Ollscoil na Gaillimhe
University of Galway

CT 420
Real-Time Systems

Emerging Protocols-II

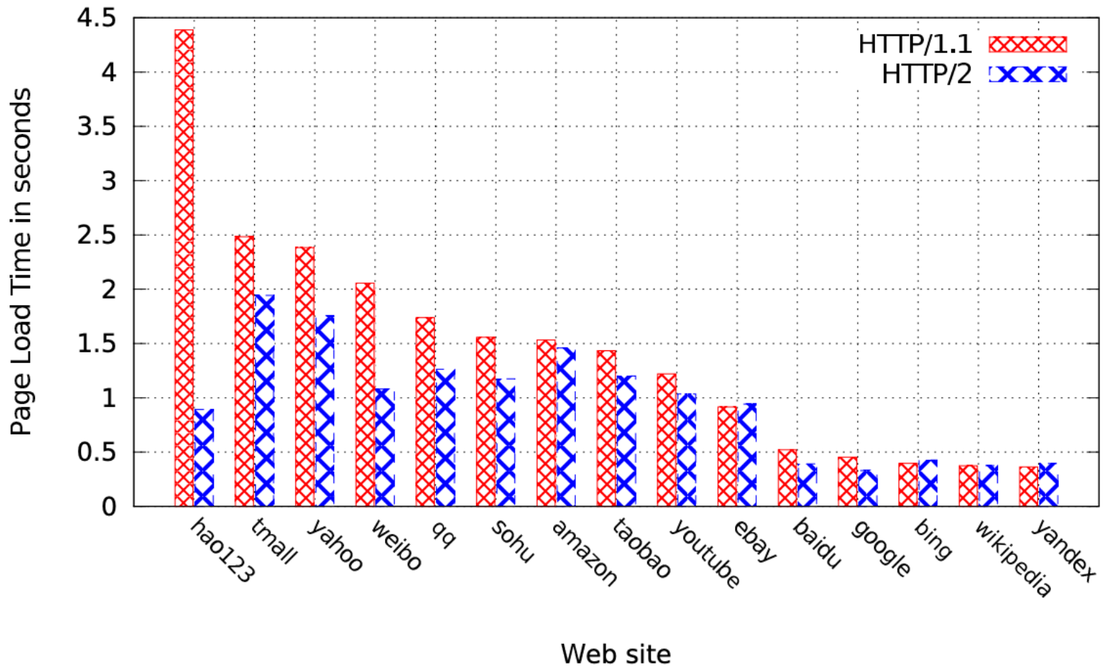
Dr. Jawad Manzoor
Assistant Professor
School of Computer Science

Contents

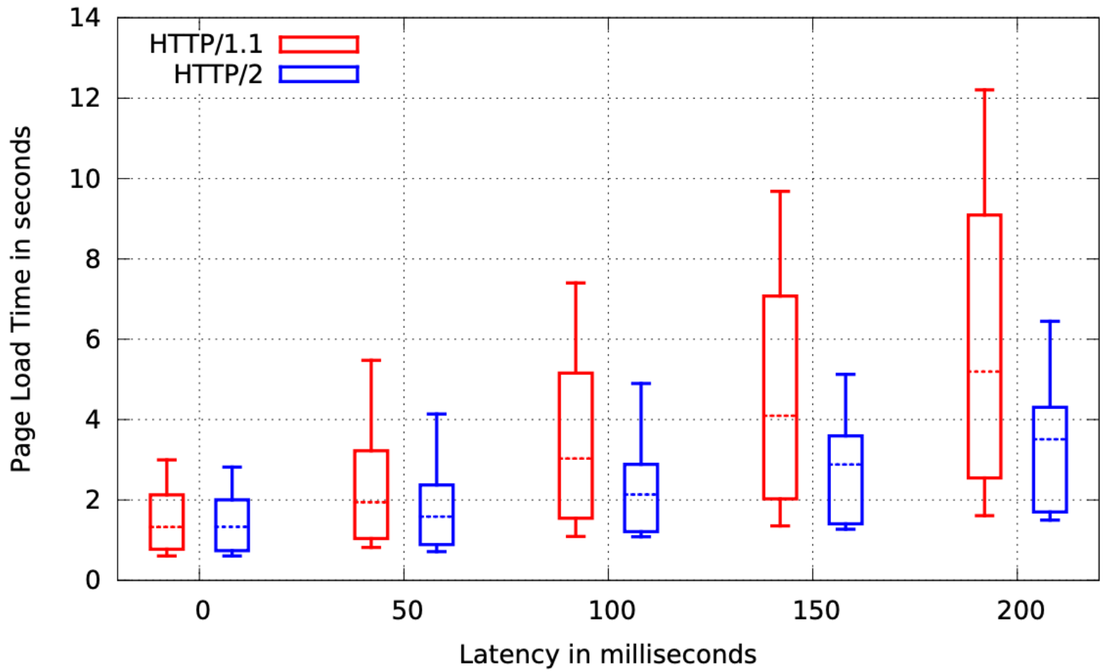
Ollscoil na Gaillimhe
University of Galway

- HTTP/3
- TCP Performance Issues
- QUIC protocol
- QUIC traffic analysis in wireshark

HTTP 2.0 performance



Page load time for different websites using 50ms latency



Impact of latency on page load time

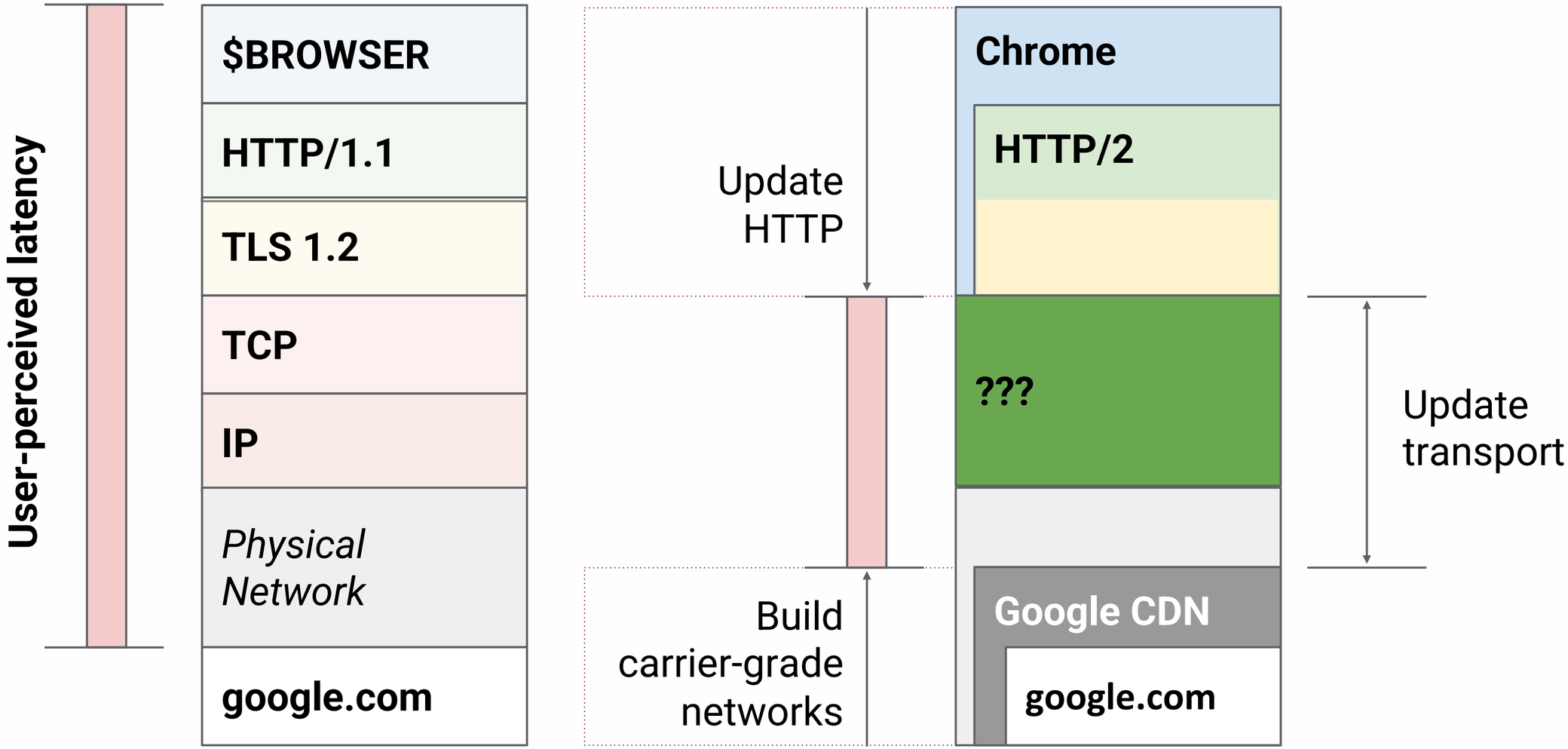
"Is HTTP/2 really faster than HTTP/1.1?." *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015.

Can we do better?

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

TCP

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

unicast

UDP

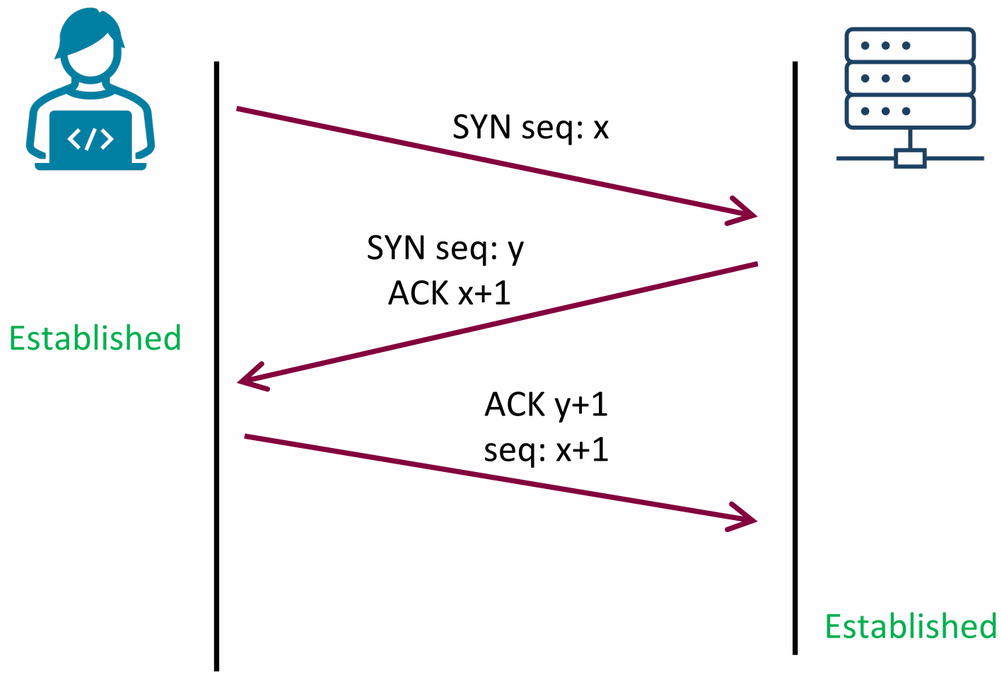
- Faster but not guaranteed transfers ("best effort")
- Typical Applications:
  - Live Streaming
  - Online Games
  - VoIP

