The Network Layer: Quality of Service

CS 352, Lecture 18, Spring 2020

http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

RUTGERS UNIVERSITY | NEW BRUNSWICK

Course announcements

- Quiz 6 due on Tuesday at 10 PM
  - Covers lectures 17 and 18
- Project 2 due later today
- Project 3 will go out this weekend
  - Start early

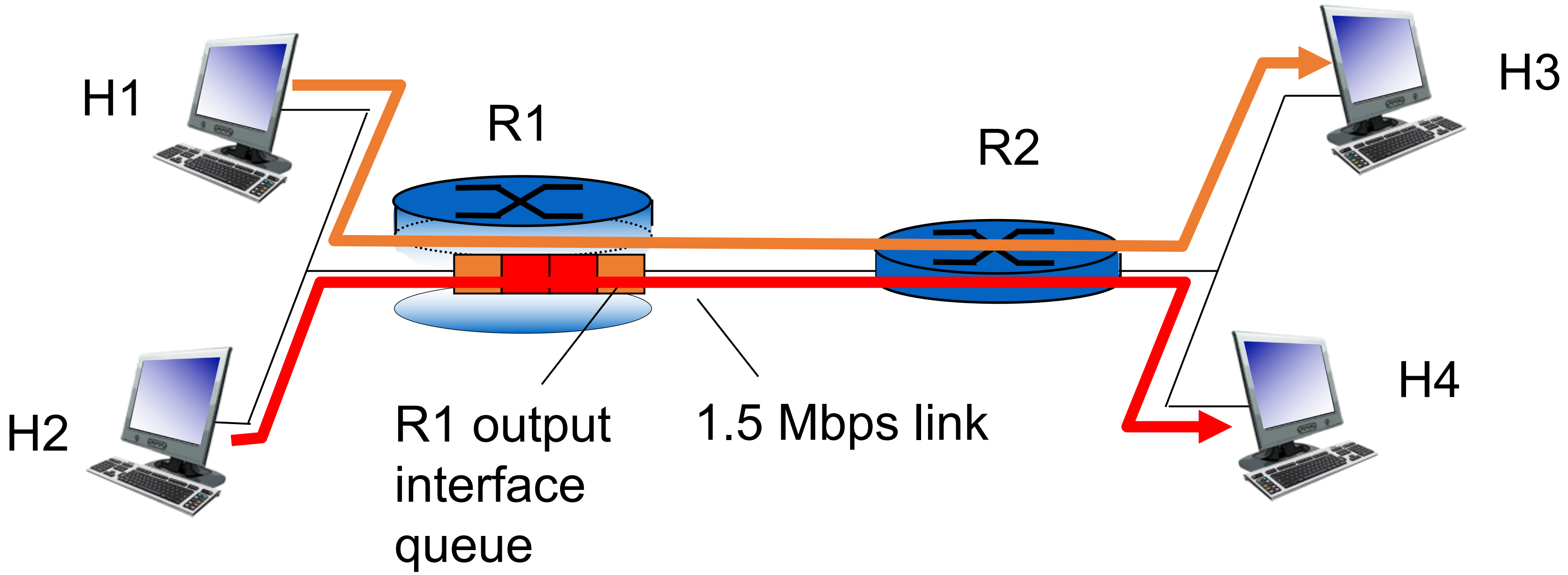
Review of concepts

- Border Gateway Protocol: a path-vector protocol
  - Announce paths to specific destinations through prefix + attributes
    - AS Path, next hop
- Business relationships influence route export and selection
- BGP paths get propagated inside ASes using iBGP
  - Used together with intra-domain routing to compute forwarding table
  - Two tables used together (BGP next hop, intra-domain lookup)
- Intra- and inter-AS routing protocols address different concerns
  - Policy vs. performance
- Quality of service within the network
  - … when best-effort isn't enough for apps (e.g., robotic surgery over the net)
  - Traffic identified using packet marking; now how to arbitrate traffic?

Quality of Service

How can the network make application performance better?

