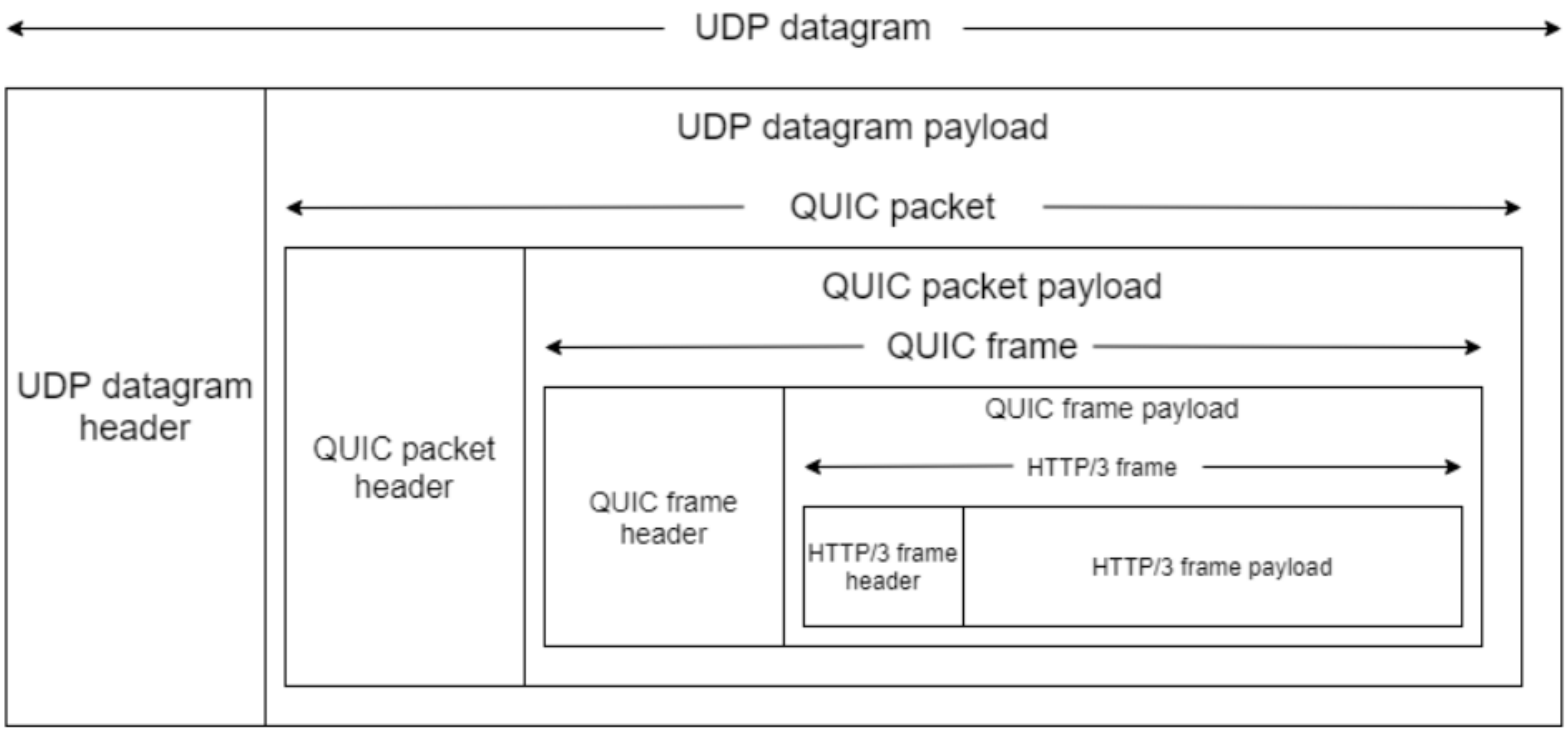
QPACK Field Compression

- QPACK is a field compression format for HTTP/3.
- Field compression eliminates redundant metadata by assigning indexes to fields that are used multiple times during the connection.
- The goal of QPACK is to reduce the space taken up by HTTP headers.
  - Smaller size translates to higher throughput and better user experience.
  - Effectiveness of data compression is expressed as compression ratio.
- For example, if you compare a string “LiteSpeed” to “ls”, the compression ratio is 2/9, or about 0.22. The smaller the number, the better the compression performance.