unicast multicast broadcast

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

Things we'd like to change about TCP

- Slow Connection Establishment

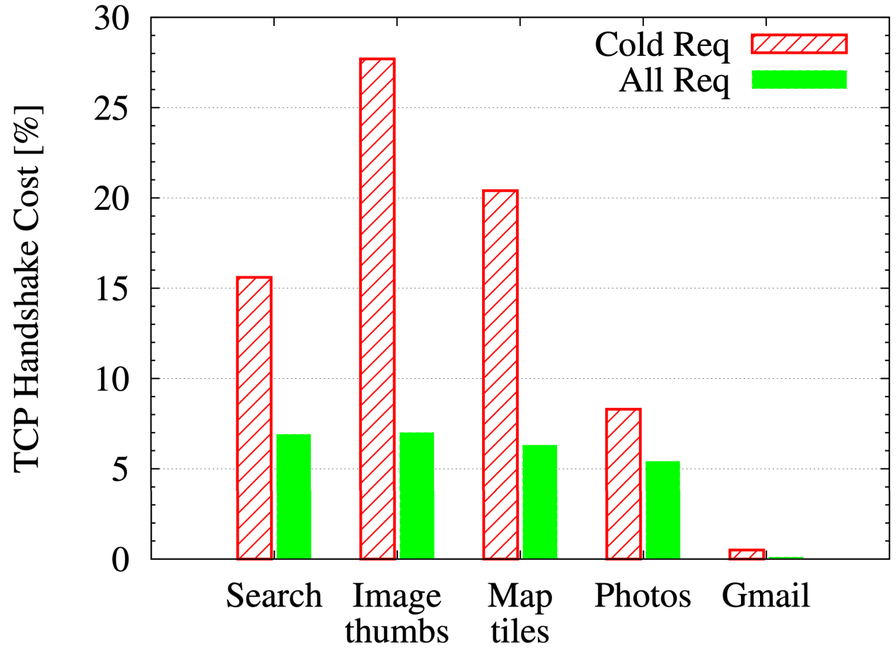


Things we'd like to change about TCP

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

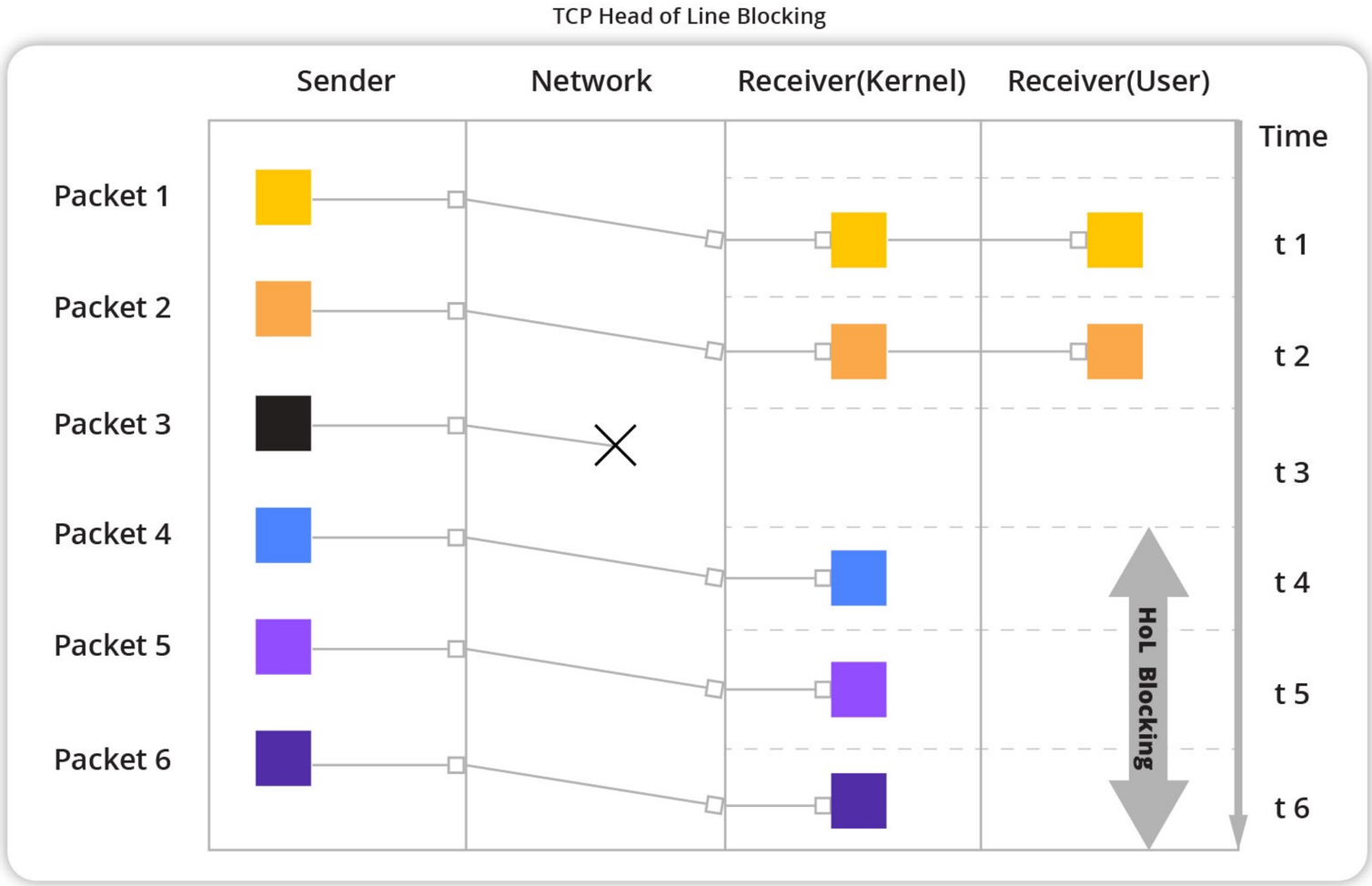
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37517.pdf



"TCP Fast Open" CoNext 2011

Things we'd like to change about TCP

- Head-of-line blocking

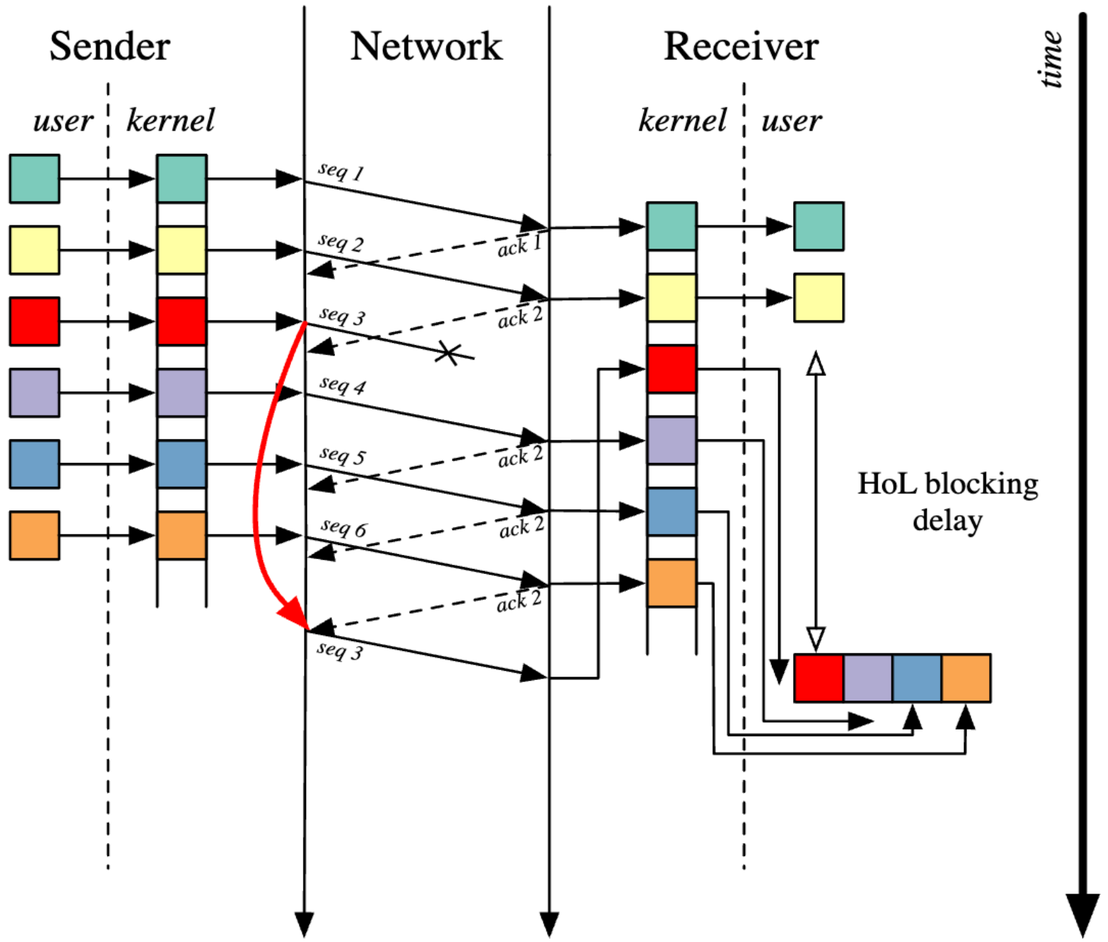


https://www.cdnetworks.com/media-delivery-blog/what-is-quic/

Things we'd like to change about TCP

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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	14-15.6					14-15.8									
79-84	79-84	16.0-16.3	2-71	10-72			16.0-16.3									
85-86	85-86	16.4-17.6	72-87	73			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

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

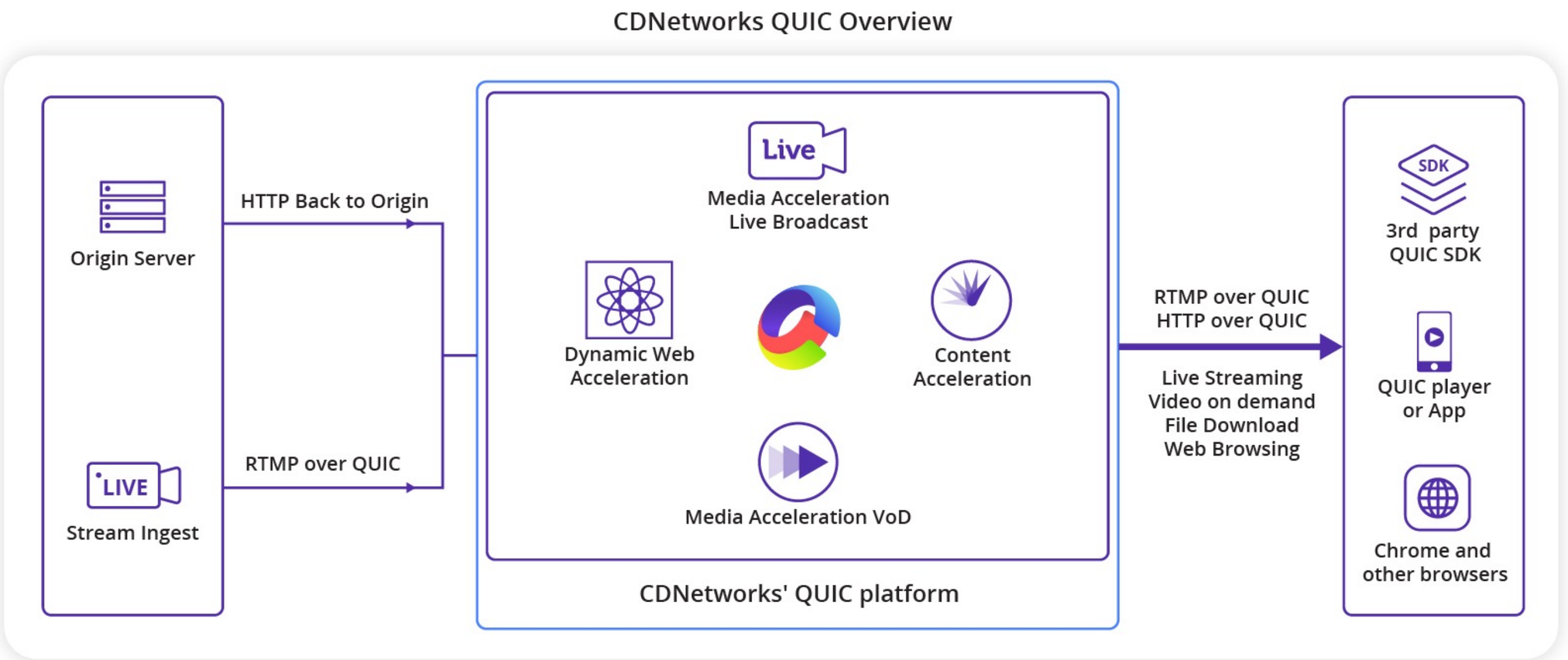
- QUIC implementations:

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

QUIC Performance

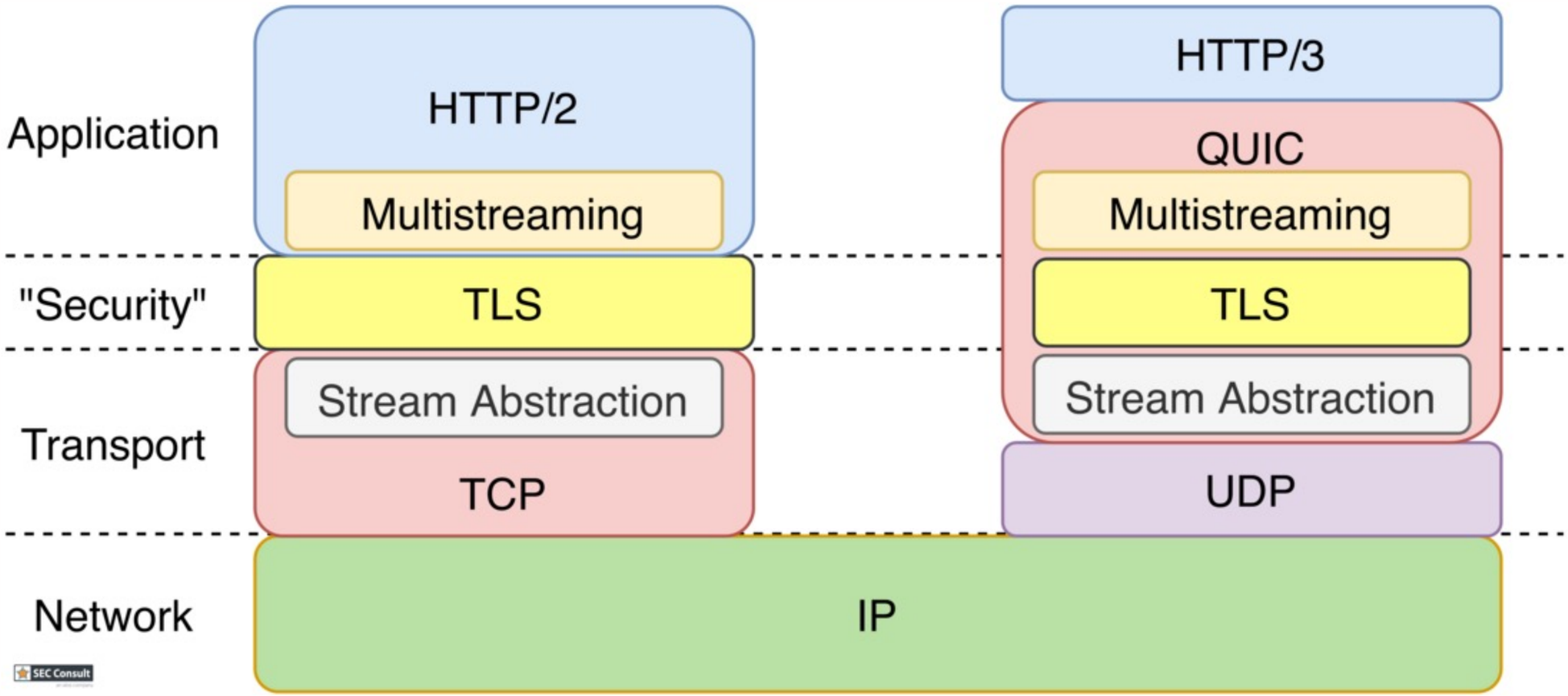
Ollscoil na Gaillimhe
University of Galway

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



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

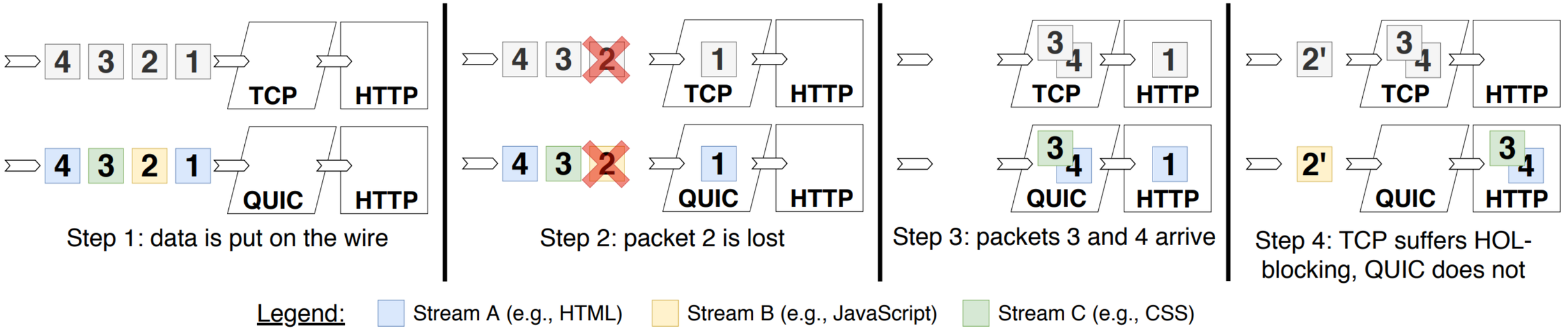
Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

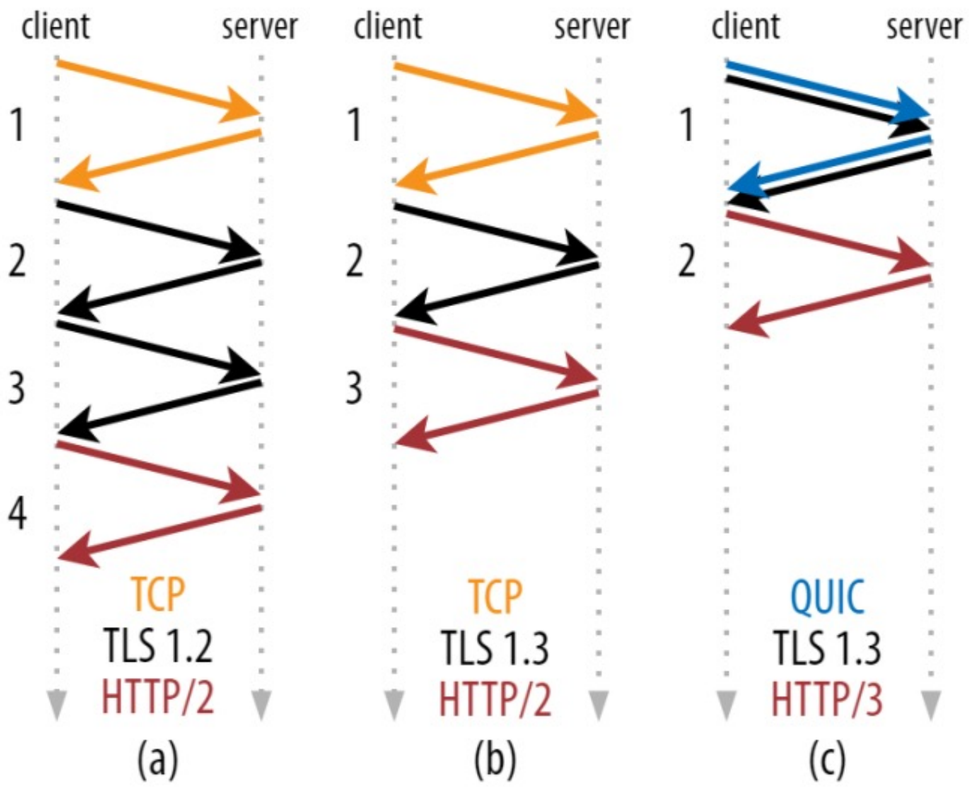
- In QUIC each byte stream is transported independently over the network.
- QUIC ensures the in-order delivery of packets within the same byte stream.
- If a packet gets lost, only the affected stream gets blocked and waits for its re-transmission.



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

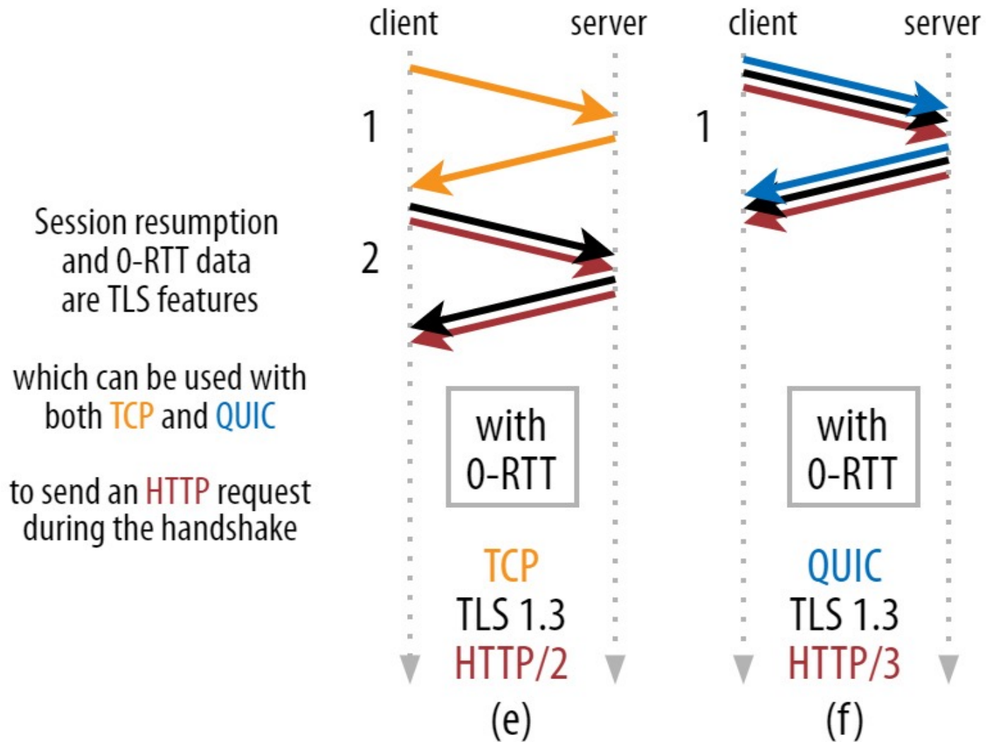
Connection Establishment

- 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!