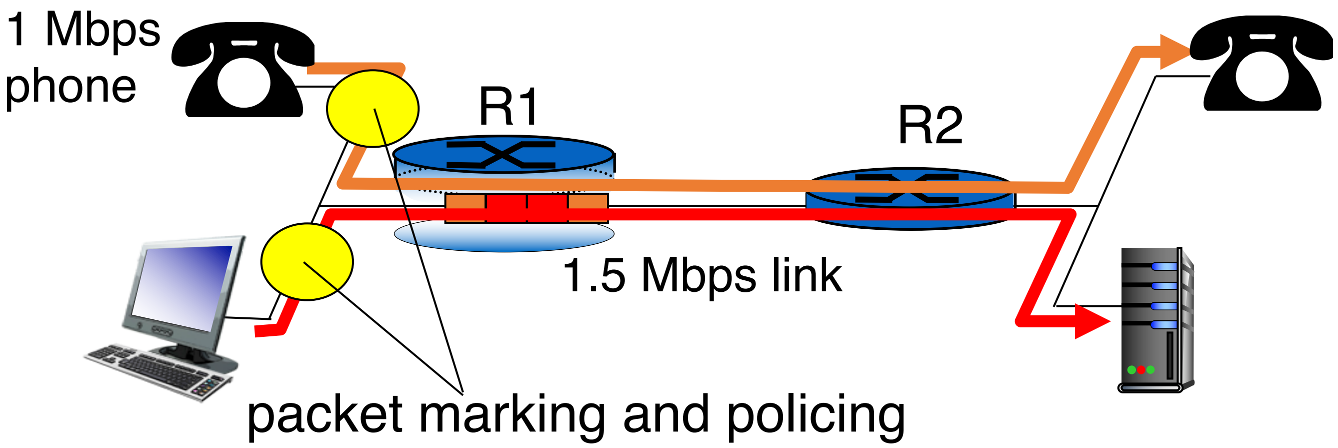
Multiple classes of service: scenario



- example: 1Mbps VoIP, HTTP share 1.5 Mbps link.
  - HTTP bursts can congest router, cause audio loss
  - want to give priority to audio over HTTP

Principles for QOS guarantees

- what if applications misbehave (VoIP sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations
- *marking*, *policing* at network edge

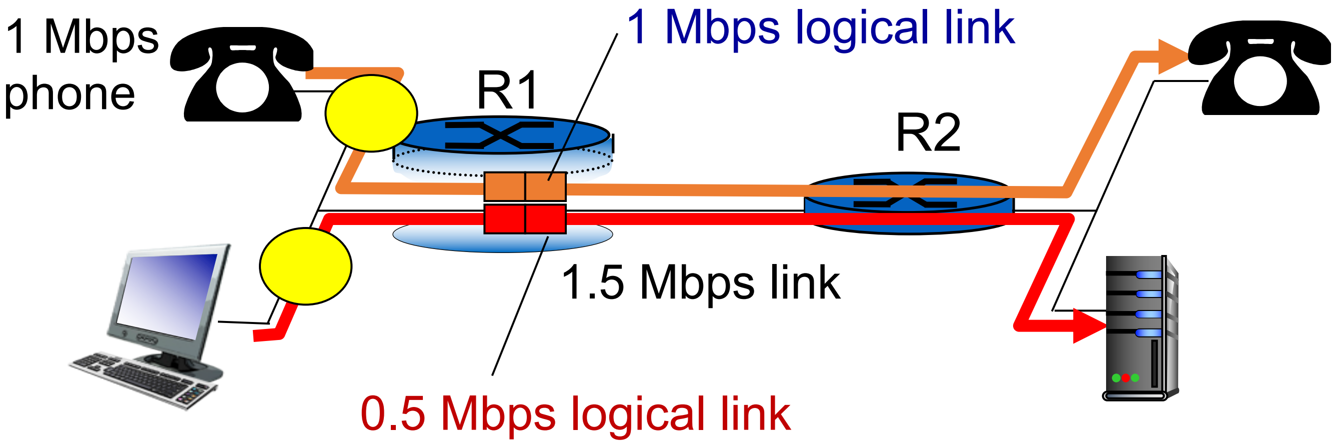


Principle 1: Isolation

Provide protection for one class from others

Principles for QoS guarantees

- allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flows doesn't use its allocation



Principle 2: Work conservation

While providing isolation, use resources as efficiently as possible

Poll #1

- Where does contention between different traffic classes (e.g., resulting in long queues) typically occur within routers?
  - (a) switch fabric
  - (b) input line termination
  - (c) output port buffers
  - (d) forwarding table

Poll #2

- What router mechanisms might you use to implement quality of service mechanisms?
  - (a) forwarding
  - (b) scheduling
  - (c) buffer management
  - (d) switching

Packet scheduling for QoS

Shaping and policing

Why care about packet scheduling?