QUIC Packet Structure

Ollscoil na Gaillimhe
University of Galway

- A QUIC packet is composed of a common header followed by one or more frames.



QUIC packets

- QUIC endpoints communicate by exchanging packets.
- QUIC packet consists of header and payload.
- QUIC has two different types of headers.
  - The long header is used prior to the connection establishment.
  - The short header is used after the first connection established.
- Packets are carried in UDP datagram.

QUIC packets

- QUIC provides different packet types:
  - Initial packet - It transports the first CRYPTO frames transmitted by the client and server during the key exchange and the ACK frames in both directions.
  - Handshake packet - It is used for sending and receiving encrypted handshake messages and acknowledgments between the server and the client.
  - 0-RTT packet – It sends "early" data from the client to the server before the handshake is completed.
  - 1-RTT packet – It is used to exchange data between client and server after the handshake is completed.

QUIC packets

- QUIC initial packet example

```
QUIC IETF
    QUIC Connection information
    [Packet Length: 1350]
    1... .... = Header Form: Long Header (1)
    .1.. .... = Fixed Bit: True
    ..00 .... = Packet Type: Initial (0)
    .... 00.. = Reserved: 0
    .... ..00 = Packet Number Length: 1 bytes (0)
    Version: draft-29 (0xff00001d)
    Destination Connection ID Length: 8
    Destination Connection ID: 45fb5955dfaa8914
    Source Connection ID Length: 0
    Token Length: 0
    Length: 1332
    Packet Number: 1
    Payload:
5a99e5b29413627619ca3b5add4cf8b6ce348355b1c1a2be9874c7961e7996a24aeec
860…
    TLSv1.3 Record Layer: Handshake Protocol: Client Hello
    PADDING Length: 997
```

QUIC frames

Ollscoil na Gaillimhe
University of Galway

- A QUIC packet is composed of a common header followed by one or more frames.
- There are various types of frames:
  - ACK frame - Receivers send ACK frames to inform senders of packets they have received and processed. The ACK frame contains one or more ACK Ranges.
  - CRYPTO frame - It is used to transmit cryptographic handshake messages.
  - STREAM frame – It implicitly create a stream and carry stream data. It contains a Stream ID, Offset, Length and Stream Data.
  - MAX_DATA frame - Used in flow control to inform the peer of the maximum amount of data that can be sent on the connection as a whole.
  - MAX_STREAM_DATA frame - Used in flow control to inform a peer of the maximum amount of data that can be sent on a stream.
  - MAX_STREAMS frame - Informs the peer of the cumulative number of streams of a given type it is permitted to open.

QUIC frames

- QUIC CRYPTO frame examples

```
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
    Frame Type: CRYPTO (0x0000000000000006)
    Offset: 0
    Length: 314
    Crypto Data
    Handshake Protocol: Client Hello
```

```
TLSv1.3 Record Layer: Handshake Protocol: Server Hello
  Frame Type: CRYPTO (0x0000000000000006)
  Offset: 0
  Length: 90
  Crypto Data
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 86
    Version: TLS 1.2 (0x0303)
    Random:
0f58bdbd934450c7aa98242121447bef2fe0733aa5fc3beffab6513c7177f9a4
    Session ID Length: 0
    Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
    Compression Method: null (0)
    Extensions Length: 46
    Extension: key_share (len=36)
    Extension: supported_versions (len=2)
```

QUIC streams

- Streams in QUIC provide a lightweight, ordered byte-stream abstraction to an application.
- Streams can be created by either endpoint, can concurrently send data interleaved with other streams.
- Streams are identified within a connection by a numeric value, referred to as the stream ID (a 62-bit integer) that is unique for all streams on a connection.
- Client-initiated streams have even-numbered stream IDs and server-initiated streams have odd-numbered stream IDs

Ollscoil na Gaillimhe
University of Galway

Thank you for your attention!