

TCP modeling

- Given the congestion behavior of TCP can we predict what type of performance we should get?
- What are the important factors
 - Loss rate
 - Affects how often window is reduced
 - RTT
 - Affects increase rate and relates BW to window
 - RTO
 - Affects performance during loss recovery
 - MSS
 - Affects increase rate

TCP modeling

- Some additional assumptions
 - Fixed RTT
 - No delayed ACKs
- In steady state, TCP losses packet each time window reaches W packets
 - Window drops to $W/2$ packets
 - Each RTT window increases by $1 \text{ packet} \cdot W/2 * \text{RTT}$ before next loss

TCP continues to be an active topic of research

- Networks are heterogeneous
- TCP meltdown problem (outside the scope of this class)
 - Tunneling is currently en vogue
 - Nesting congestion control algorithms can be EXTREMELY bad
 - Answer:
 - Disable CC for all but one of the nested flows
- Bufferbloat problem
 - Huge buffers were popular for a minute... TCP doesn't care about queues/buffers, it cares about how many packets it can safely have in flight.
Bad mix

Evolution of transport layer functionality

TCP, UDP: principal transport protocols for 40 years

different “flavors” of TCP developed, for specific scenarios:

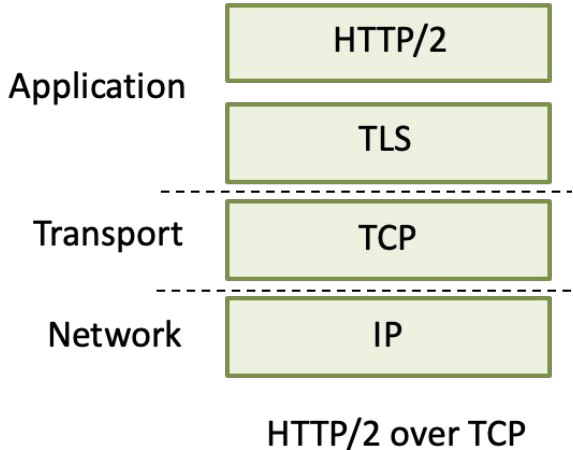
Scenario	Challenges
Long, fat pipes (large data transfers)	Many packets “in flight”; loss shuts down pipeline
Wireless networks	Loss due to noisy wireless links, mobility; TCP treat this as congestion loss
Long-delay links	Extremely long RTTs
Data center networks	Latency sensitive
Background traffic flows	Low priority, “background” TCP flows

moving transport-layer functions to application layer, on top of UDP

- HTTP/3: QUIC

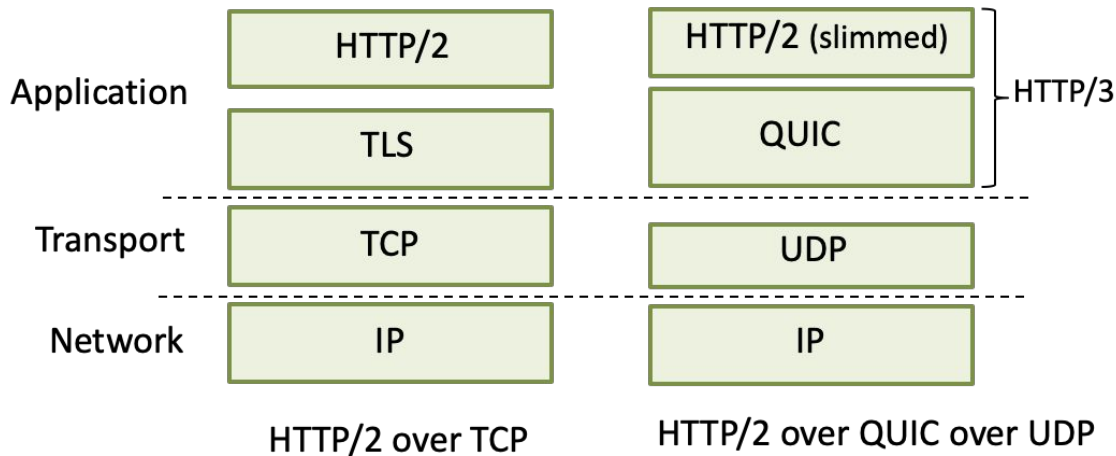
QUIC: Quick UDP Internet Connections

- application-layer protocol, on top of UDP
 - increase performance of HTTP
 - deployed on many Google servers, apps (Chrome, YouTube)



QUIC: Quick UDP Internet Connections

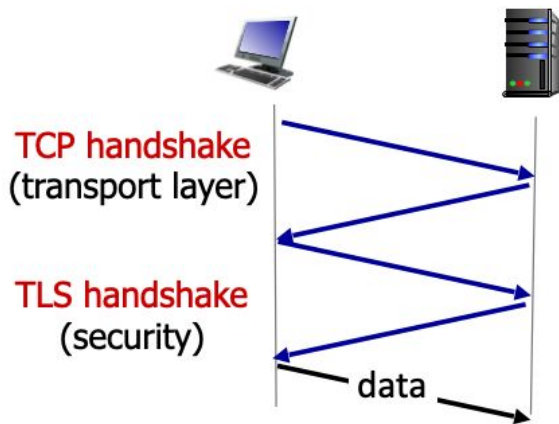
- application-layer protocol, on top of UDP
 - increase performance of HTTP
 - deployed on many Google servers, apps (Chrome, YouTube)



