CS120: Computer Networks

Lecture 19. Other Topics in Transportation Layer

Haoxian Chen

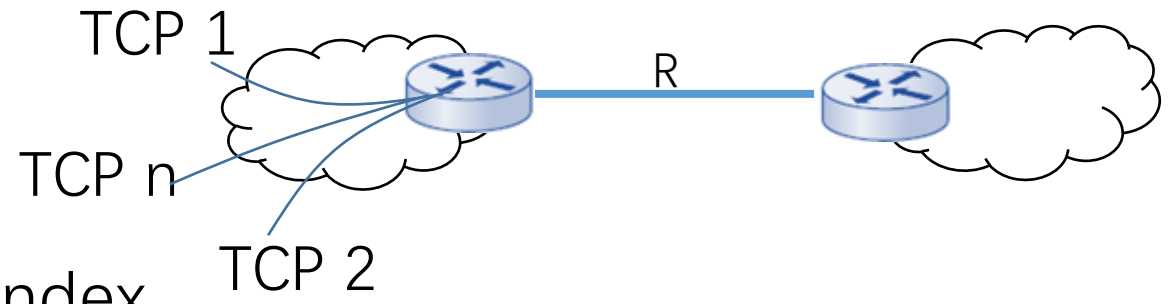
Slides adopted from: Zhice Yang

Outline

- TCP Fairness
- QUIC
- QoS

Evaluation Criteria

- Defining fairness is hard
  - In terms of a host, a TCP link, or an application ?
- TCP fairness goal: if **n** TCP sessions share same bottleneck link of bandwidth **R**, each should have average rate of ***R/n***



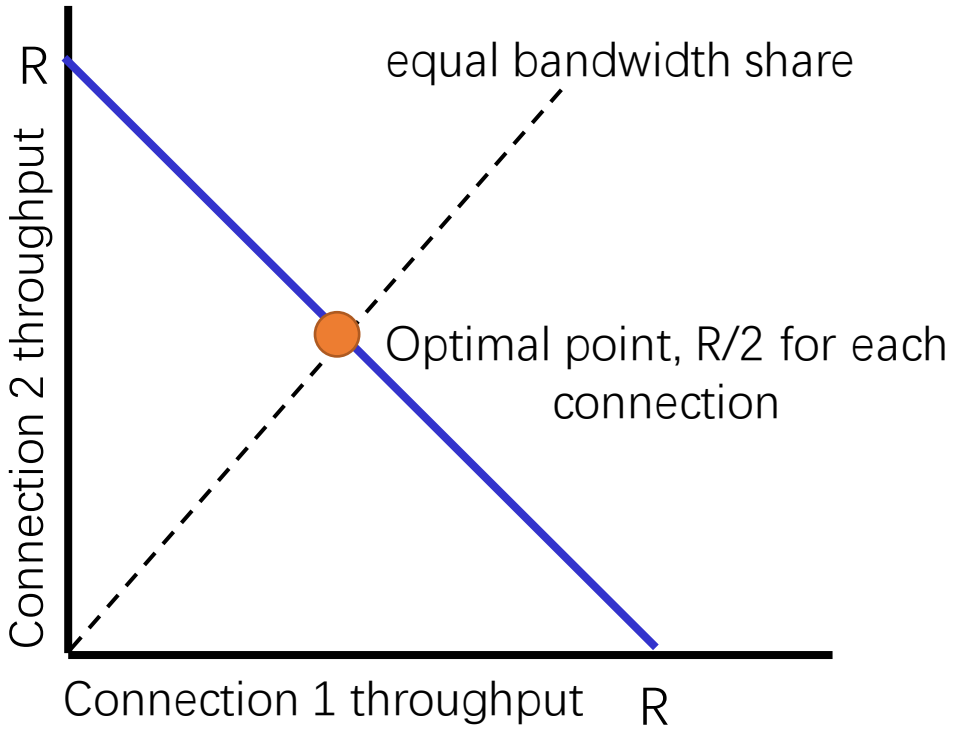
- Fairness Index
  - $f(x_1 \dots x_n) = \frac{(\sum x_i)^2}{n * \sum x_i^2}$

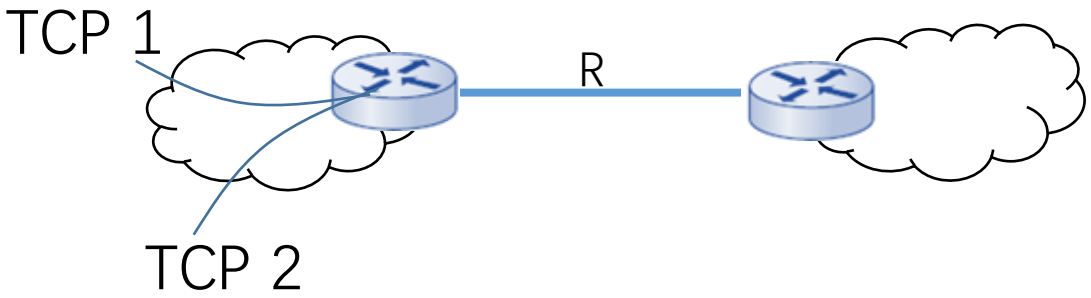
Fairness in TCP

- Consider the steady state, TCP uses a (linear) scheme to adjust its window cwnd
  - cwnd' = b*cwnd + a
- Possible Designs
  - Additive increase, additive decrease
  - Additive increase, multiplicative decrease (AIMD)
  - Multiplicative increase, additive decrease
  - Multiplicative increase, multiplicative decrease

Fairness in TCP

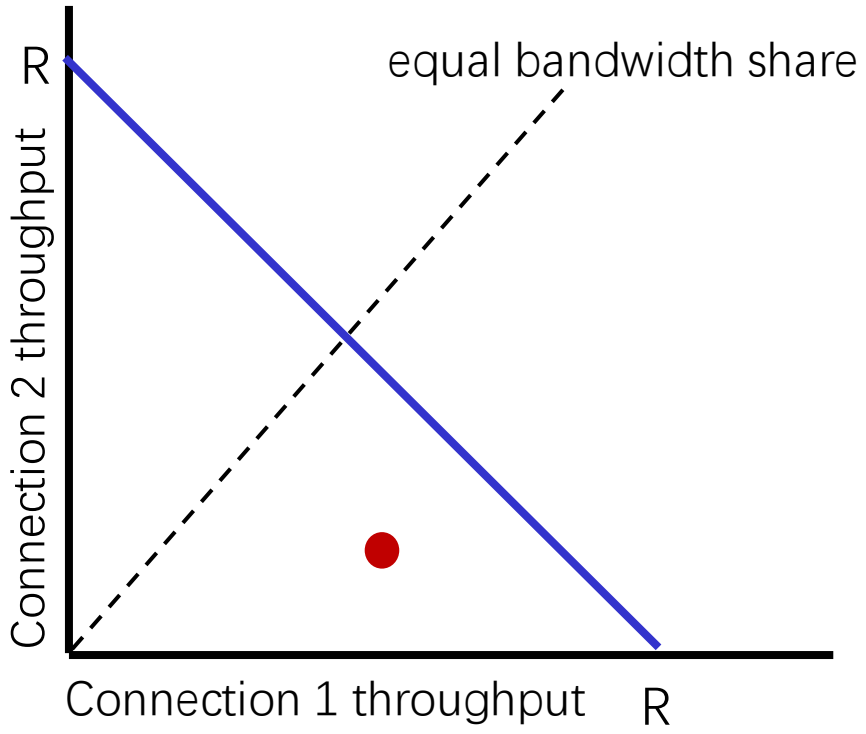
- Consider a case with two TCP connections





Fairness in TCP

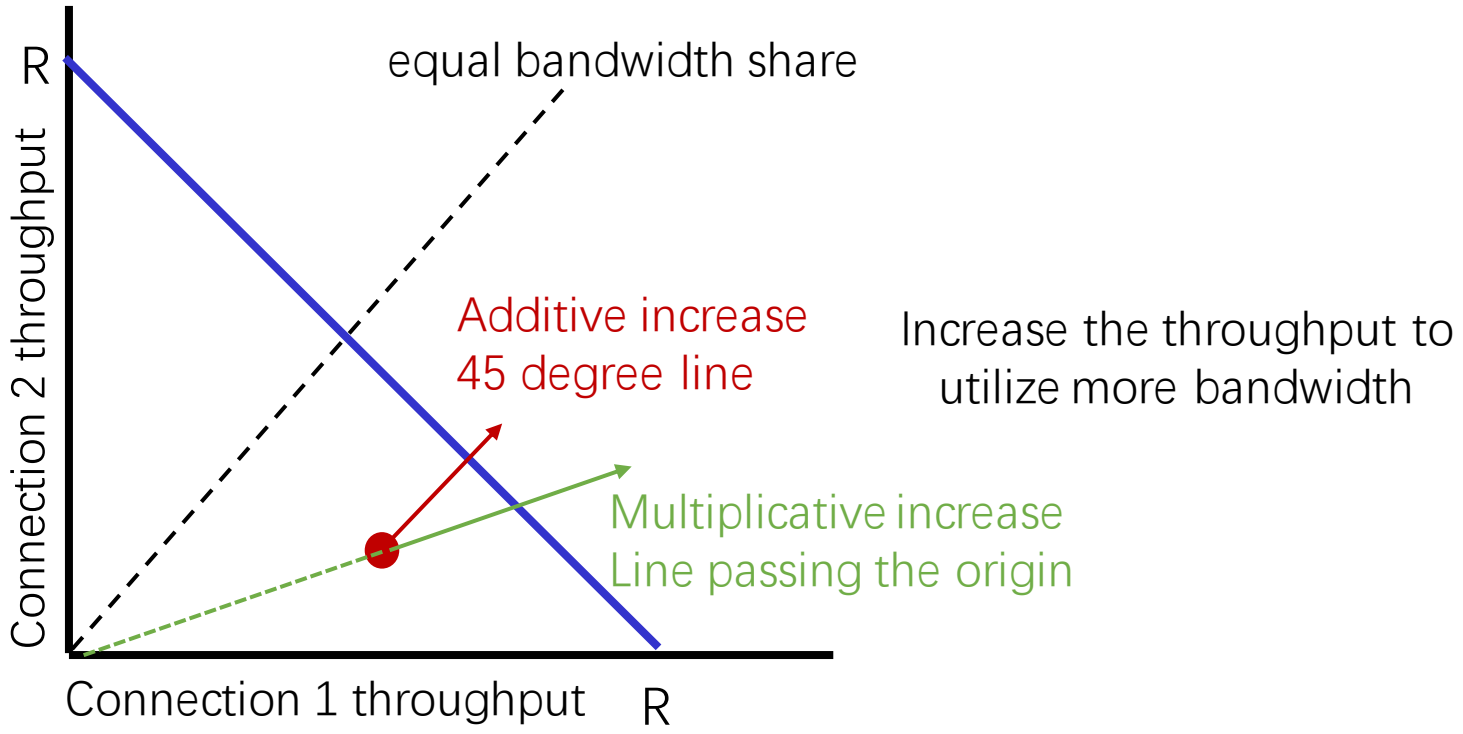
- Consider a case with two TCP connections