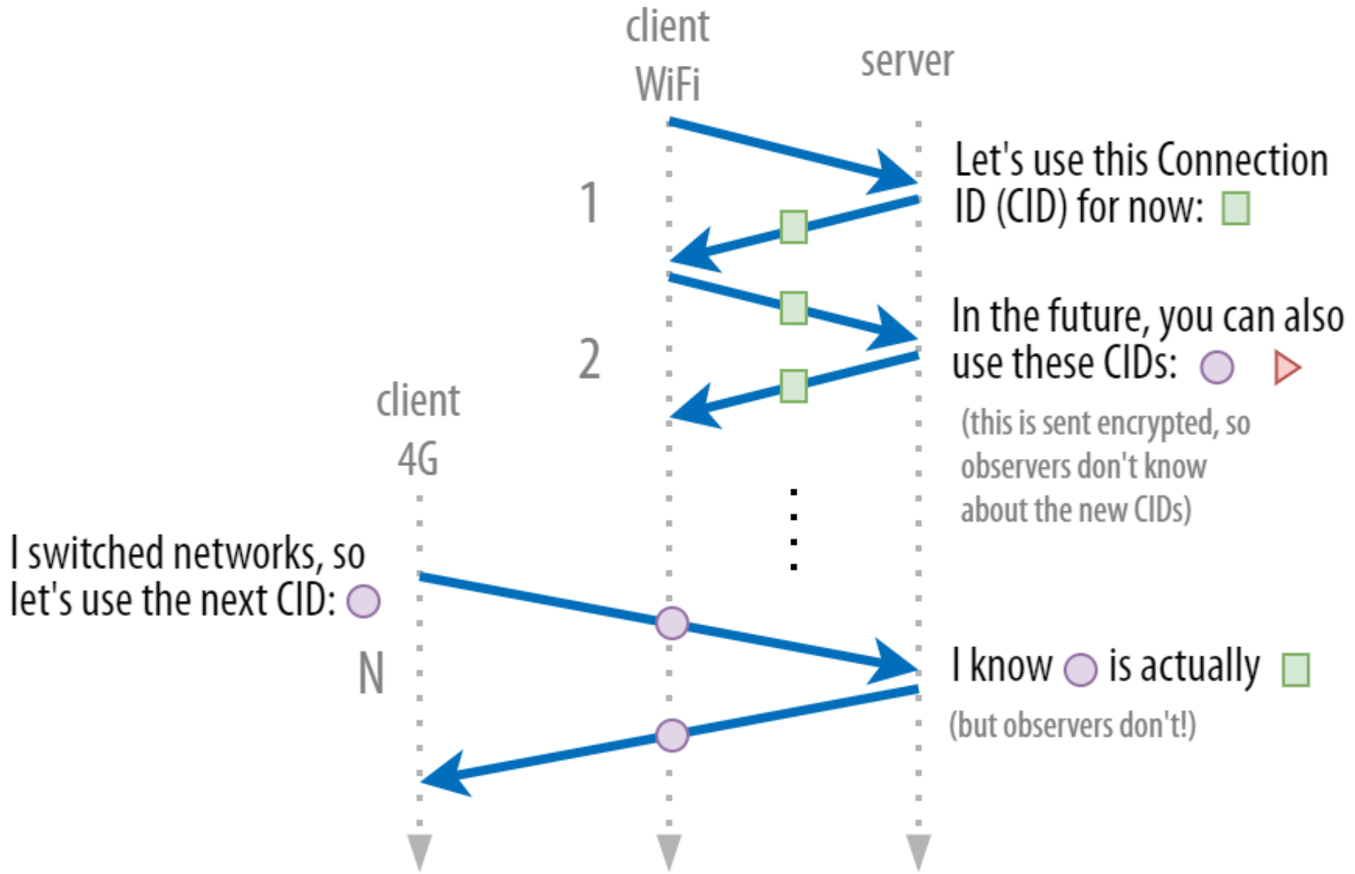
Connection Migration

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

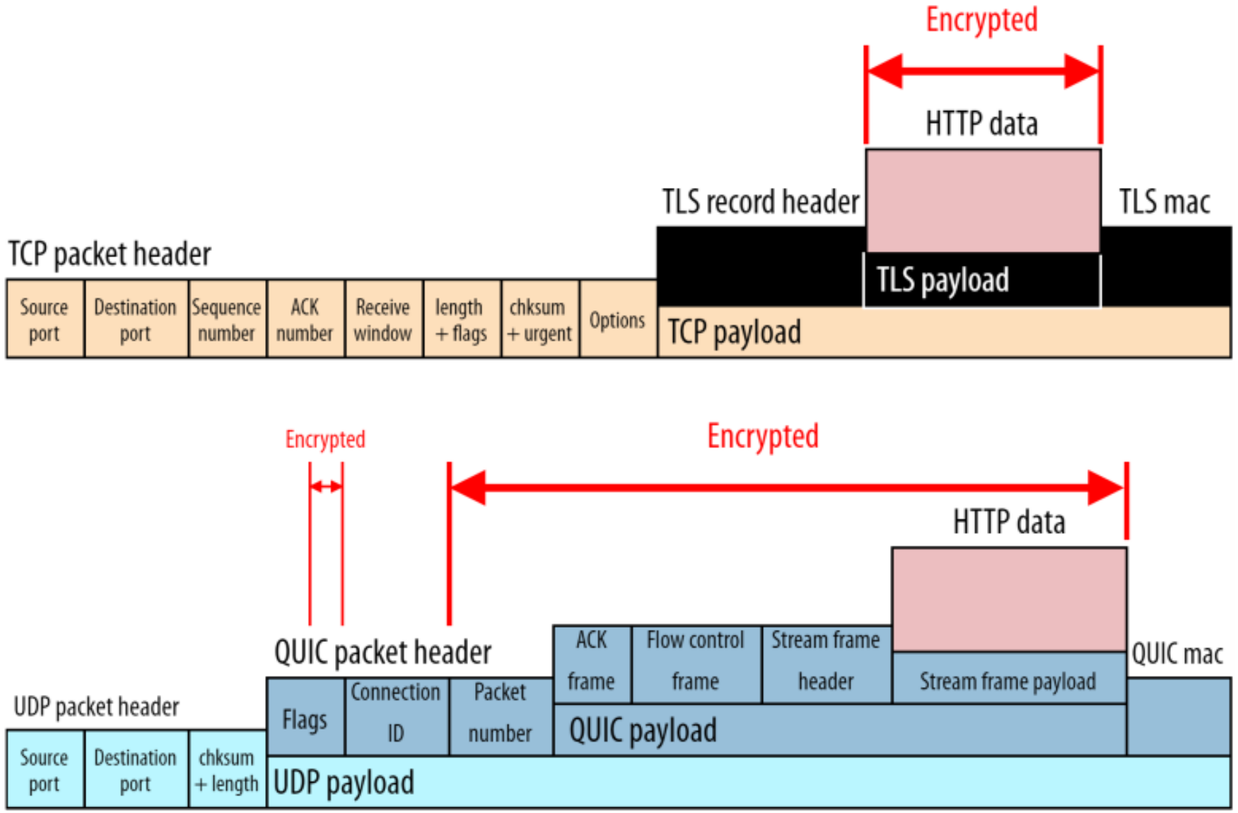
- To avoid privacy issues, e.g. to prevent hackers from following the physical movement of a smartphone user by tracking the unencrypted CID across networks, QUIC uses a list of connection identifiers instead of just one.



Encryption

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

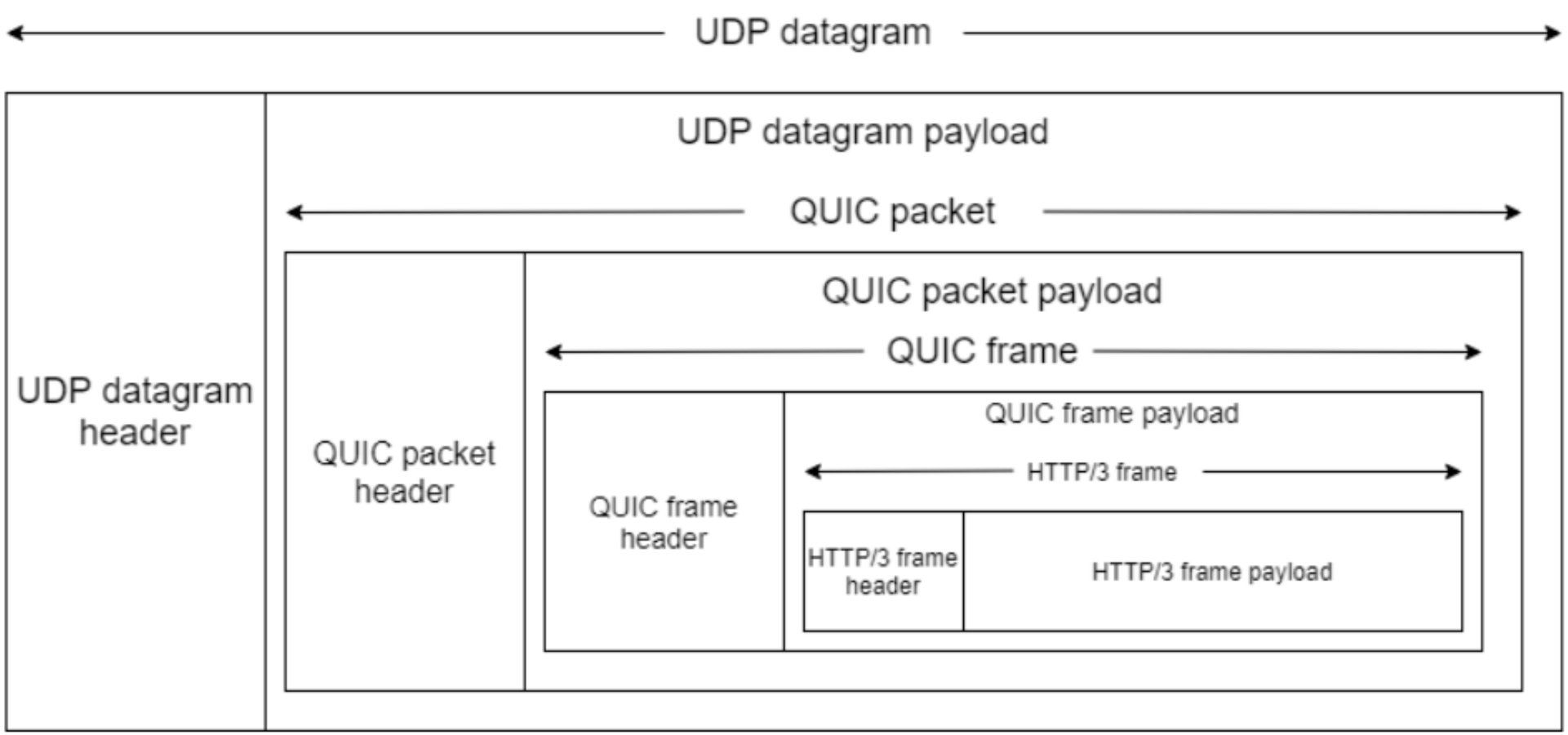
Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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

Ollscoil na Gaillimhe
University of Galway

- 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!

University
*of*Galway.ie

Ollscoil na Gaillimhe
University of Galway

CT 420
Real-Time Systems

Congestion Control in QUIC

Dr. Jawad Manzoor
Assistant Professor
School of Computer Science

University *of* Galway.ie

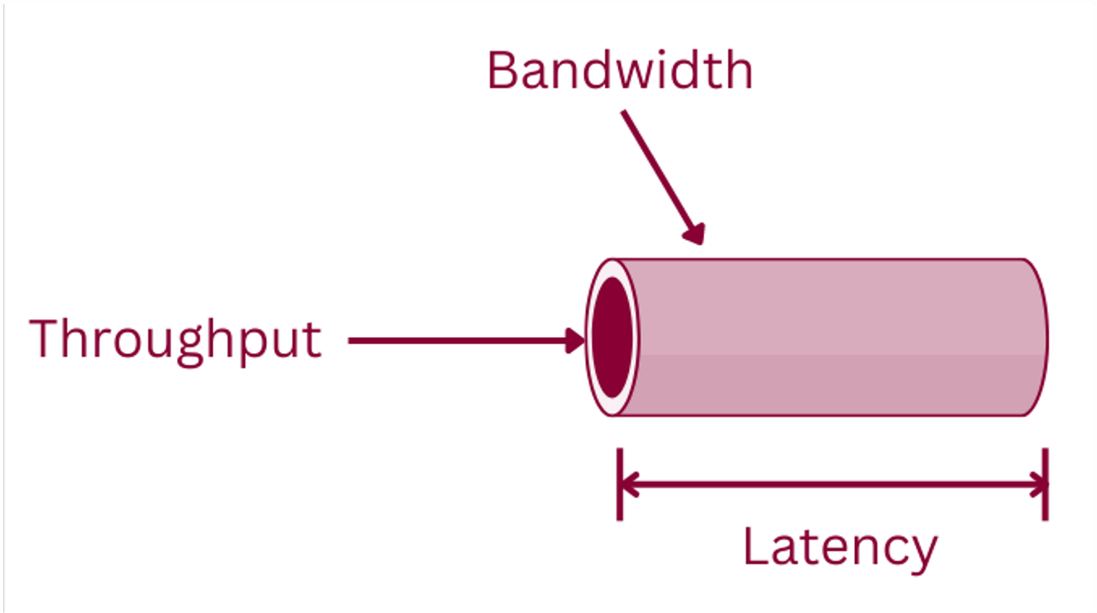
Contents

Ollscoil na Gaillimhe
University of Galway

- Need for congestion control
- Loss-based algorithms
- Delay-based algorithms
- QUIC congestion control

Web Performance Metrics

- Web performance is usually measured using the following network aspects:
  - Latency
  - Bandwidth
  - Throughput



Latency

Ollscoil na Gaillimhe
University of Galway

- We will often need quite a few round trips to load even a single file, due to features such as congestion control.
- Even low latencies of less than 50 milliseconds can add up to considerable delays.
- This is one of the main reasons why content delivery networks (CDNs) exist: They place servers physically closer to the end user in order to reduce latency, and thus delay, as much as possible.