- Significantly influences how packets are treated regardless of the endpoint transport
  - Implementations of Quality of Service (QoS) within large networks
  - Implications for net neutrality debates
- Intellectually interesting, foundational problem that a network solves
  - Classic Demers et al paper on scheduling (WFQ) has ~ 1500 citations
  - Important connections to other literature (e.g., job scheduling)
- Scheduling algorithms influence many daily life decisions ☺

Scheduling vs. Buffer Management

- How packets enter vs. how packets leave the switch buffer
  - Common architecture: shared-buffer switches, where buffer memory is shared between different output ports
  - Typical buffer management strategy: tail-drop

- How should buffer memory be partitioned across ports?
- Static partitioning?
  - Inefficient: even if port 1 has nothing to send, might drop port 2
- Also want fair sharing of buffer
  - If output port 1 is congested, why should port 2 traffic suffer?

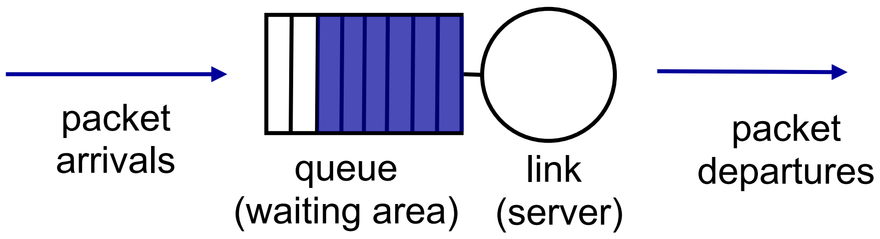
- State of the art: demand-aware buffer sharing algos

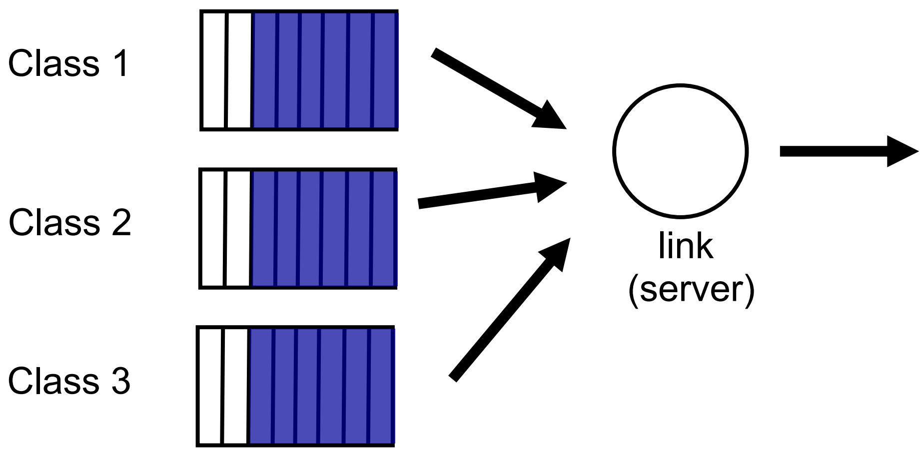
Packet scheduling for QoS

- Choose next queued packet to send on outgoing link

- Common scheduling disciplines
  - first come first served (FIFO)
  - simple multi-class priority
  - round robin
  - weighted fair queueing (WFQ)
  - Rate limiting

- All except rate-limiting are work-conserving





Poll #3

- When cars from two lanes on a congested freeway take turns to merge into one lane, what scheduling discipline is being implemented?
  - (a) priority-queueing
  - (b) weighted fair queueing
  - (c) rate limiting
  - (d) none of the above