Increase the throughput to utilize more bandwidth

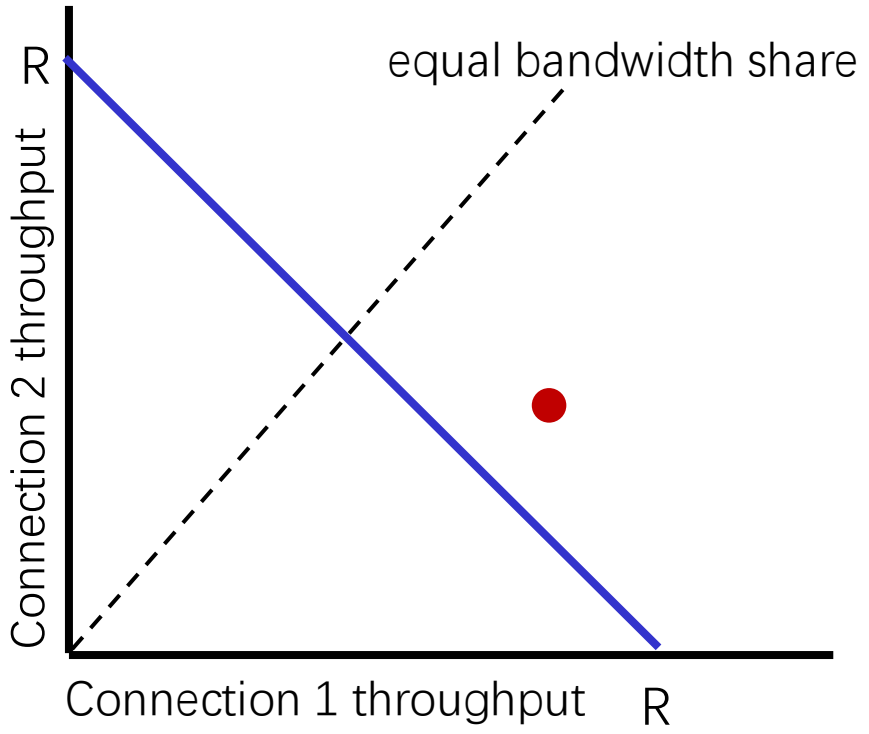
Fairness in TCP

- Consider a case with two TCP connections



Fairness in TCP

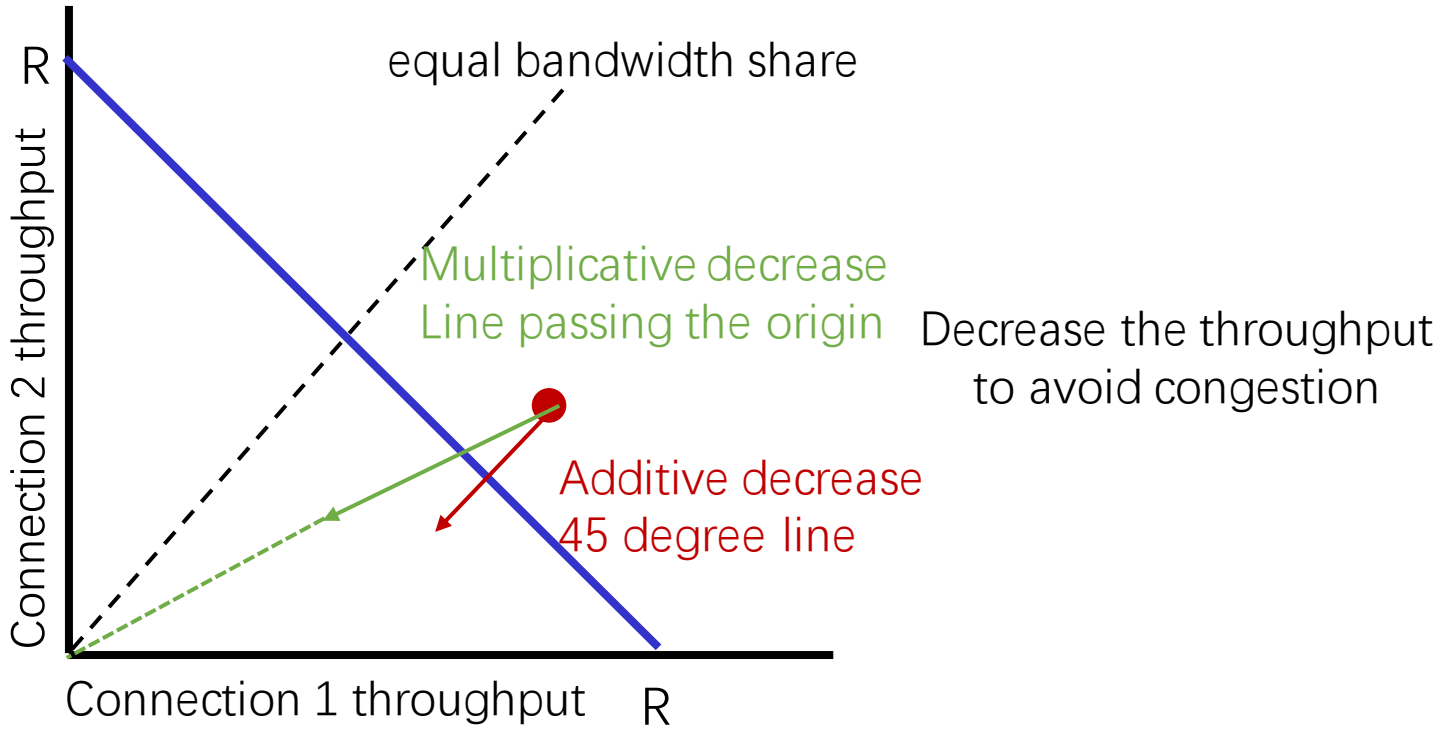
- Consider a case with two TCP connections



Decrease the throughput to avoid congestion

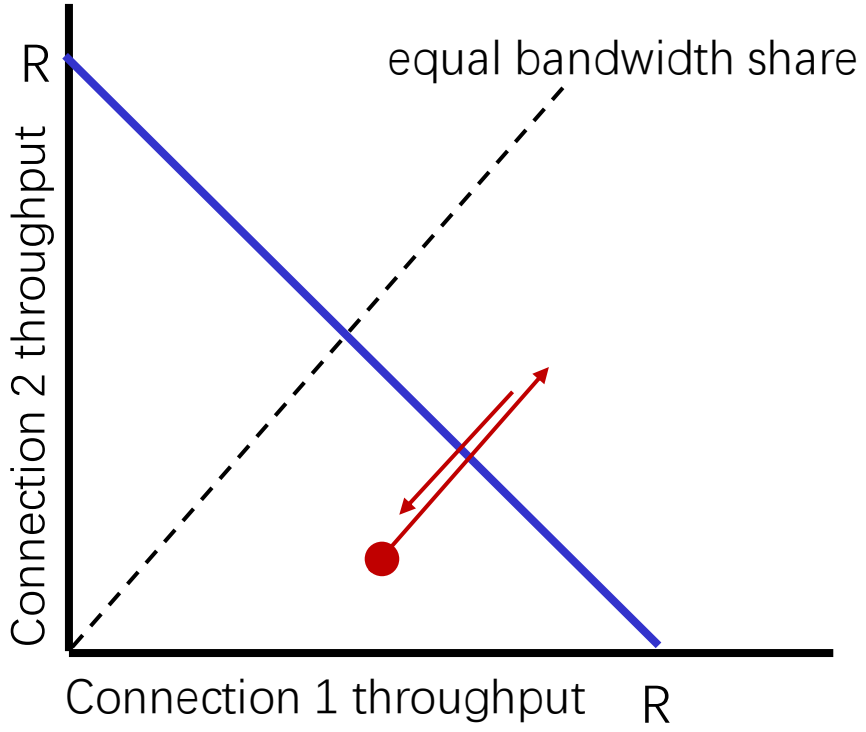
Fairness in TCP

- Consider a case with two TCP connections



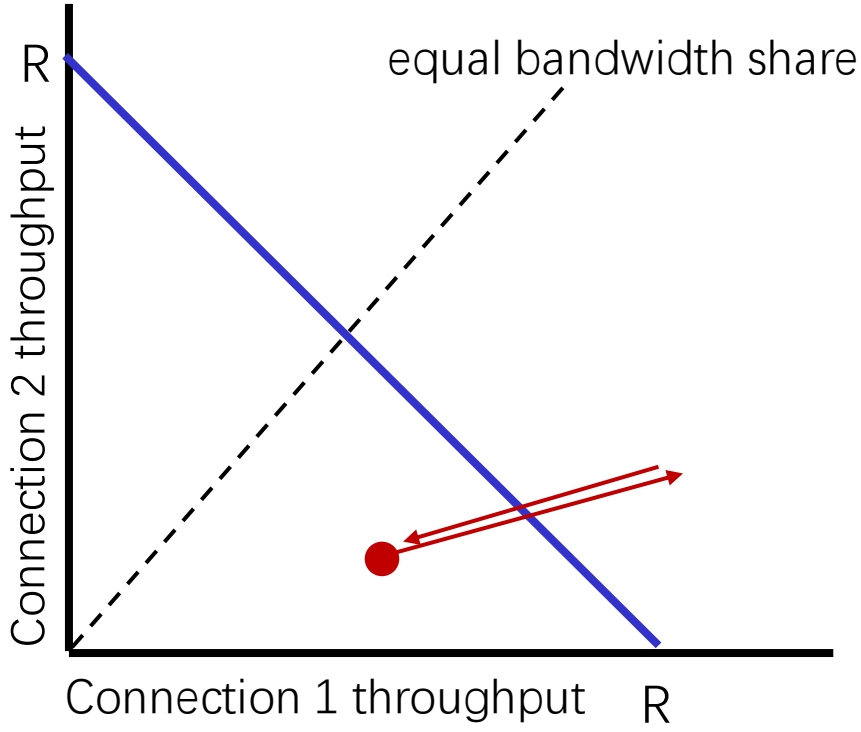
Fairness in TCP

- Consider a case with two TCP connections
  - Behavior of additive increase additive decrease
    - Stable but not fair



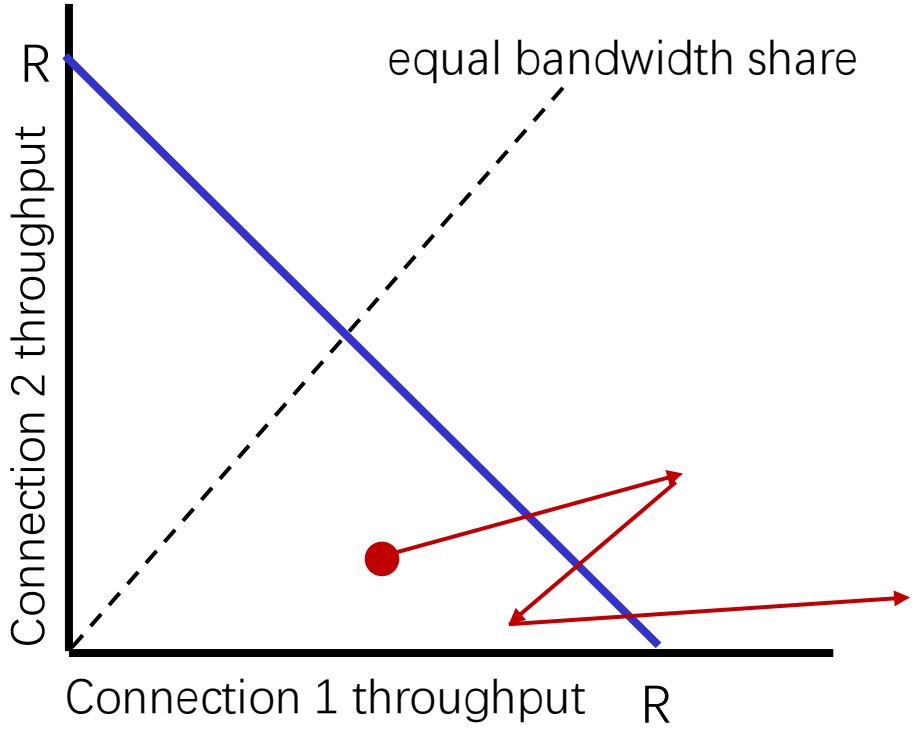
Fairness in TCP

- Consider a case with two TCP connections
  - Behavior of multiplicative increase multiplicative decrease
    - Stable but not fair



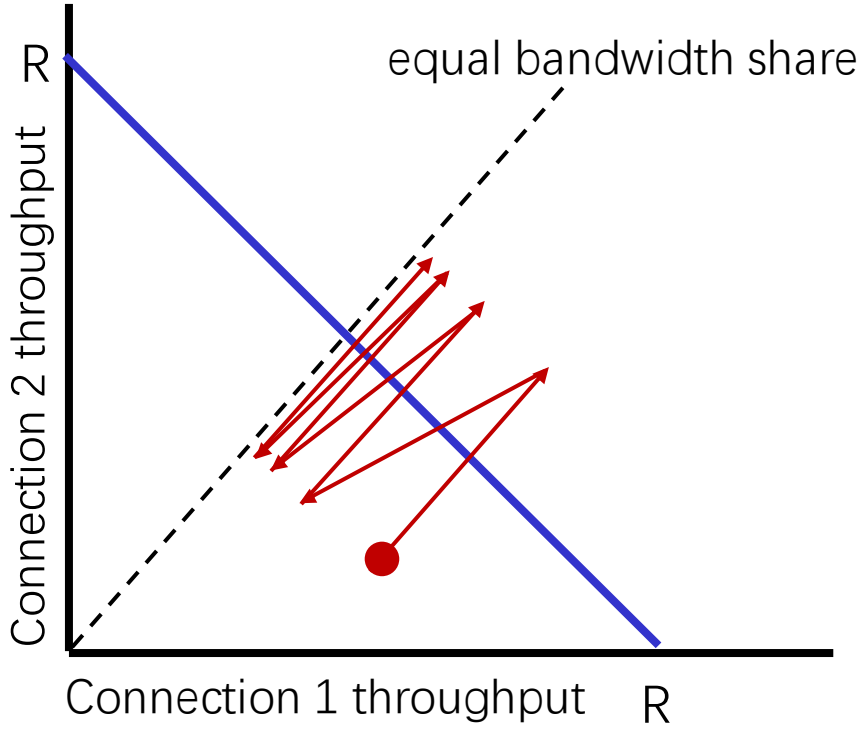
Fairness in TCP

- Consider a case with two TCP connections
  - Behavior of multiplicative increase additive decrease
    - Not stable