Bandwidth

Ollscoil na Gaillimhe
University of Galway

- Bandwidth measures the amount of data that is able to pass through a network at a given time.
- It is measured in bits per second (bps), such as megabits per second (Mbps) and gigabits per second (Gbps).

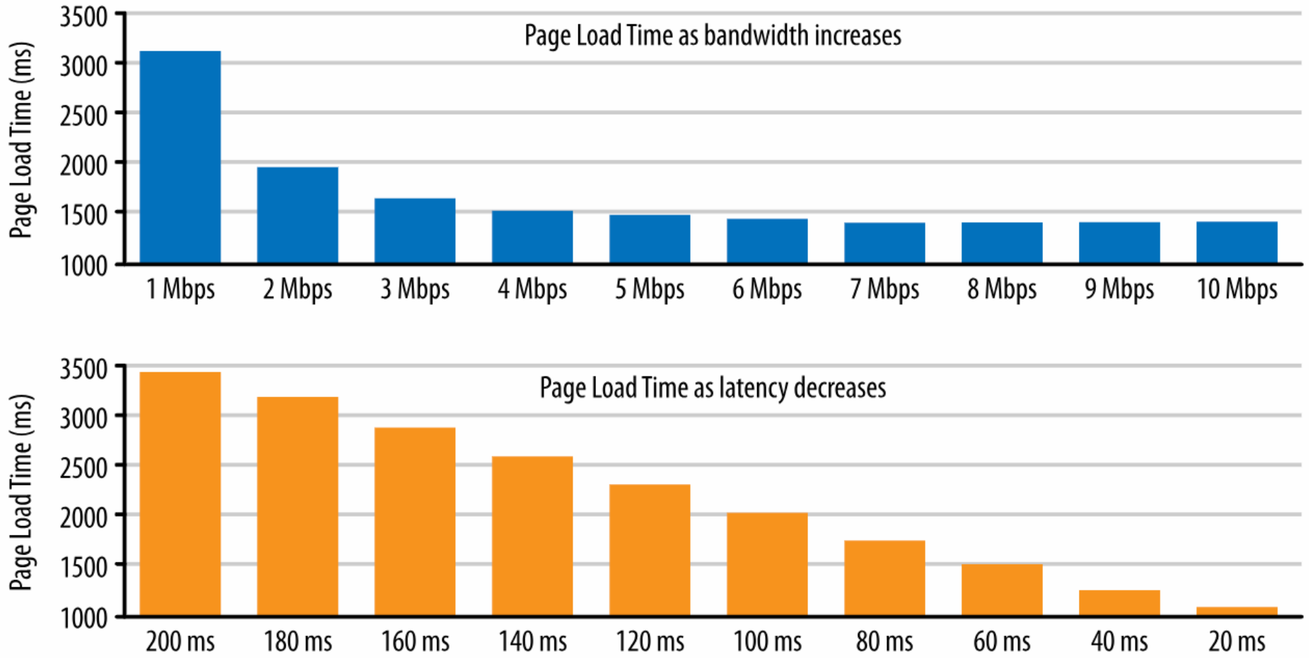
Throughput

Ollscoil na Gaillimhe
University of Galway

- Throughput is the number of data packets that are able to reach their destination within a specific period of time.
- Throughput can be affected by a number of factors, including bus or network congestion, latency, packet loss/errors, and the protocol used.
  - The system will likely experience low throughput even with a high bandwidth

Performance Bottleneck

- Access to higher bandwidth data rates is always good
  - video and audio streaming
  - large data transfer
- However, latency is the limiting factor for everyday web browsing
  - requires fetching hundreds of relatively small resources from dozens of different hosts



Class Activity

Ollscoil na Gaillimhe
University of Galway

- How do bandwidth and latency affect these applications?
- Cloud Gaming
  - Bandwidth: Medium (Requires around 5Mbps per device)
  - Latency: High impact (vital to a good experience of online games—especially in fast-paced games)
- Streaming
  - Bandwidth: High (4K stream averages around 25Mbps)
  - Latency: Medium impact
- Video chat
  - Bandwidth: Medium impact (Requires around 5Mbps. Low bandwidth will reduce the quality of your chat e.g. FaceTime or Skype
  - Latency: High impact (High latency causes sync issues and freezing)
- Browsing
  - Bandwidth: Medium impact
  - Latency: High impact (High delay causes long page load times and makes websites feel unresponsive)

Deciding Sending Rate

Ollscoil na Gaillimhe
University of Galway

- **Why can't we transmit data between end points at the highest rate?**

- One aspect of performance is about how efficiently a transport protocol can use a network's full (physical) bandwidth.
- Reliable transports don't just start sending data at full rate, because this could end up congesting the network.
- Each network link has only a certain amount of data it can process every second.
  - Sending excessive data will lead to packet loss
  - Lost packets need to be retransmitted and can seriously affect performance in high latency networks.

Estimating Link Capacity

Ollscoil na Gaillimhe
University of Galway

- **How to know about the available bandwidth between end-points?**

- We don’t know up front how much the maximum bandwidth will be.
- It depends on a bottleneck somewhere in the end-to-end connection, but we cannot predict or know where this will be. The Internet also doesn’t have mechanisms (yet) to signal link capacities back to the endpoints.
- Even if we knew the available physical bandwidth, that wouldn’t mean we could use all of it ourselves.
  - Several users are typically active on a network concurrently, each of whom need a fair share of the available bandwidth.

Congestion Control

Ollscoil na Gaillimhe
University of Galway

- In the 1980s when the internet was still run by the government and TCP was young, engineers were learning about bad TCP behavior and networks.
- At the time TCP with no congestion control, would occasionally "collapse" whole segments of the internet.
- Clients were programmed to respect the capacity of the machine at the other end of the connection (flow control) but not the capacity of the network.
- In 1986 in an especially bad incident, backbone links across the network, under incredible congestion, passed only 40bps, 1/1000th of their rated 32kbps capacity.
- The internet was suffering congestion collapse.

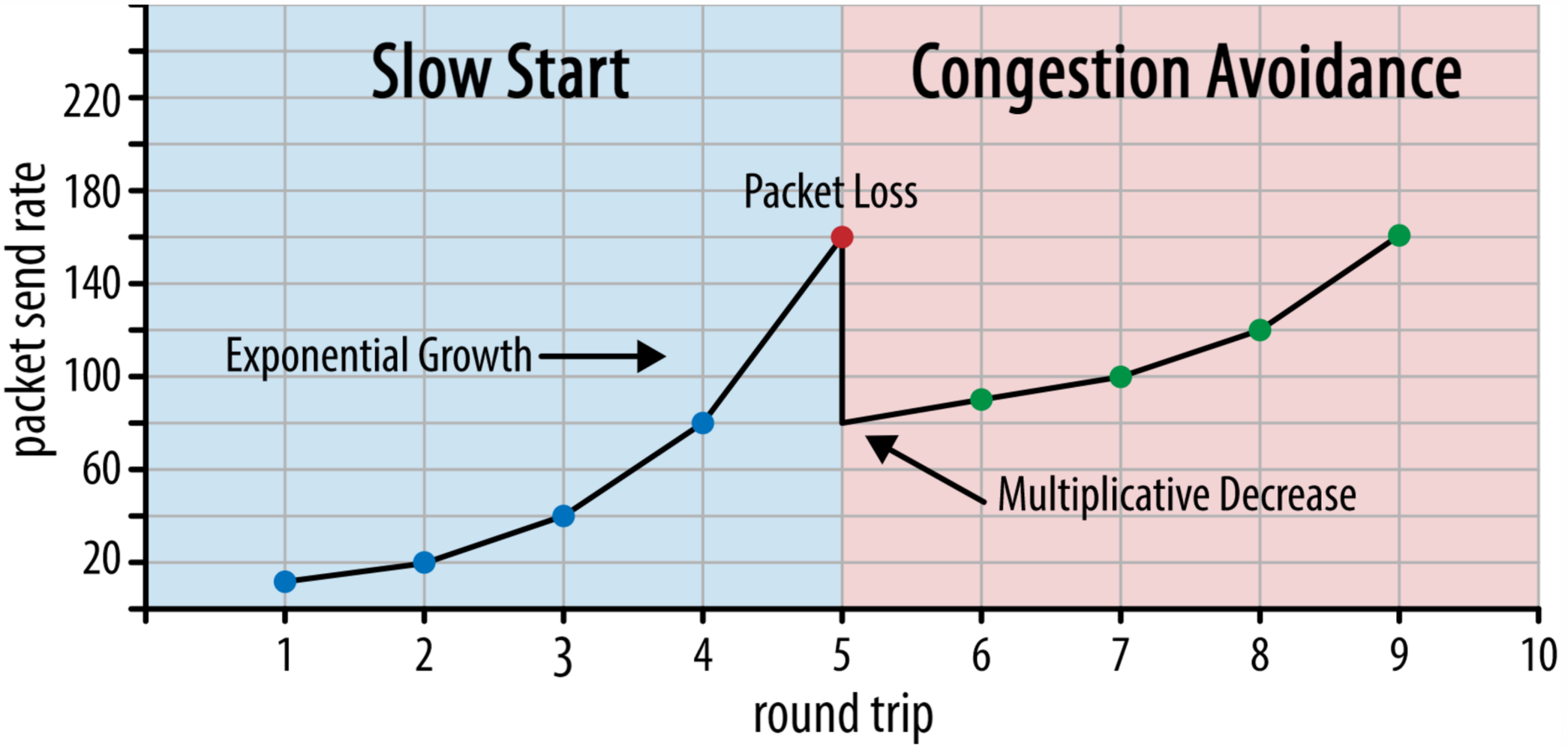
Congestion Control

Ollscoil na Gaillimhe
University of Galway

- In the network transport area, congestion control is how to decide how much data the connection can send into the network.
- Reliable transport protocols such as TCP and QUIC constantly try to discover the available bandwidth over time by using congestion control algorithm.

- Terminology
- MSS
  - Maximum segment size, is the largest data payload that a device will accept from a network connection
- CWND size
  - The congestion window (cwnd) is a sender-side limit on the amount of data the sender can transmit into the network before receiving an acknowledgment (ACK)

Congestion Control



Slow Start Phase

Ollscoil na Gaillimhe
University of Galway

- A connection is started slow
- It waits one round trip to receive acknowledgements of these packets
- If they are all acknowledged, this means the network has capacity, and the send rate is increased every iteration (usually doubled).

CWND Growth in Slow Start

Ollscoil na Gaillimhe
University of Galway

- MSS = 1460 bytes
- CWND = 10 segments
- What is the CWND size in KB after 5 round trips?

Round Trip	MSS	CWND (# segments)	CWND (KB)
1	1460	10	14
2	1460	20	28
3	1460	40	57
4	1460	80	114
5	1460	160	228

Congestion Avoidance Phase

Ollscoil na Gaillimhe
University of Galway

- The send rate continues to grow until some packets are not acknowledged (which indicates packet loss and network congestion).
- On packet loss, the send rate is slashed, and afterward it is gradually increased in much smaller increments.
- This reduce-then-grow logic is repeated for every packet loss.