Isolation by Rate Limits

Providing Isolation through Rate Limiting

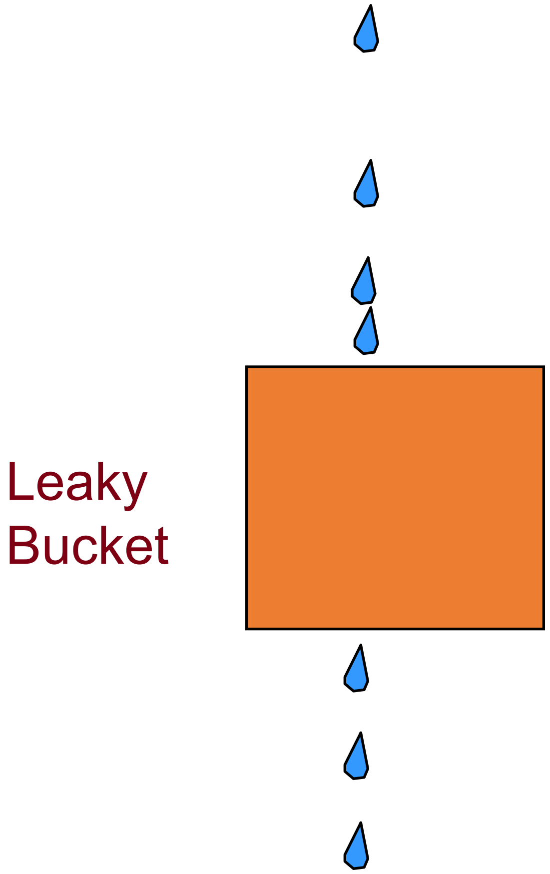
Three commonly used terms:

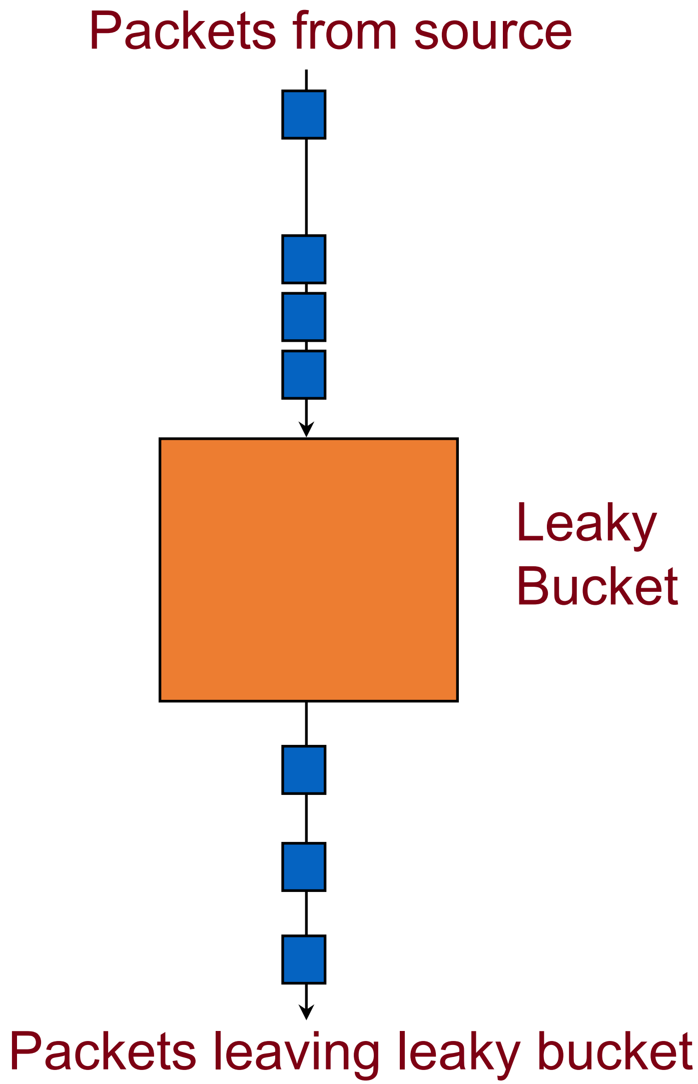
- *(long term) average rate:* how many pkts can be sent per unit time (in the long run)
  - crucial question: what is the interval length? 100 packets per sec or 6000 packets per min have same average! Instantaneous behaviors of the two may look very different.
- *peak rate:* e.g., 6000 pkts per min (ppm) avg.; 1500 ppm peak rate
- *(max.) burst size:* max number of pkts sent consecutively (with no intervening idle)

Shaping and Policing

Policing	Enforces rate by *dropping* excess packets immediately – Can result in high loss rates + Does not require memory buffer + No RTT inflation
Shaping	Enforces rate by *queueing* excess packets + Only drops packets when buffer is full – Requires memory to buffer packets – Can inflate RTTs due to high queueing delay