Fairness in TCP

- Consider a case with two TCP connections
  - Behavior of AIMD
    - Stable and fair



Fairness and RTT

- TCP connation with smaller RTT occupies more bandwidth
  - When congestion happens, they recover more quickly
    - TCP adjust cwnd in RTT basis

Fairness and Parallel TCP Connections

- Application can open multiple parallel connections between two hosts
  - web browsers do this , e.g., link of rate R with 9 existing connections:
    - new app asks for 1 TCP, gets rate R/10
    - new app asks for 11 TCPs, gets R/2

Fairness and UDP

- Some apps do not use TCP
  - do not want rate throttled by congestion control
- Instead, use UDP:
  - send audio/video at constant rate, tolerate packet loss
- There is no “Internet police” policing use of congestion control

Outline

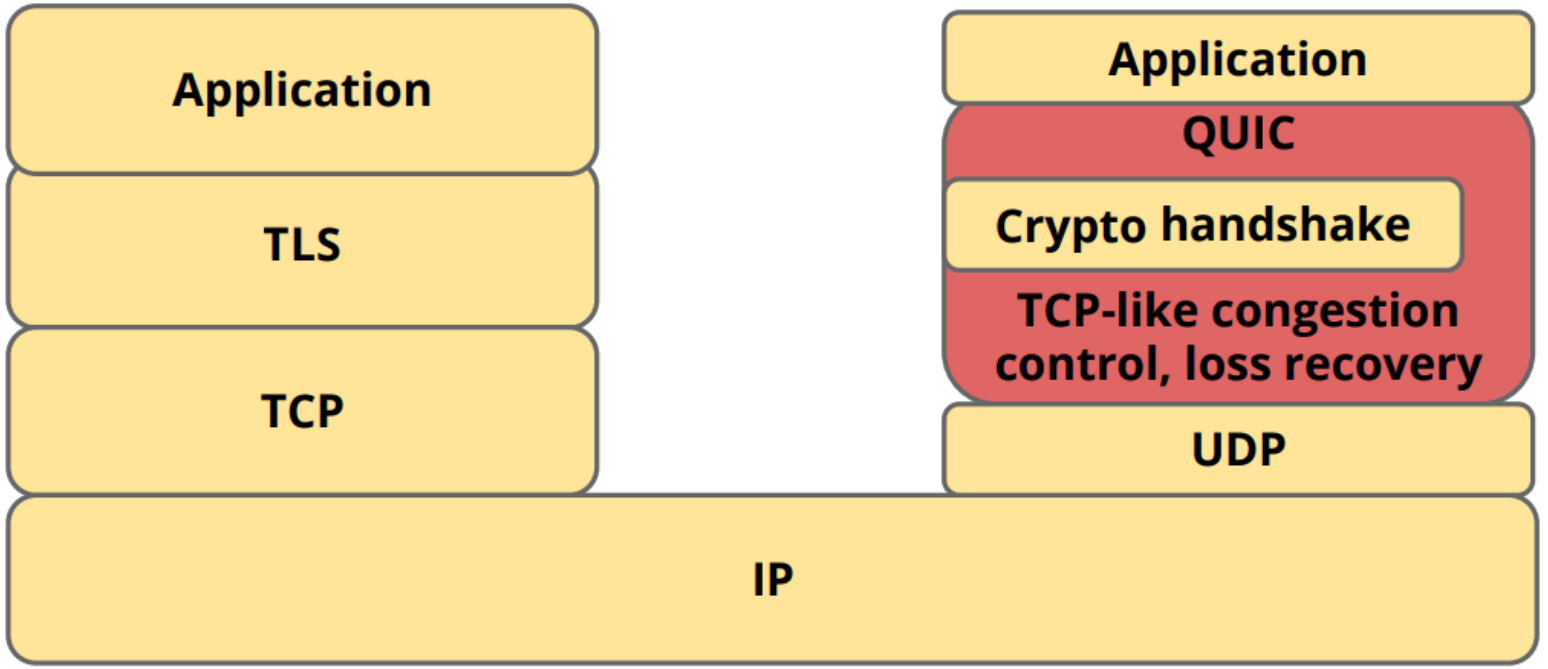
- TCP Fairness
  - ➢QUIC
- QoS

QUIC

- QUIC: Quick UDP Internet Connections
- Application-layer protocol, on top of UDP
  - Deployed by Google staring at 2014
    - Deployed on many Google servers, apps (Chrome, mobile YouTube app)
  - QUIC working group formed in Oct 2016
- Initial goal: increase performance of HTTP

QUIC

- Protocol Stack

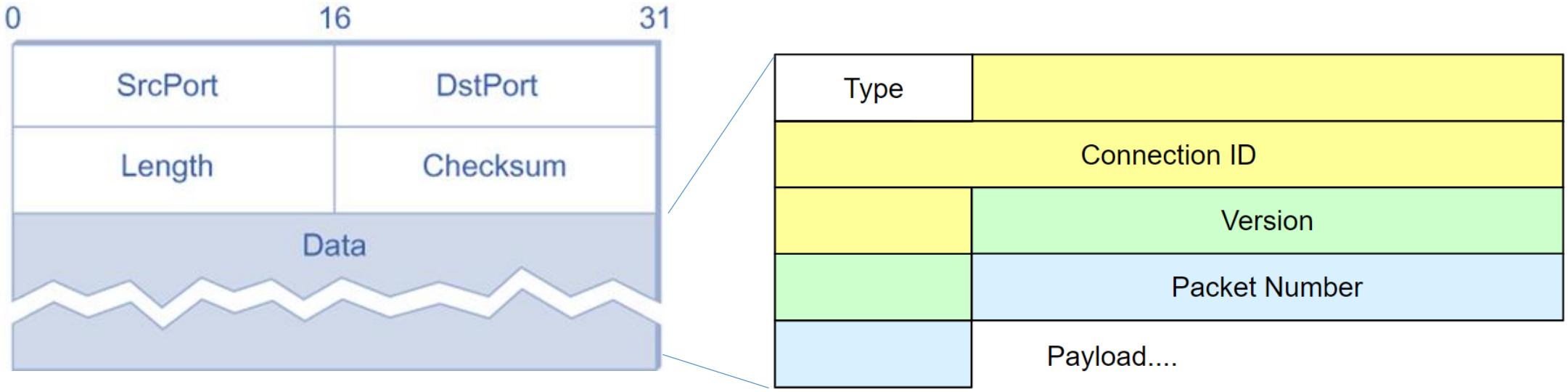


https://www.ietf.org/proceedings/98/slides/slides-98-edu-sessf-quic-tutorial-00.pdf

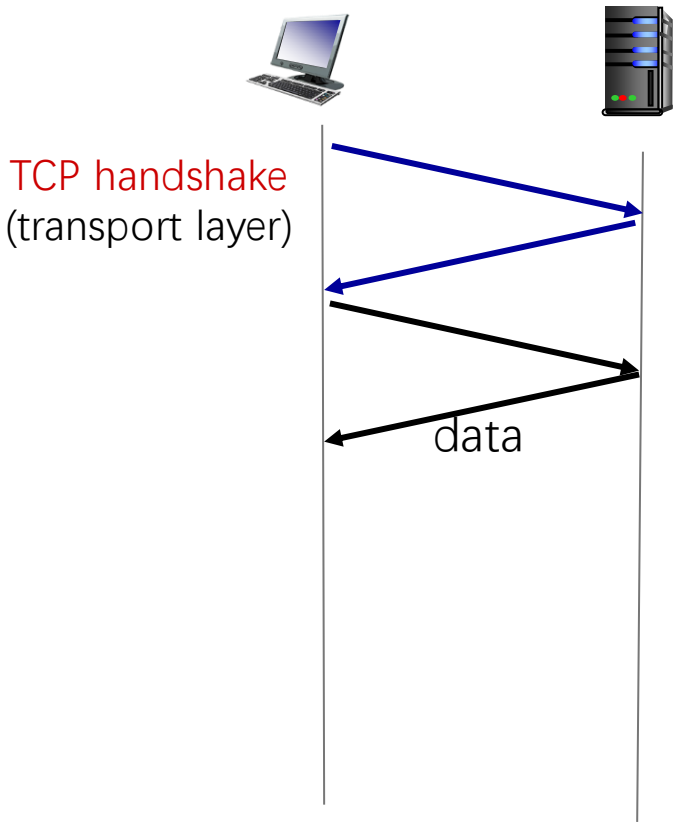
QUIC

- Key features
  - Always encrypted
  - 0-RTT connection establishment
  - Connection migration
  - Congestion control
  - Parallel Streams

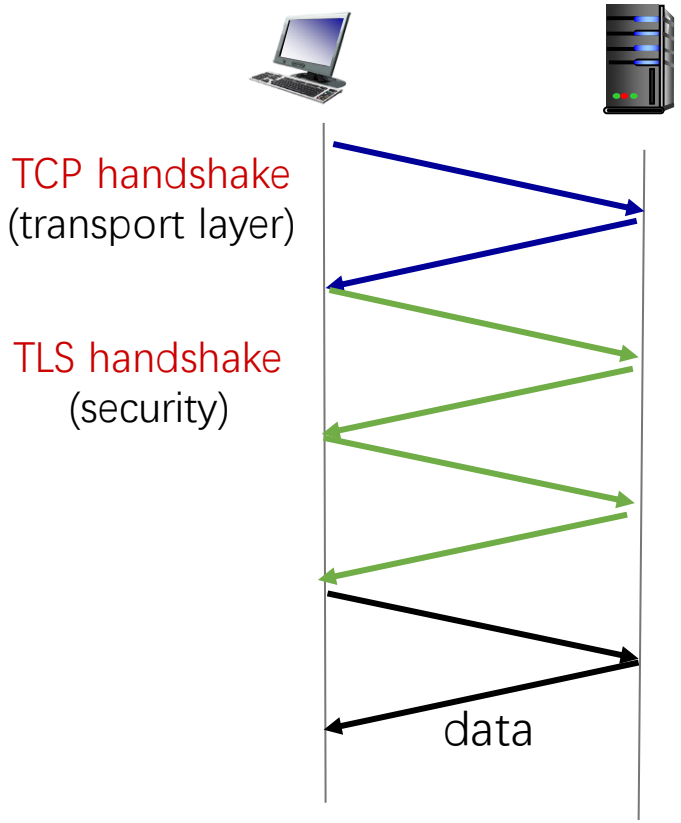
QUIC - Header


0
16
31
SrcPort
DstPort
Length
Checksum
Data
Type
Connection ID
Version
Packet Number
Payload....