Congestion Control Strategies

Ollscoil na Gaillimhe
University of Galway

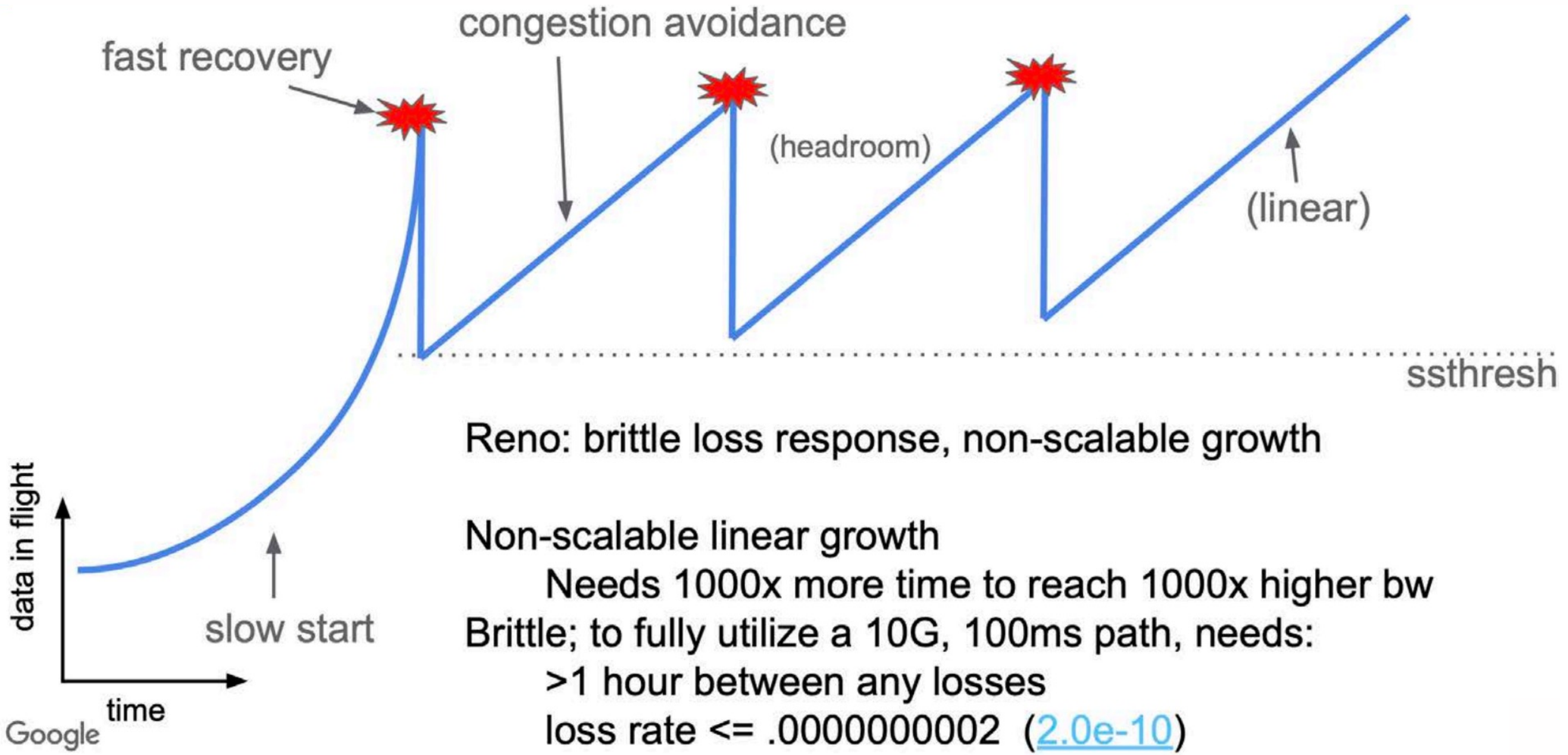
- Largely there are two categories:
- Loss-based congestion control - where the congestion control responds to a packet loss event. E.g. Reno and CUBIC
- Delay-based congestion control - the algorithm tries to find a balance between the bandwidth and RTT increase and tune the packet send rate. E.g. Vegas and BBR

Reno

Ollscoil na Gaillimhe
University of Galway

- Reno (often referred as NewReno) is a standard congestion control for TCP and QUIC .
- Reno starts from "slow start" mode which increases the CWND (limits the amount of data a TCP can send) roughly 2x for every RTT until the congestion is detected.
- When packet loss is detected, it enters into the “recovery” mode until packet loss is recovered.
- When it exits from recovery (no lost ranges) and CWND > SSTHRESH (slow start threshold), it enters into the "congestion avoidance" mode where the CWND grows slowly (roughly a full packet per RTT) and tries to converge on a stable CWND.
- A “sawtooth” pattern is observed when you make a graph of the CWND over time.

Reno



Reno: brittle loss response, non-scalable growth

Non-scalable linear growth

Needs 1000x more time to reach 1000x higher bw

Brittle; to fully utilize a 10G, 100ms path, needs:

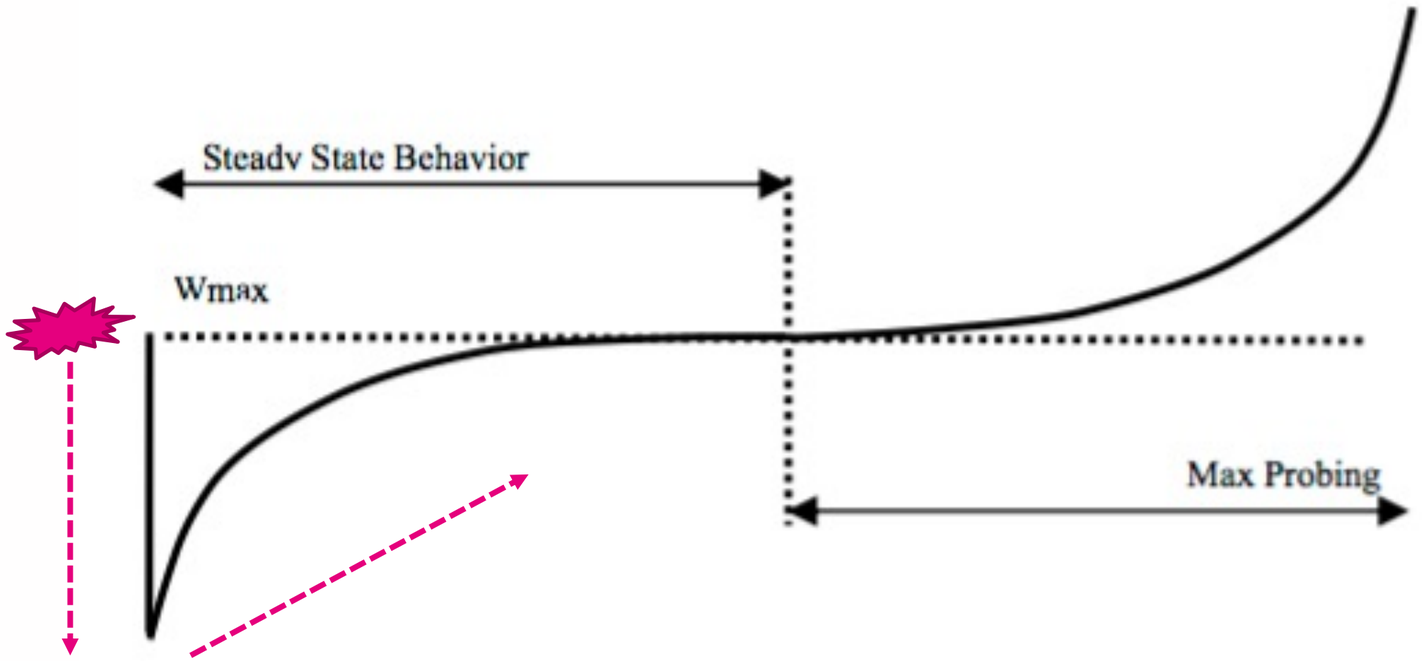
>1 hour between any losses

loss rate <= .0000000002 (2.0e-10)

Slide from Google presentation at IETF 104

Cubic

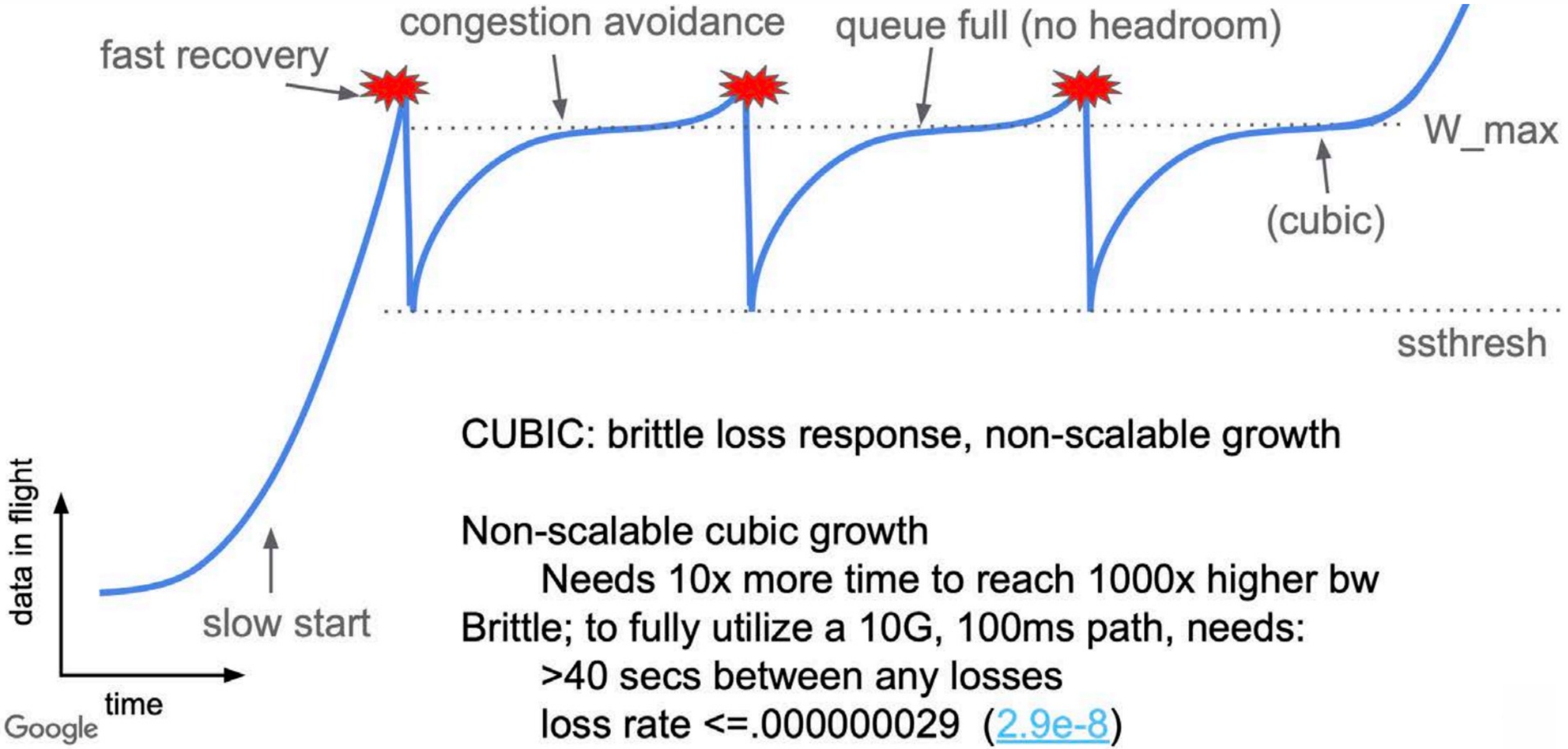
- Cubic is defined in RFC8312 and implemented in many OS including Linux, BSD and Windows.
- The difference from Reno is that during congestion avoidance its CWND growth is based on a cubic function instead of a linear function as follows:



Cubic

- Wmax is the value of CWND when the congestion is detected.
- Then it will reduce the CWND by 30% and then the CWND starts to grow again using a cubic function as in the graph, approaching Wmax aggressively in the beginning in the first half but slowly converging to Wmax later.
- This makes sure that CWND growth approaches the previous point carefully and once we pass Wmax, it starts to grow aggressively again after some time to find a new CWND (this is called "Max Probing").