QUIC: Quick UDP Internet Connections

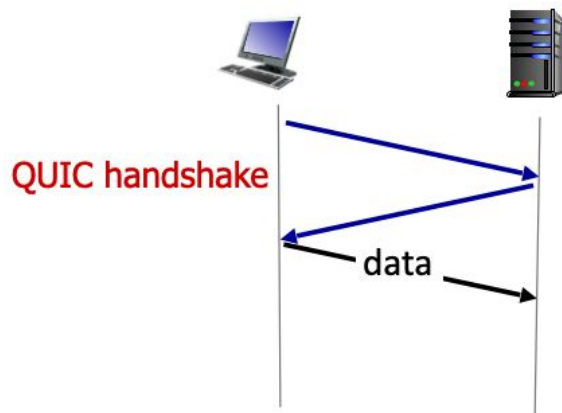
- adopts approaches from TCP for connection establishment, error control, congestion control
- error and congestion control: “Readers familiar with TCP’s loss detection and congestion control will find algorithms here that parallel well-known TCP ones.” [from QUIC specification]
- connection establishment: reliability, congestion control, authentication, encryption, state established in one RTT
- multiple application-level “streams” multiplexed over single QUIC connection
 - separate reliable data transfer, security
 - common congestion control

QUIC connection establishment



TCP (reliability, congestion control state)
+ TLS (authentication, crypto state)

- 2 serial handshakes



QUIC: reliability, congestion control,
authentication, crypto state

- 1 handshake

QUIC: streams: parallelism, no HOL blocking

Transport wrap up

- Muxing data streams to applications
- TCP vs UDP
 - Properties, pros and cons of each
- Reliable data transfer
- Fairness (max-min)
- Flow control
- Congestion control
- Congestion avoidance (TCP AIMD)

DNS

Leaving the transport layer for application layer

- DNS uses UDP or TCP
- Special protocol - not simply an application, it's a fundamental network protocol for making the Internet operate
- www.hawaii.edu ->
 - web3x-vip-www00.its.hawaii.edu ->
 - 128.171.133.5

DNS

- The Internet has one global system for:
 - Addressing hosts IP
(by design)
 - Naming hosts DNS
By accident, an afterthought

DNS

- The Internet has one global system for:
 - Addressing hosts IP
(by design)
 - Naming appreciated by humans
 - Provide little (if any) information about location
 - Naming hosts DNS
By accident, an afterthought
 - Numerical addresses appreciated by routers
 - Hierarchical, related to host location

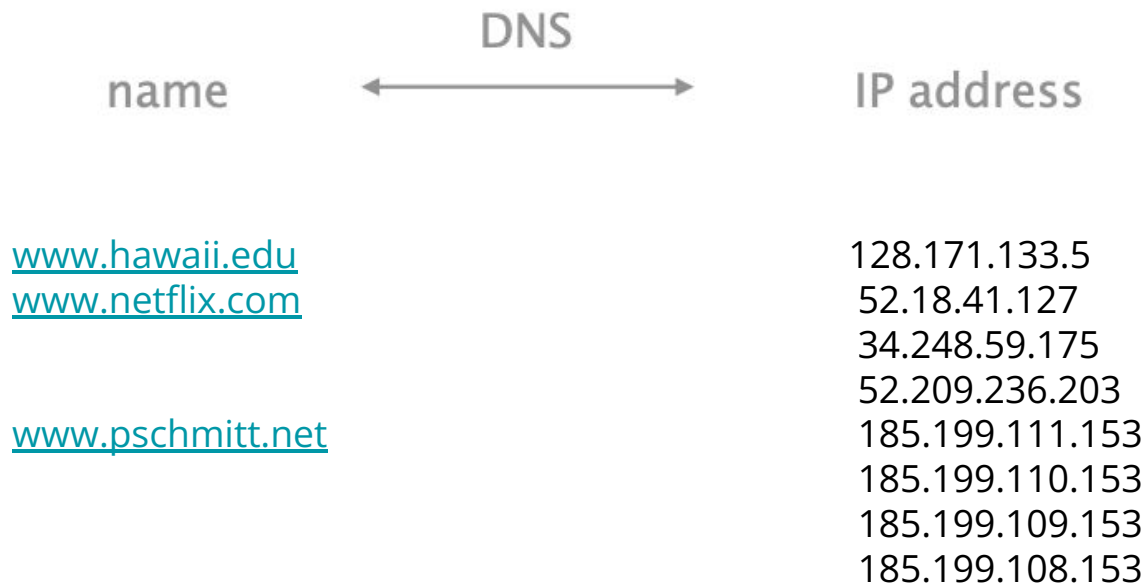
Using Internet services can be divided into four logical steps

1. A person has name of entity she wants to access www.hawaii.edu
2. She invokes an application to perform the task Chrome
3. The application invokes DNS to resolve the name into an IP address 128.171.133.5
4. The application invokes transport protocol to establish an app-to-app connection

The DNS system is a distributed database
which enables to resolve a name into an IP address



In practice, names can be mapped to more than one IP



In practice, names can be mapped to more than one IP

na

Why is this a good thing?

www.hawaii.edu

www.netflix.com

www.pschmitt.net

128.171.133.5

52.18.41.127

34.248.59.175

52.209.236.203

185.199.111.153

185.199.110.153

185.199.109.153

185.199.108.153

In practice, names can be mapped to more than one IP

na

Why is this a good thing?

Load balancing

Reduce latency by picking nearby servers

Tailored content based on requester's location/identity

www.hawaii.edu

www.netflix.com

www.pschmitt.net

128.171.133.5

52.18.41.127

34.248.59.175

52.209.236.203

185.199.111.153

185.199.110.153

185.199.109.153

185.199.108.153

In practice, IPs can be mapped by more than one name