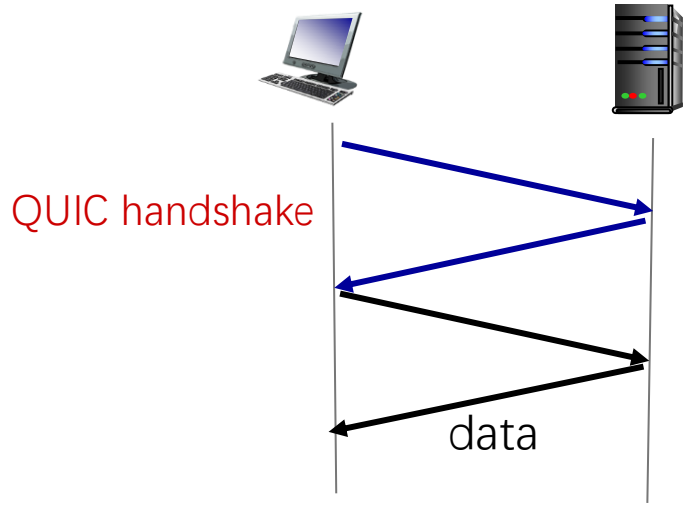
QUIC Connection Establishment



TCP
(2RTT)



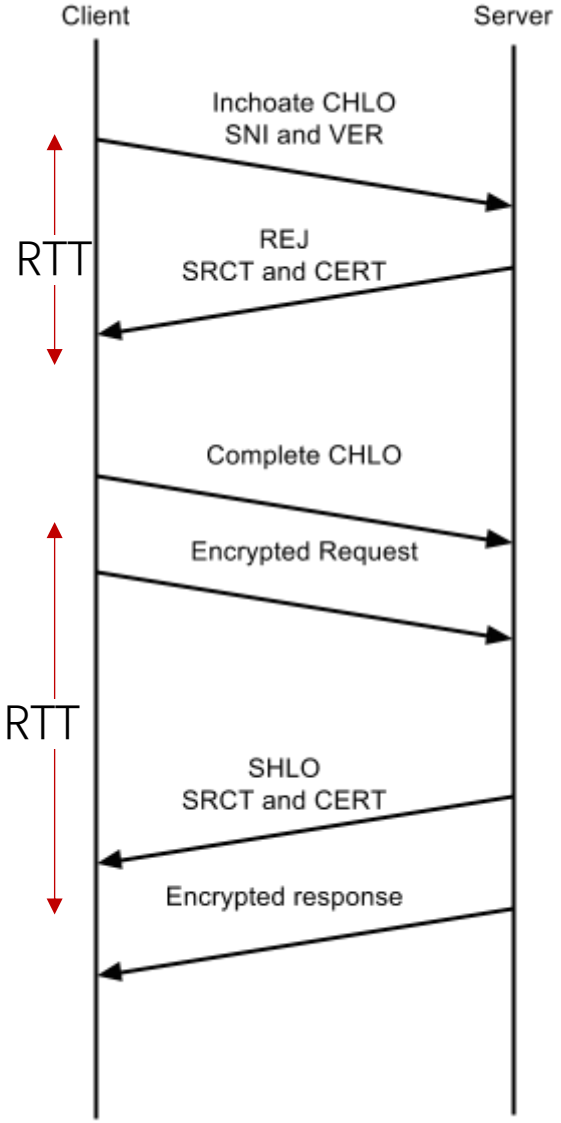
TCP+TLS 1.2
(new 4RTT
resumed 3RTT)



QUIC
(new 2RTT
Resumed 1RTT)

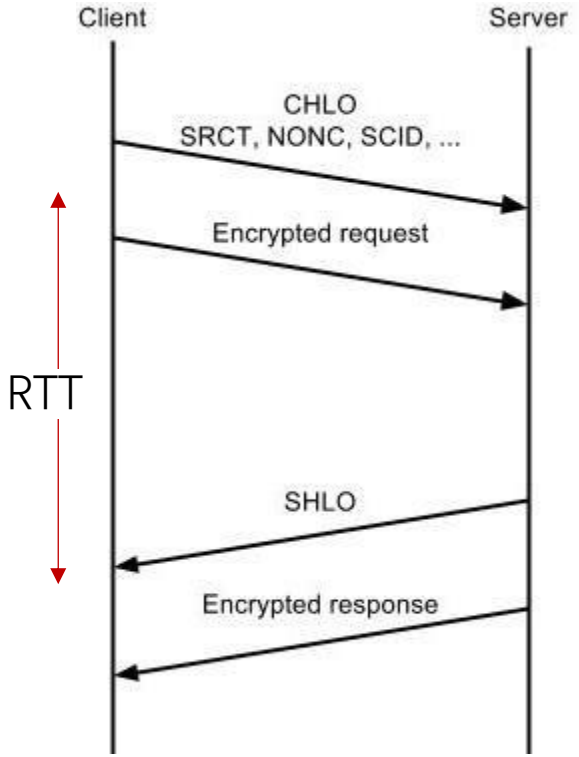
QUIC Connection Establishment

- 1-RTT (First-ever connection)
  - No cached information available
  - First CHLO is inchoate (empty)
    - Simply includes version and server name
  - Server responds with REJ
    - Includes server config, certs, etc.
    - Allows client to make forward progress
  - Second CHLO is complete
    - Followed by initially encrypted request data
  - Server responds with SHLO
    - Followed immediately by forward-secure encrypted response data



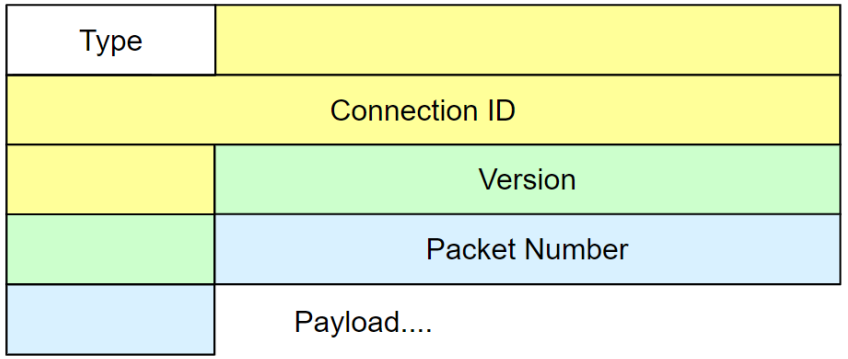
QUIC Connection Establishment

- 0-RTT (Subsequent connection)
  - Motivation: client can cache information about the *origin* it connected to
  - First CHLO is complete
    - Based on information from previous connection
    - Followed by initially encrypted data.
  - Server responds with SHLO
    - Followed immediately by forward-secure encrypted data



QUIC Connection Migration

- NAT Rebinding
  - NATs remaps port
    - Frequency (~ mins)
    - Why ? to release unused ports
      - According to TCP connection state (if they are closed)
    - UDP does not have connection state, QUIC state is encrypted
- Mobility
  - Switching between different IP
    - Wi-Fi and cellular network
- Connection Migration
  - Keep QUIC connections alive even if port and IP are chang
  - Detect connection path changes via Connection ID and IP/port
    - Connection is identified by connection ID rather than <IP, port>
    - 64-bit connection ID
    - randomly chosen by client



QUIC Congestion Control

- Incorporates TCP best practices
  - TCP Cubic, Fast Retransmission, Selective ACK, etc.
- Better signaling than TCP
  - Each packet carries a monotonically increasing packet number
    - Better RTT measurement
  - Retransmitted packets also consume new sequence numbers
    - no retransmission ambiguity
- More verbose ACK
  - support 256 ACK ranges (vs. TCP's 3 SACK ranges)