Cubic



CUBIC: brittle loss response, non-scalable growth

Non-scalable cubic growth
- Needs 10x more time to reach 1000x higher bw

Brittle; to fully utilize a 10G, 100ms path, needs:
- >40 secs between any losses
- loss rate <=.000000029 (2.9e-8)

Slide from Google presentation at IETF 104

HyStart++

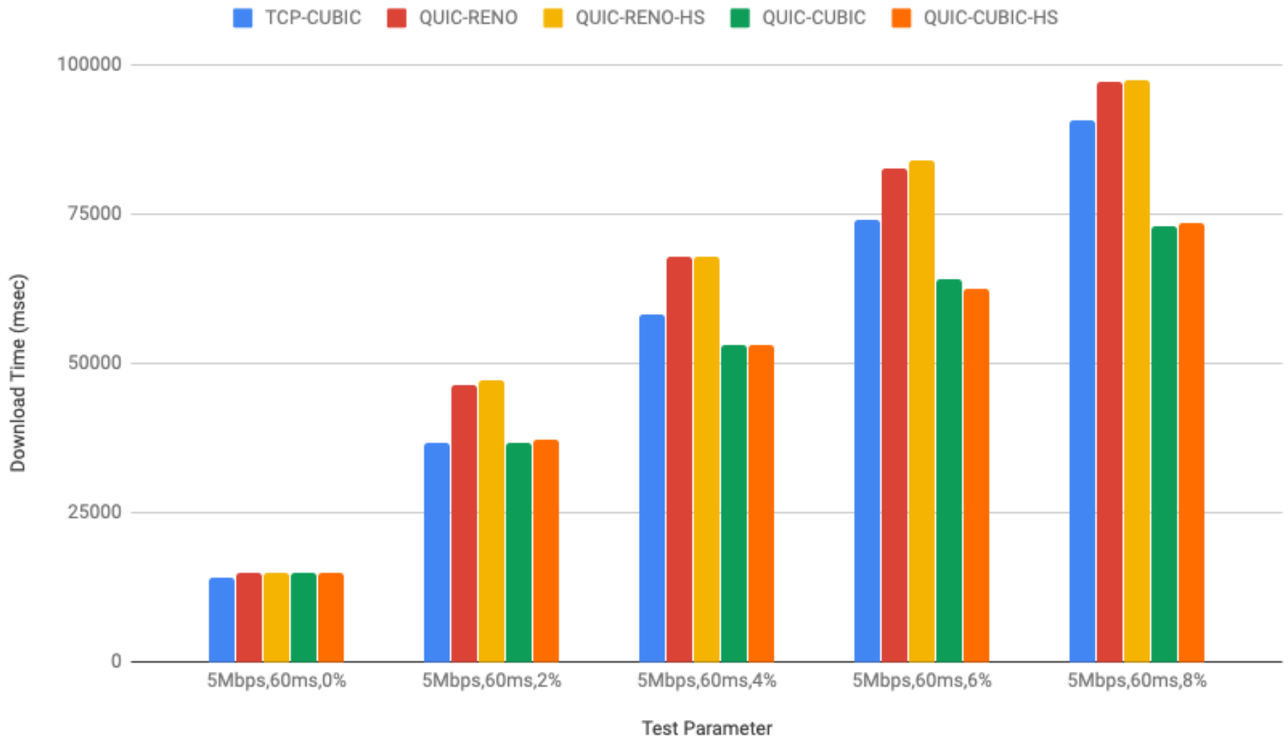
Ollscoil na Gaillimhe
University of Galway

- The authors of CUBIC made a separate effort to improve slow start
- It is based on RTT delay samples - when the RTT is increased during slow start and over the threshold, it exits slow start early and enters into LSS (Limited Slow Start).
- LSS grows the CWND faster than congestion avoidance but slower than Reno slow start.

QUIC vs TCP congestion control

Ollscoil na Gaillimhe
University of Galway

- Performance comparison of QUIC congestion control variants.
  - with 0% packet loss TCP and QUIC are almost doing the same
  - As packet loss increases QUIC CUBIC performs better than TCP CUBIC.
  - With HyStart++, overall performance is the same

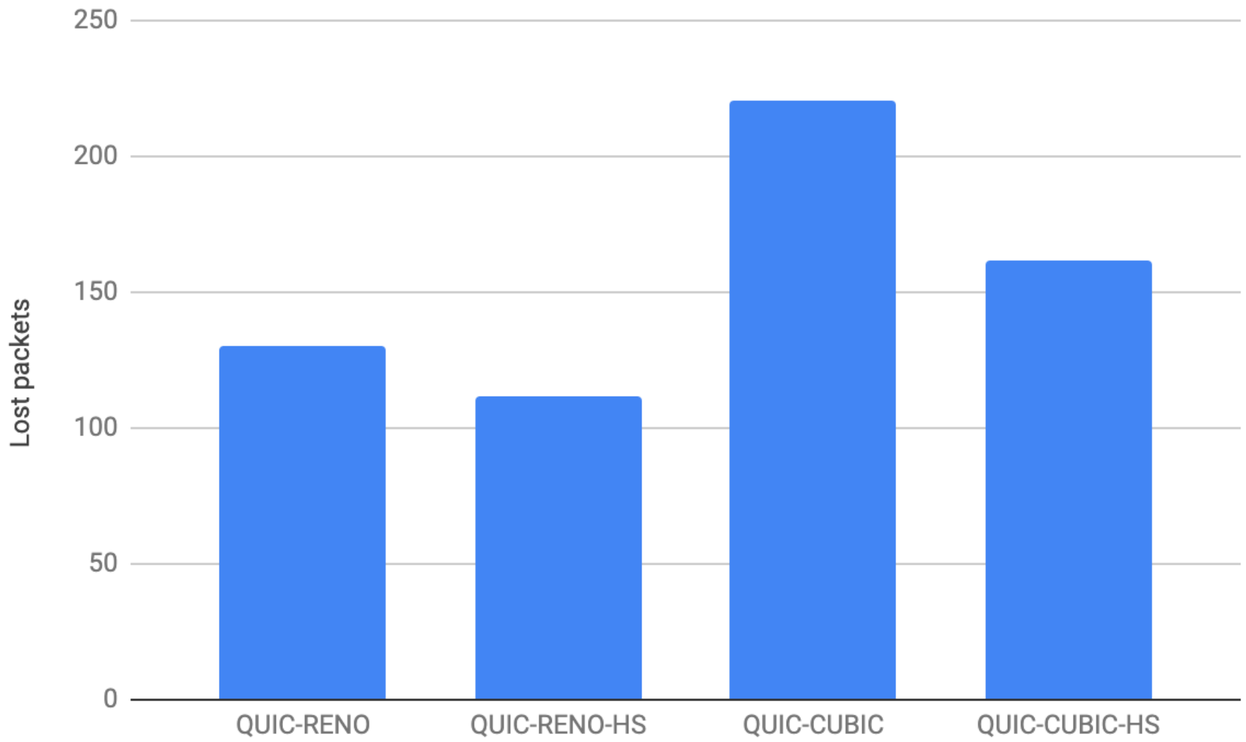


Source: cloudflare.com

QUIC vs TCP congestion control

Ollscoil na Gaillimhe
University of Galway

- Performance comparison of QUIC congestion control variants.
  - Compared with Reno, CUBIC can create more packet loss in general.
  - HyStart++ significantly reduces packet loss.



Source: cloudflare.com

Question

Ollscoil na Gaillimhe
University of Galway

- Is packet loss always an indication of congestion?
- Are there any other causes of packet loss?

- Packet loss may not always mean that there is congestion in the network.
  - For example, a packet may be lost due:
  - Transient radio interference
  - Software bugs
  - Hardware issues

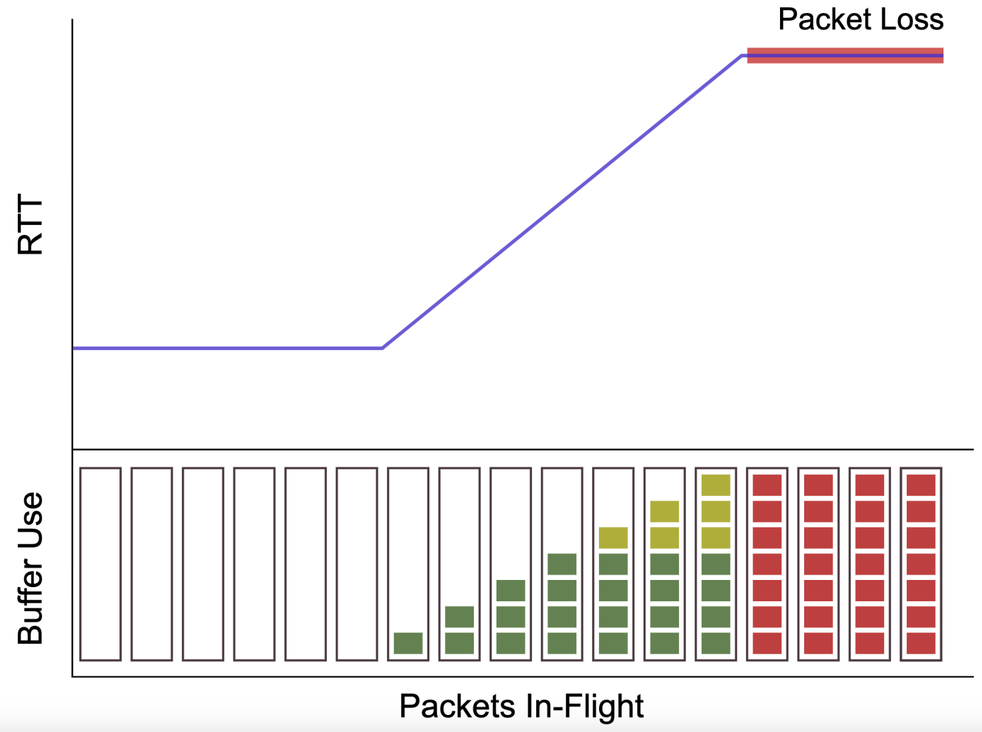
Delay based Congestion Control

Ollscoil na Gaillimhe
University of Galway

- So far, we have discussed loss-based congestion control
- Cubic and other loss-based algorithms do not distinguish between spurious packet losses and real congestion, reducing their send rate in both cases.
- Another issue is that when bottleneck buffers are large, loss-based congestion control keeps them full, causing bufferbloat.
- The main idea behind delay-based congestion control is to detect network congestion using packet delays.

BBR

- BBR (Bottleneck Bandwidth and RTT) is new congestion control algorithm developed by Google in 2016.
- BBR monitors RTT and attempts to detect when packets are getting getting buffered based on timing. The larger the increase in RTT, the longer packets are being buffered, and the higher the likelihood the network is entering congestion.



BBR

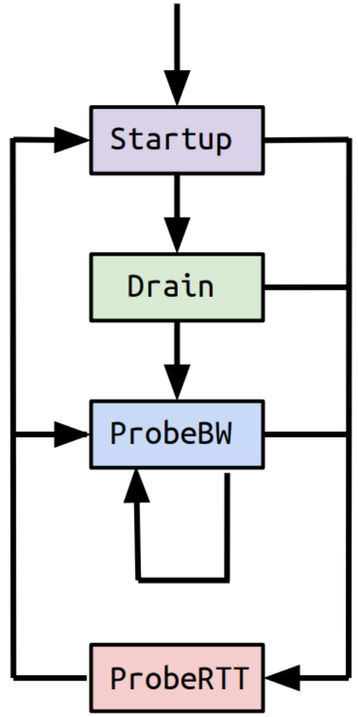
Ollscoil na Gaillimhe
University of Galway

- BBR limits its number of packets in flight to be a multiple of the bandwidth-delay product (BDP).
- BBR strives to optimize both throughput and latency by estimating the bottleneck bandwidth and RTT to compute a pacing rate.
- The goal is is to avoid filling up the bottleneck buffer, which might induce bufferbloat.
- A description of the BBR algorithm was published in the 2016.
- The latest version is BBR v3, that includes several improvements over BBR v1 and v2

BBR Phases

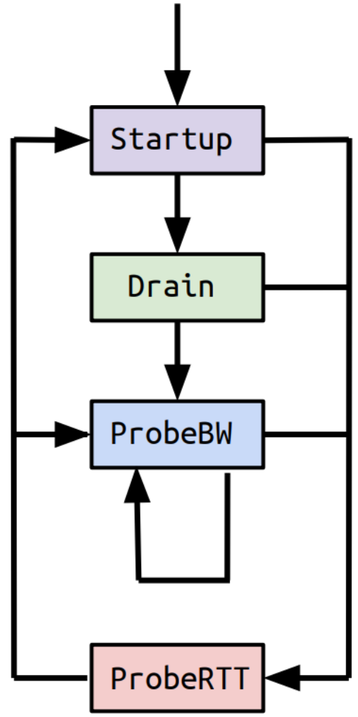
Ollscoil na Gaillimhe
University of Galway

- BBR is split up into four different phases; Startup, Drain, ProbeBW and ProbeRTT.
- The first phase, Startup, uses the same exponential starting behavior as CUBIC, which doubles the sending rate with each round-trip.
- BBR assumes the bottleneck bandwidth to have been reached once the measured bandwidth does not increase any further.
- This observation causes a queue to form at the bottleneck, due to it being delayed by one RTT.
- BBR then moves to the Drain phase, where it attempts to drain the queue by reducing the pacing gain temporarily.