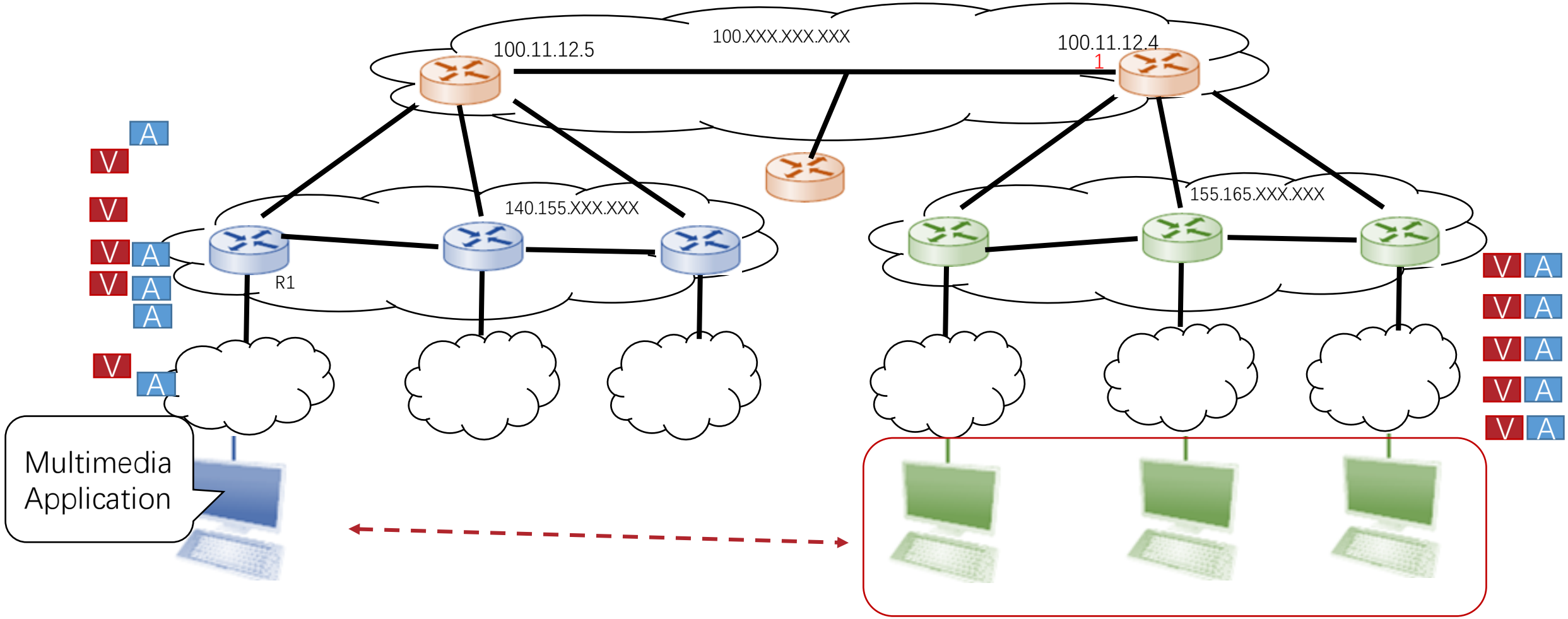
QUIC – Parallel Streams

- Handle HOL blocking

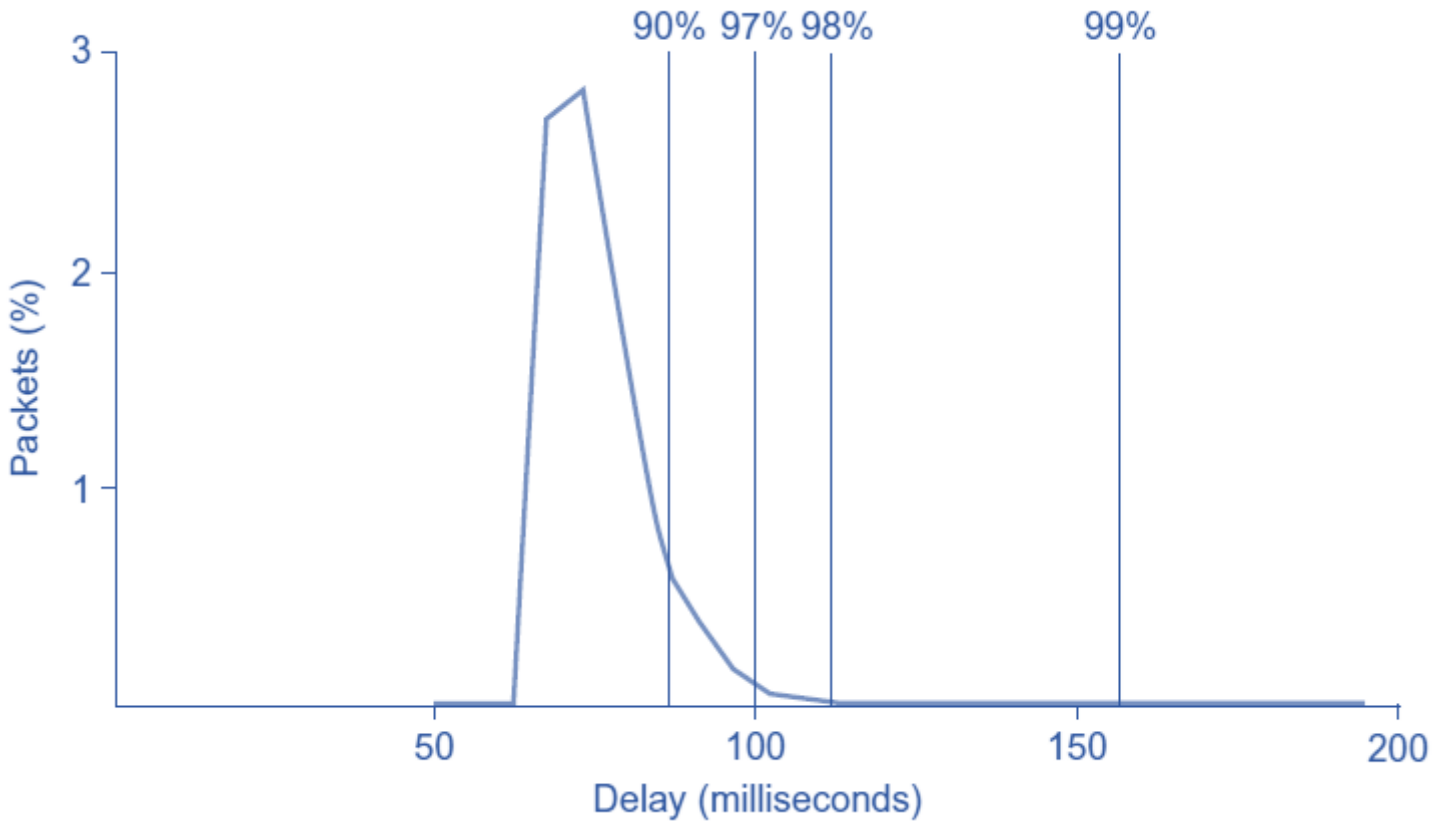
Outline

- TCP Fairness
- QUIC
- ➢QoS

Realtime Application

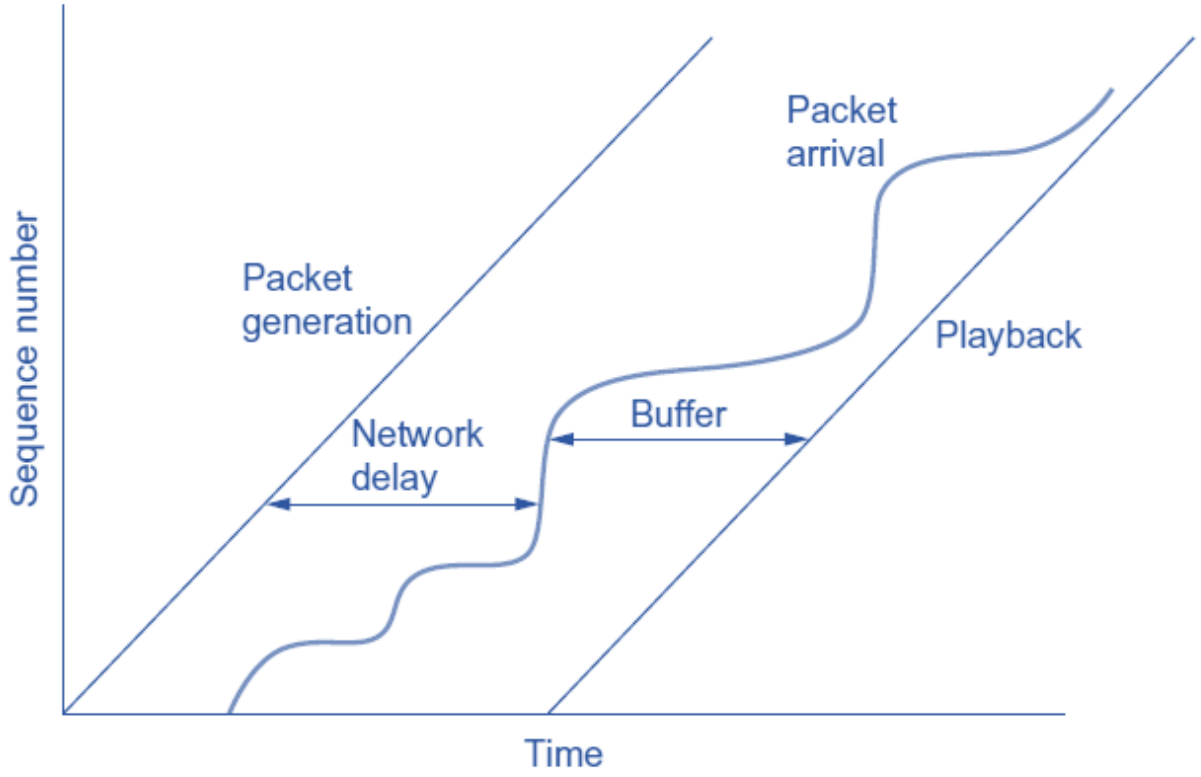


Delay Profile



Host Solution: Playback Buffer

- Buffer can be used to handle delay variance



Quality of Service (QoS)

- Objective: to provide different service (network quality) to different applications
- Service Model
  - Best effort
  - Integrated Services (IntServ)
    - QoS supports every individual applications/flows
  - Differentiated Services (DiffServ)
    - QoS supports multiple/two classes of data or aggregated traffic

Integrated Services (IntServ)

- Flow Specification
  - What is the flow
  - What we want to guarantee for the flow
- Admission Control
  - How network decides if it can accept the flow spec
- Resource Reservation Protocol
  - How service request gets from host to network
- Packet Classification and Scheduling
  - How routers deliver service

Flow Specification

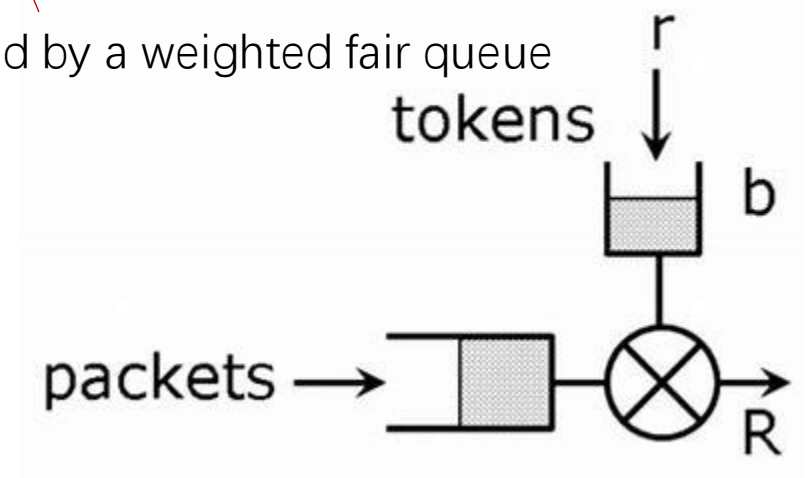
There are multiple options

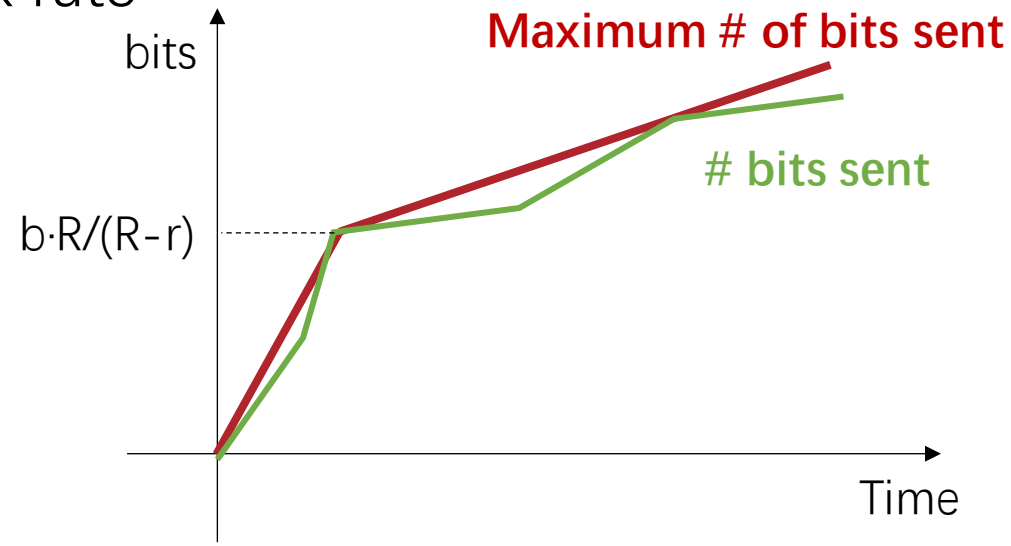
- Specify the maximum bit rate
  - Maximum bit rate may be much higher than average
  - Reserving for the worst case is wasteful
- Specify the average bit rate
  - Network will not be able to carry bursty traffic
- **Specify the burstiness of the traffic**
  - Specify both the average rate and the burst size

Specify Burstiness: Token Bucket

- Token Bucket: limit input to specified burst size and average rate
- Parameters:
  - r: average rate, i.e., rate at which tokens fill the bucket
  - b: bucket depth (limits size of burst)
  - R: maximum link capacity or peak rate

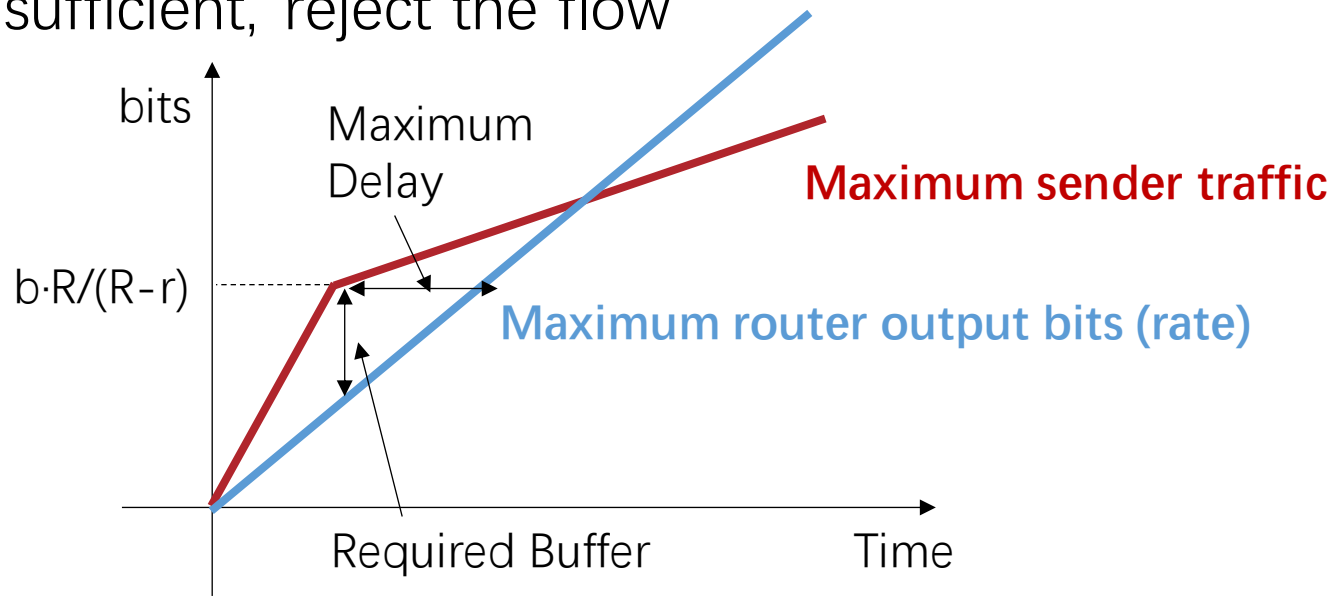
Can be limited by a weighted fair queue





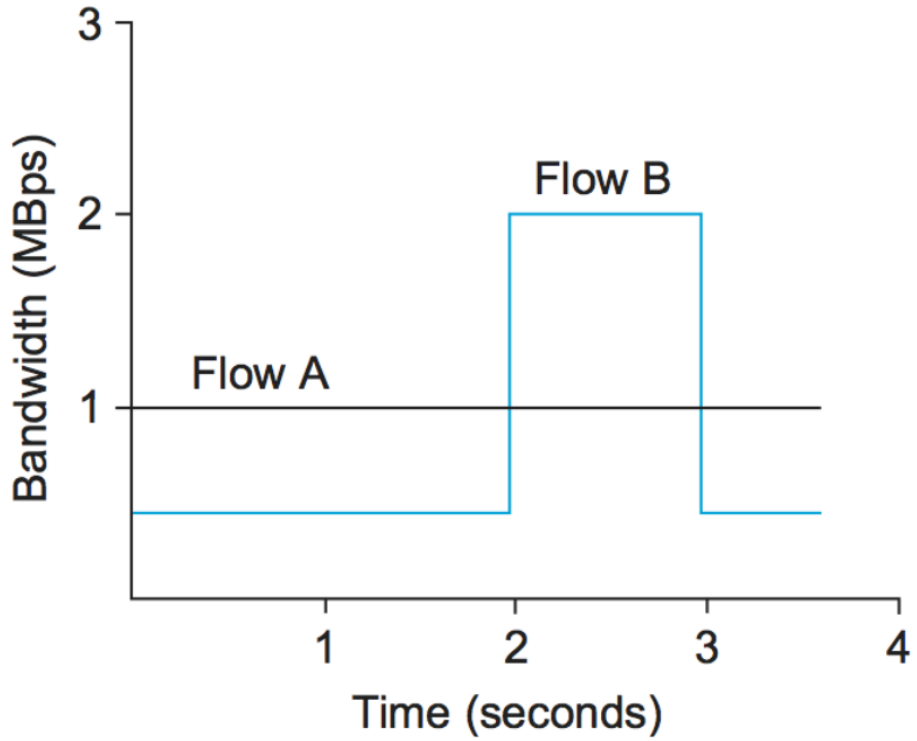
Specify Burstiness: Token Bucket

- Host
  - Specify token bucket to describe its traffic
- Router
  - Allocate buffer and bandwidth to guarantee delay
  - If resource is insufficient, reject the flow

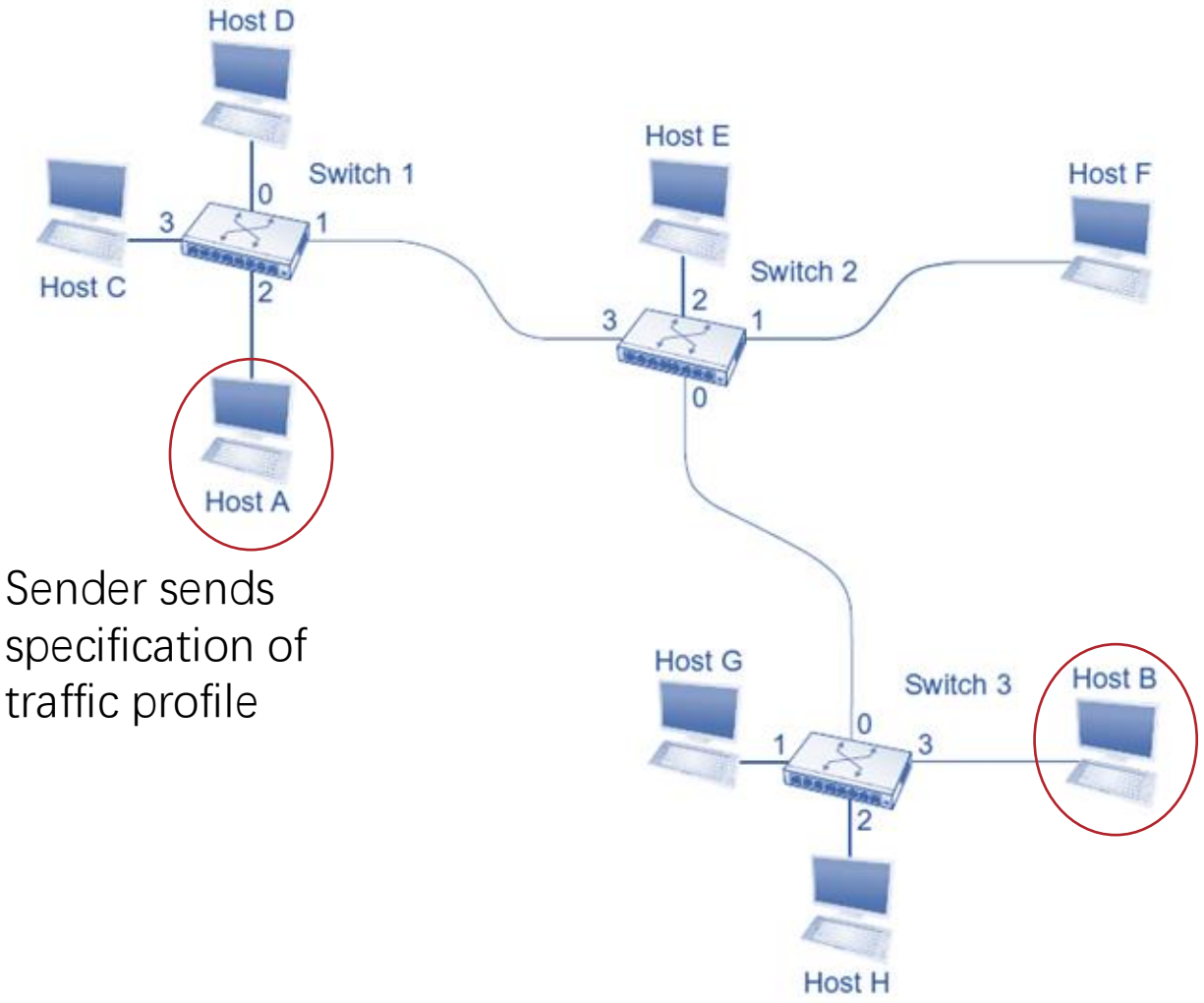


Token Bucket

A flow can be described by multiple token bucket.