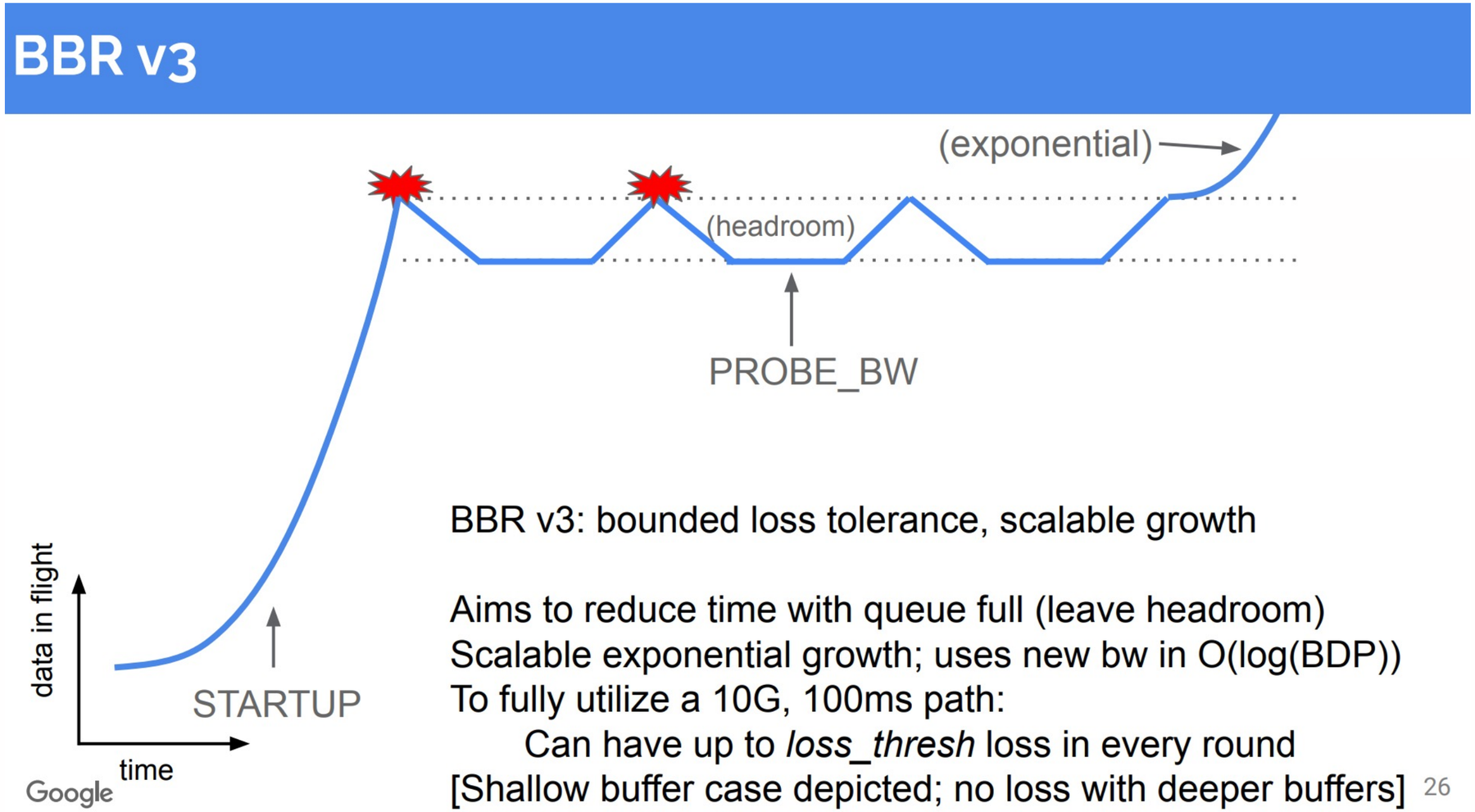


BBR Phases

- Next, BBR moves to the third phase, ProbeBW, which is also its steady state, since it spends most of its time in it probing and utilizing the pipe's bandwidth in a fair manner, with a small, bounded queue.
- If a flow hasn't seen an RTT sample that matches or decreases its min_rtt estimate for 10 seconds, BBR stops the bandwidth probing and moves on to the ProbeRTT phase, where the bandwidth gets reduced to four packets to drain any queues and get a more accurate estimation of the RTT.
- ProbeRTT is run for 200 ms + one RTT, where it will afterwards go back to the steady state, ProbeBW.



BBR v3



Slide from Google presentation at IETF 104

BBRv3 performance impact for public Internet traffic

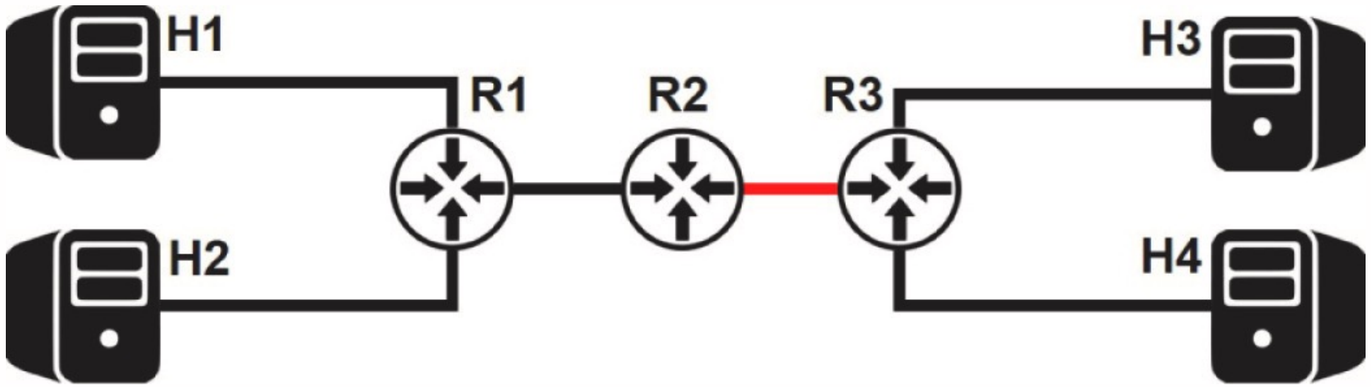
Ollscoil na Gaillimhe
University of Galway

- Impact of BBRv3 vs BBRv1 on Google.com and YouTube TCP public Internet traffic:
  - Lower retransmit rate (12% reduction)
  - Slight latency improvement (0.2% reduction) for:
    - Google.com web search
    - Starting YouTube video playback
  - Latency wins seem to be from lower loss rate (less/faster loss recovery)

Performance Evaluation

Ollscoil na Gaillimhe
University of Galway

- You can test the performance of various congestion control algorithms using simulations
- Mininet testbed
  - The bottleneck link is the link between R2 and R3 (marked in red)
  - Delay and packet loss are configured on the link between R1 and R2
  - Traffic is generated between sender and receiver pairs (H1 and H3, H2 and H4)
  - Different congestion control algorithms are used for each pair



QUIC Congestion Control

Ollscoil na Gaillimhe
University of Galway

❑ Incorporating existing algorithms

- Similar to TCP congestion control, QUIC utilizes a window-based congestion control scheme
- QUIC does not aim to develop its own new congestion control algorithms, nor use any specific one.
- QUIC has pluggable congestion control which allows the sender to choose the best algorithm for its application.
- To avoid unnecessary congestion window reduction, QUIC does not collapse the congestion window unless it detects persistent congestion using both loss-based and delay-based mechanisms.

QUIC RTT Estimation

Ollscoil na Gaillimhe
University of Galway

❑ Estimating the Round-Trip Time

- At a high level, an endpoint measures the time from when a packet was sent to when it is acknowledged, which allows to calculate the actual time used in transmitting a packet over the network.
- An endpoint computes the following three values for each path:
  - smoothed_rtt – a stable average of the RTT measurements over a period of time
  - min_rtt - the minimum value over a period of time
  - rttvar - and the mean deviation (referred to as "variation") in the observed RTT samples

QUIC Loss Detection

Ollscoil na Gaillimhe
University of Galway

❑ Detecting packet loss

- QUIC uses two main signals to detect packet loss:
  i. Packet threshold loss detection
  ii. Time threshold loss detection

QUIC Loss Detection

Ollscoil na Gaillimhe
University of Galway

❑ Packet Threshold Loss Detection

- QUIC considers a packet lost if a later packet has been acknowledged, and it was sent at least `kPacketThreshold` packets after the one we're considering.
- The recommended initial value for the packet reordering threshold is 3.

- Example:
  - You send packets 1, 2, 3, 4, 5.
  - You receive an ACK for 5, but 2 is still unacknowledged.
  - Since 5 was acknowledged, it means 3 new packets have been acknowledged since 2 was sent, therefore 2 is declared lost.

```
if (largest_acked_packet_number - packet_number >= kPacketThreshold)
        => packet is lost
```

QUIC Loss Detection

Ollscoil na Gaillimhe
University of Galway

❑ Time Threshold Loss Detection

- A packet is considered lost if enough time has passed since it was sent, without being acknowledged.
- Example:
  - If a packet was sent earlier than:

```
current_time - max(kTimeThreshold * max(smoothed_rtt, latest_rtt), kGranularity)
```

`=> packet is lost`

  - The recommended time threshold (kTimeThreshold), expressed as an RTT multiplier, is 9/8 (i.e., 1.125)
  - The recommended value of the timer granularity (kGranularity) is 1 millisecond

QUIC Packet Pacing

Ollscoil na Gaillimhe
University of Galway

- Packet pacing means spreading out the transmission of packets evenly over time, instead of sending a burst of packets all at once, that could lead to congestion
- QUIC packet pacing is a mechanism designed to regulate the rate at which packets are sent over the network.
- The pacing rate is typically derived from the current congestion window (cwnd) and the RTT.

```
Pacing rate = cwnd / smoothed_rtt
```

- Based on the pacing rate, QUIC calculates a delay between sending packets to smooth packet bursts.
  - Example: If you can send 100 packets per second, you send one packet every 10ms, instead of all at once.
- Packet pacing mechanism adapts to changes in network conditions, such as changes in available bandwidth or variations in round-trip times.

QUIC vs TCP Congestion Control

Ollscoil na Gaillimhe
University of Galway

- QUIC is more flexible and easier to evolve than TCP
- TCP is typically implemented in the operating system's kernel
  - Tuning congestion logic is usually only done by a select few developers, and evolution is slow
- QUIC implementations are in "user space"
  - They are flexible to modify
  - Experimenting with new congestion-control algorithms is easier
- RFC 9002 - QUIC Loss Detection and Congestion Control
  - https://datatracker.ietf.org/doc/rfc9002/

Acknowledgement

- https://hpbn.co/
- https://vt200.com/
- https://www.smashingmagazine.com/

Ollscoil na Gaillimhe
University of Galway

Ollscoil na Gaillimhe
University of Galway

Thank you for your attention!

University
*of*Galway.ie