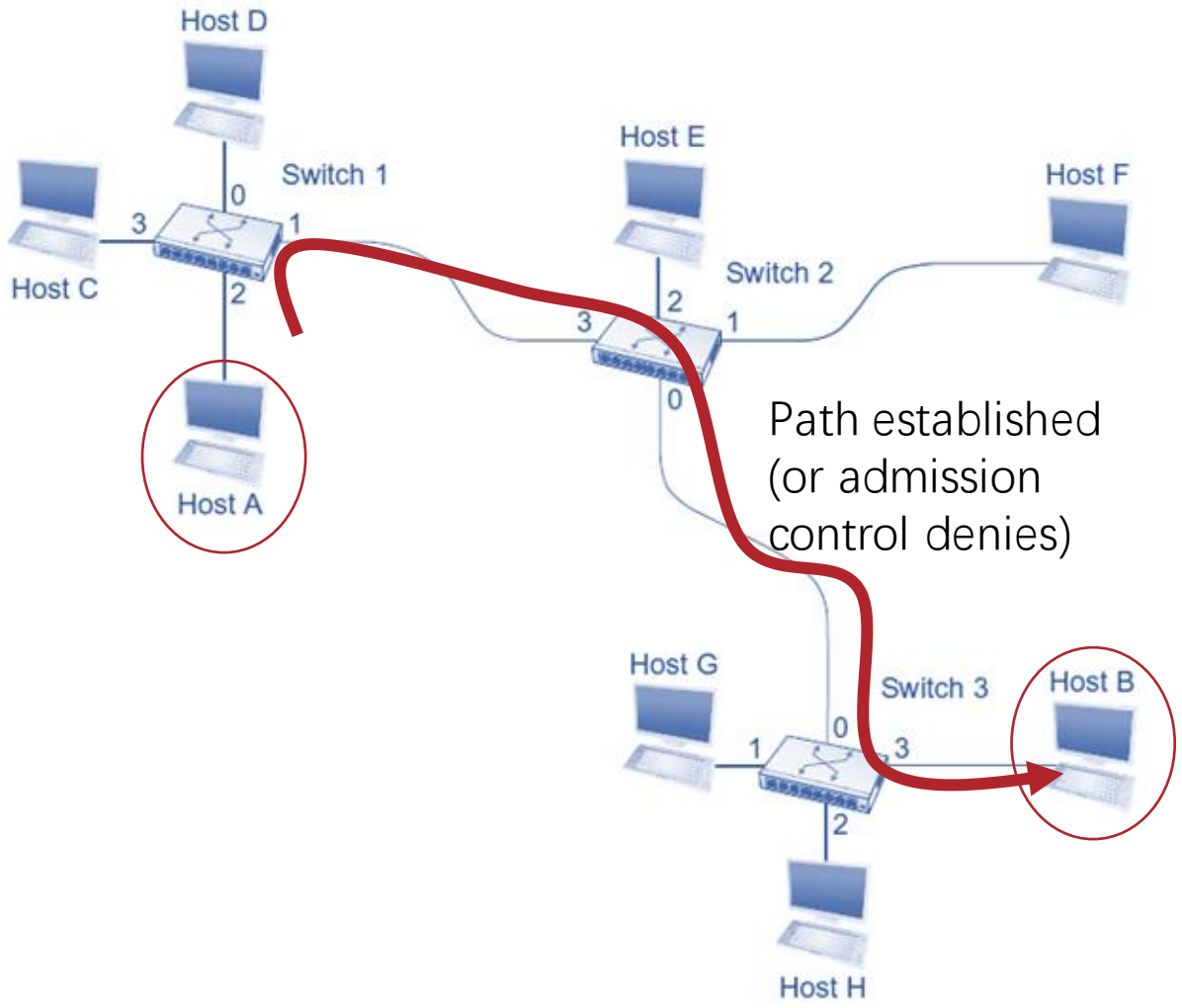


Resource Reservation Protocol

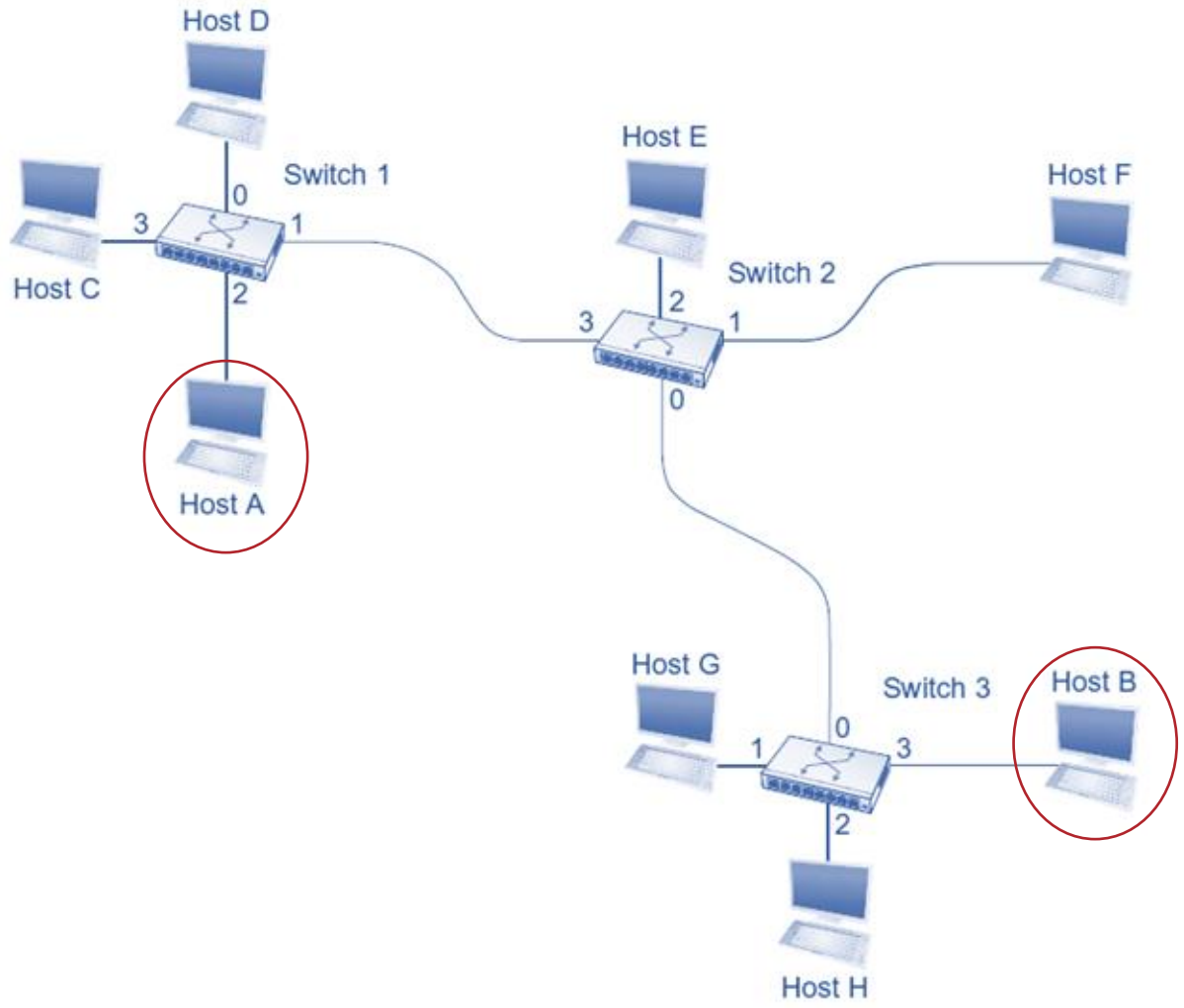


Sender sends specification of traffic profile

Resource Reservation Protocol

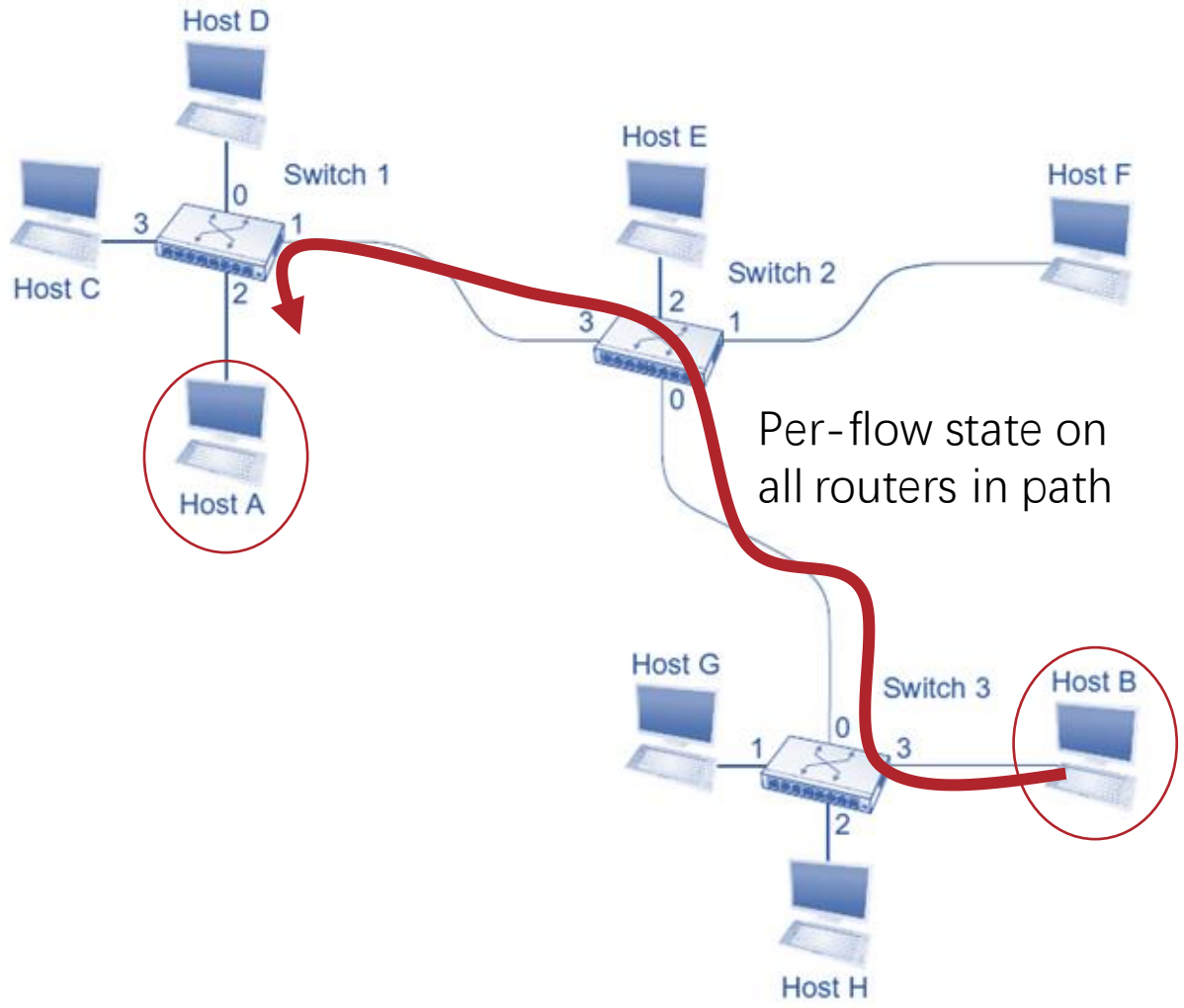


Resource Reservation Protocol

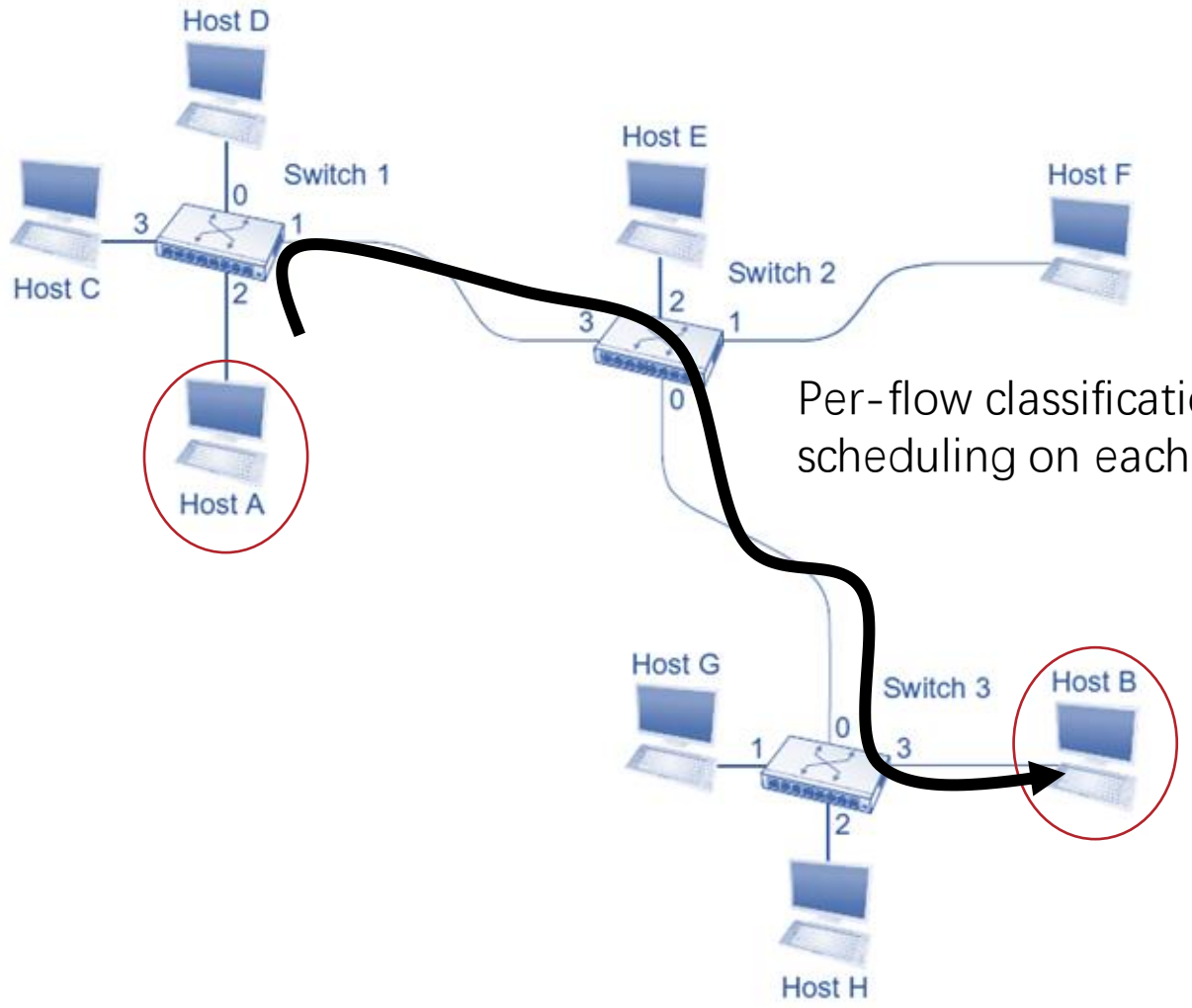


The receiver signals reservation request

Resource Reservation Protocol



Resource Reservation Protocol

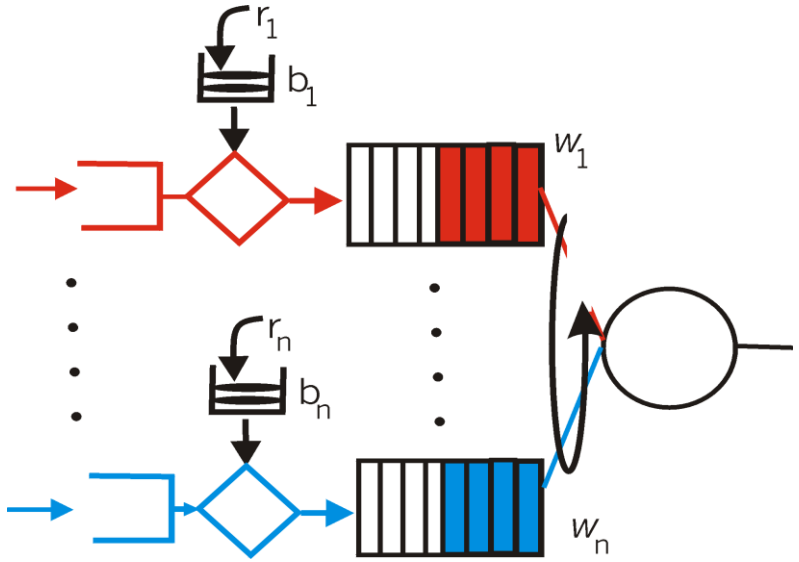


Packet Classification

- Classify Packets into Flows according to
  - Source Address
  - Destination Address
  - Protocol Number
  - Source Port
  - Destination Port

Packet Scheduling

- Implementation Dependent
  - Token bucket + Fair Queue



Scalability Issues

- Specify service for every flow is not scalable in Internet
  - Routers must keep the state of every passing flow

Quality of Service (QoS)

- Objective: to provide different service (network quality) to different applications
- Service Model
  - Best effort
  - Integrated Services (IntServ)
    - QoS supports every individual applications/flows
  - ➢ Differentiated Services (DiffServ)
    - QoS supports multiple/two classes of data or aggregated traffic

Differentiated Services (DiffServ)

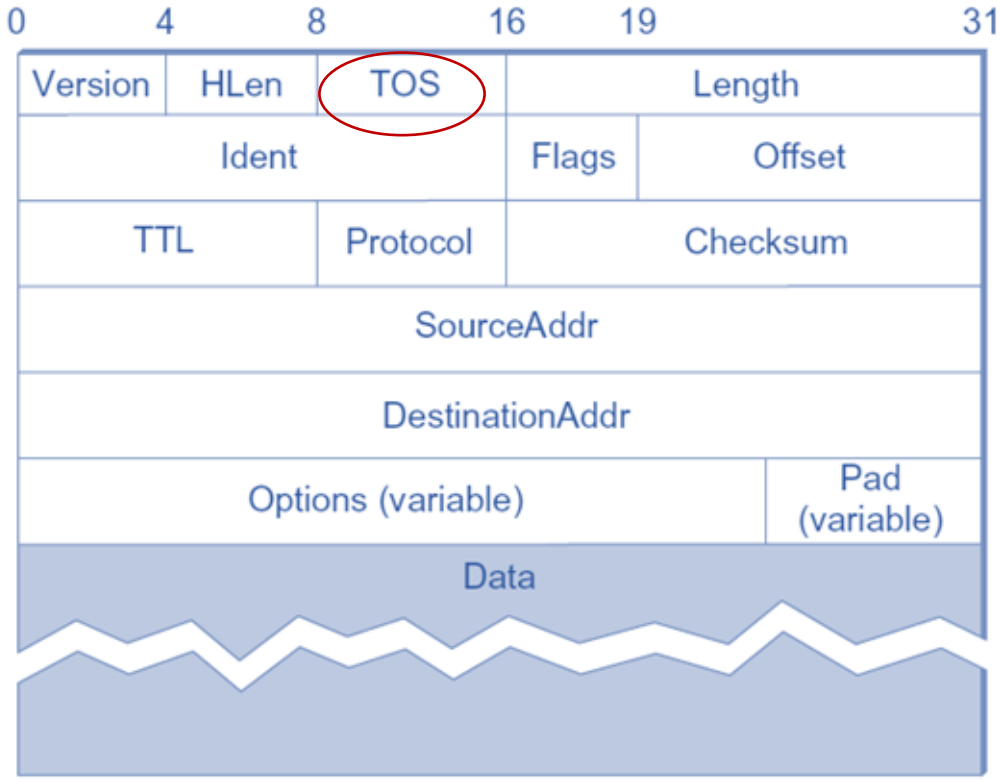
- Problem with IntServ: scalability
  - Maintain per-flow state
  - Per-flow classification
- DiffServ Approach
  - Segregate packets into a small number of (two) classes
    - Premium
    - Other
  - Class of certain packet (state) is kept in packet header
    - ToS

Differentiated Services (DiffServ)

- Edge Routers
  - Set the traffic class
    - ToS field => token bucket parameters
  - Schedule traffic according to the traffic class
- Core Routers
  - Schedule traffic according to the traffic class specified by the edge router

Per Hop Behavior

- Reuse ToS Field
  - 0-5bit: Differentiated Service Code Point (DSCP) Field
  - 6-7bit: Explicit Congestion Notification
- DSCP field encodes Per-Hop Behavior
  - Expedited Forwarding (all packets receive minimal delay & loss)
  - Assured Forwarding (packets marked with low/high drop probabilities)



Set Packet Class

- DSCP Field in Practice
  - Edge Routers
    - Set Differentiated Service (DS) Field in IP header
      - Maybe because the user paid the ISP
  - Core Routers
    - Implement Per Hop Behavior
      - According to DS Field of packets

Commonly used DSCP values

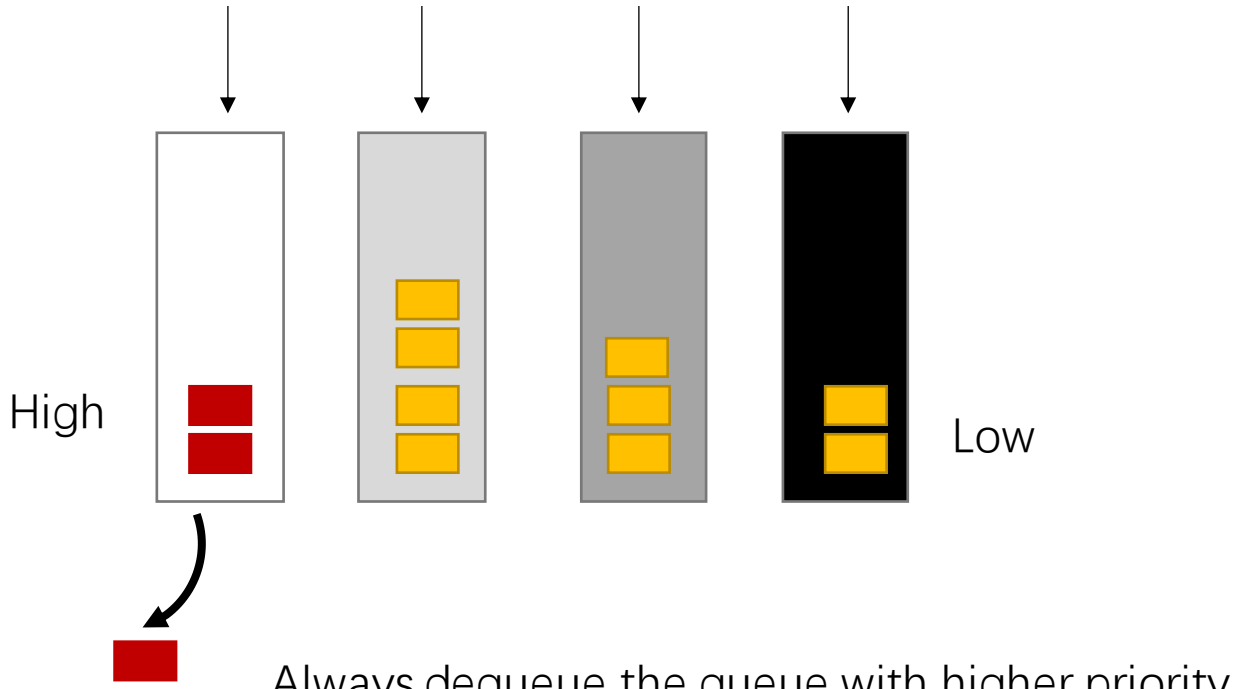
DSCP value	Hex value	Decimal value	Meaning	Drop probability	Equivalent IP precedence value
101 110	0x2e	46	Expedited forwarding (EF)	N/A	101 Critical
000 000	0x00	0	Best effort	N/A	000 - Routine
001 010	0x0a	10	AF11	Low	001 - Priority
001 100	0x0c	12	AF12	Medium	001 - Priority
001 110	0x0e	14	AF13	High	001 - Priority
010 010	0x12	18	AF21	Low	010 - Immediate
010 100	0x14	20	AF22	Medium	010 - Immediate
010 110	0x16	22	AF23	High	010 - Immediate
011 010	0x1a	26	AF31	Low	011 - Flash
011 100	0x1c	28	AF32	Medium	011 - Flash
011 110	0x1e	30	AF33	High	011 - Flash
100 010	0x22	34	AF41	Low	100 - Flash override
100 100	0x24	36	AF42	Medium	100 - Flash override
100 110	0x26	38	AF43	High	100 - Flash override

Implementation of Per-Hop Behavior

- Expedited Forwarding (EF) PHB
  - Highest Priority
- Assured Forwarding (AF) PHB
  - Different levels of priorities, drop probabilities, bandwidth, etc.

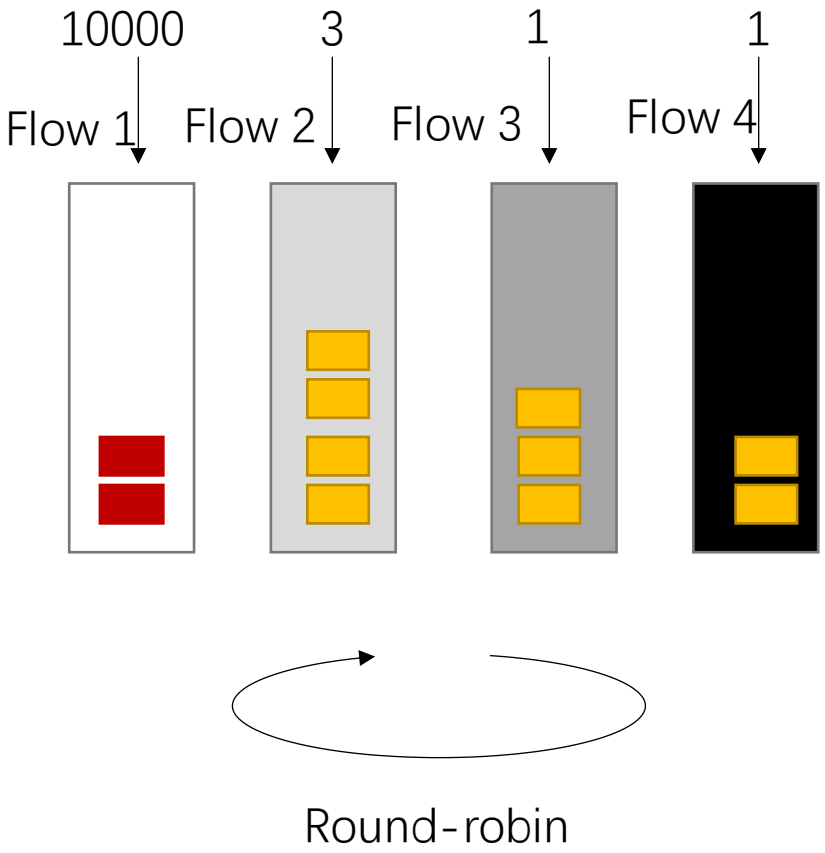
Implementation of Expedited Forwarding

- First-In-First-Out (FIFO) with Priority



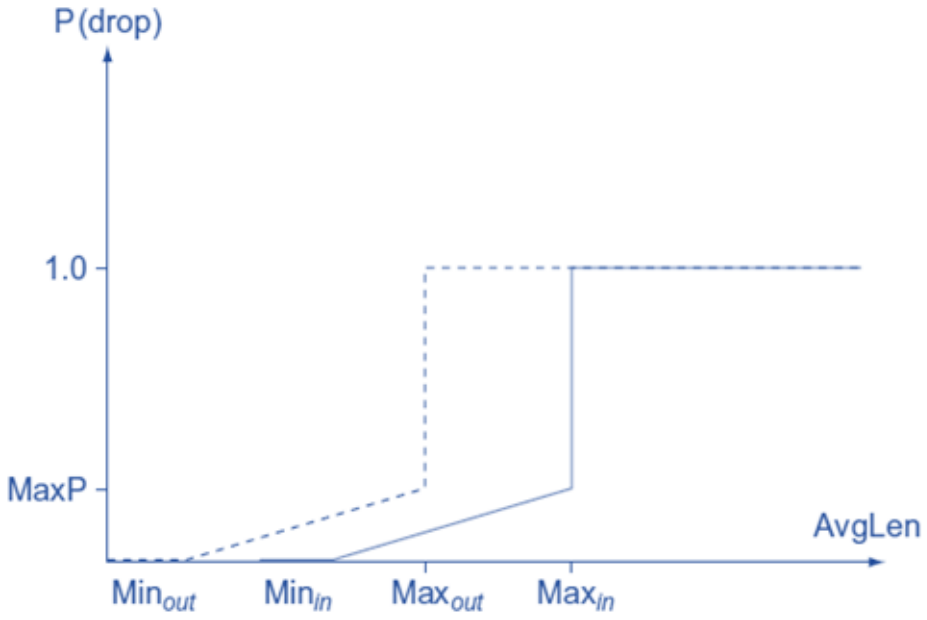
Implementation of Expedited Forwarding

- Weighted Fair Queuing (FQ)



Implementation of Assured Forwarding

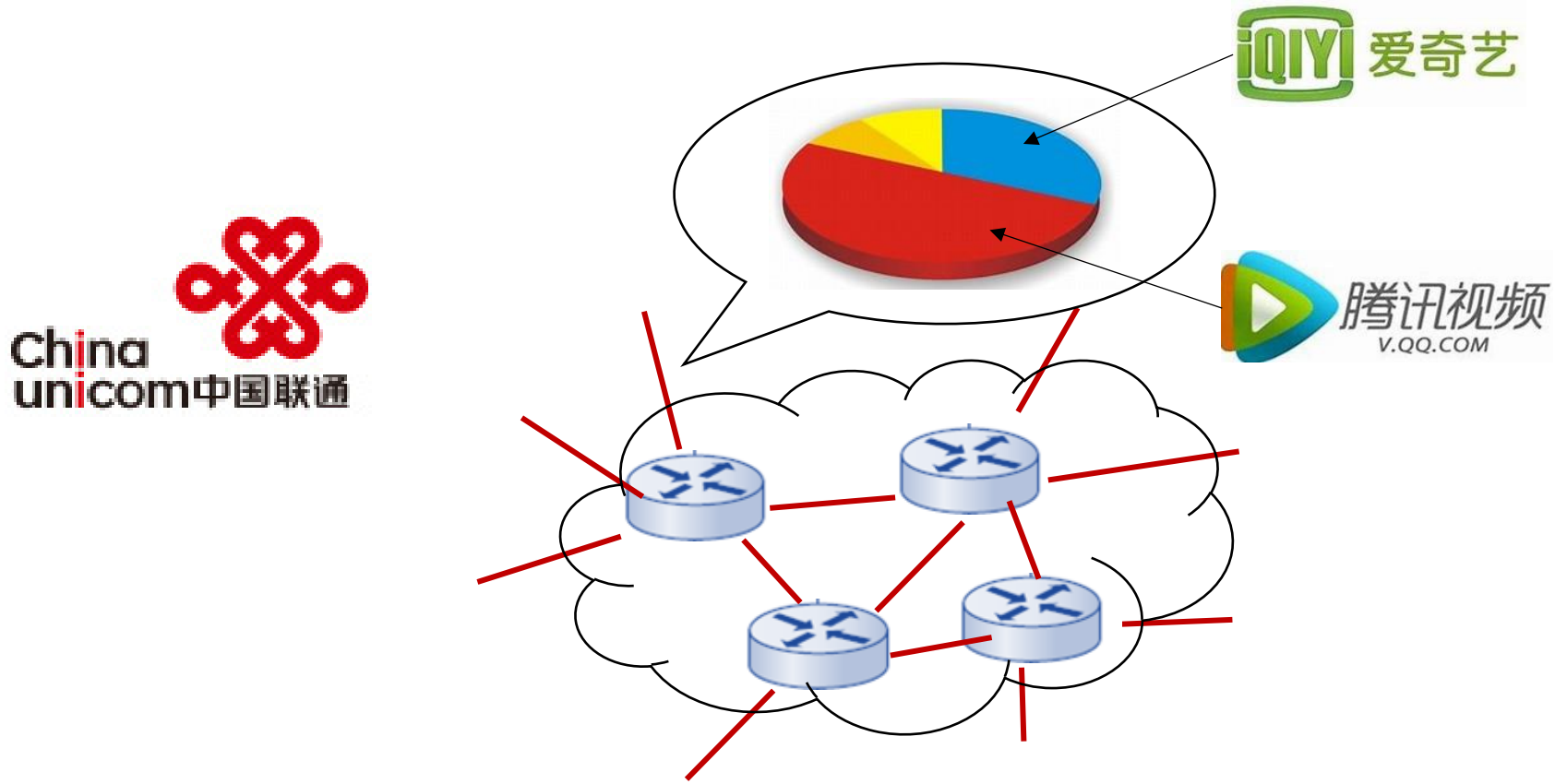
- RED with In and Out (RIO)



月基本费		258元 188元
包含	国内流量	1GB
	国内通话	800分钟
	本地流量	本地流量无限量权益 （用满40GB后限速）

Network Neutrality

- Network Neutrality
  - ISPs supply non-discriminated IP connectivity



Network Neutrality

- Opposite Counterpoint
  - ISPs only allows you to access their (often value-added) services

China unicom中国联通

Reference

- Textbook 6.5
- Some slides are adapted from http://www-net.cs.umass.edu/kurose_ross/ppt.htm by Kurose Ross
- https://www.chromium.org/quic