Mechanism (1): Leaky Bucket Shaper





Mechanism (1): Leaky Bucket Shaper

- Packets may be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced
  - The “bucket” buffers packets up to a certain point
  - If the bucket is full, packets are dropped.
- May be used in conjunction with resource reservation to police the host’s resource use
  - E.g., at the host-network interface, allow packets into the network at a constant rate
- May be used in the core of a network to limit bandwidth use

Shaping traffic with leaky buckets

- The leaky bucket is a traffic shaper: It changes the downstream arrival times & characteristics of packets passing through it
- Traffic shaping makes traffic more manageable and more predictable downstream (e.g., always under a certain rate)
- Usually, a system/network administrator would set the rate at which packets may be sent through the leaky bucket
- Administrator also sets up policies to map any connection that started up to a leaky bucket (and rate) of its own

Issues with a leaky bucket

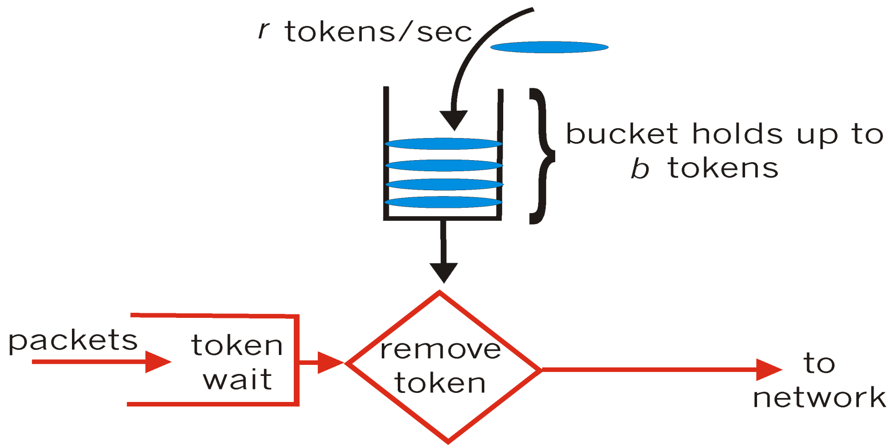
- Many short transfers just have a few packets
  - E.g., web requests and responses
  - Enforcing rate limit for those can significantly delay completion
- For a leaky bucket shaper, average rate == peak rate
- Sometimes, we wish to have peak rate higher than avg rate
- For this purpose we use a token bucket, which is a burst-tolerant version of a leaky bucket

Token buckets

Mechanism (2): Token Bucket Shaper

token bucket: limit input to specified *burst size* and *average rate*

- Tokens generated at rate *r tokens/sec,* put into bucket
- Bucket can hold b tokens. Tokens are dropped if bucket is full
- A packet can leave as long as a token is available
- Over interval of time t: number of packets that leave the token bucket is less than or equal to (r * t + b)

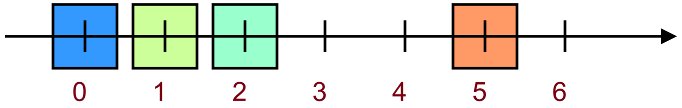


Token Bucket Shaper

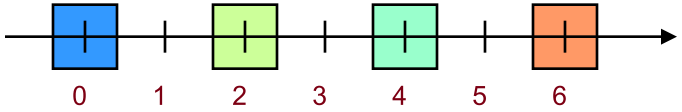
- There is a bucket for tokens and a packet buffer for packets
  - Packets will be dropped if the associated packet buffer is full
  - A token bucket policer doesn’t contain a packet buffer: any packet arriving without a token is immediately dropped
- Bucket of tokens enables small bursts to go through unscathed
  - Short flows can exceed rate limit r, since they use up the reserve in the bucket
  - Long flows will fit into the rate limit r over longer periods of time, since once they use up the reserve, they are limited by the token-fill rate r

Token Bucket vs. Leaky Bucket

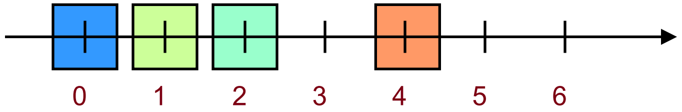
Case 1: Small burst of packet arrivals



Arrival time at bucket



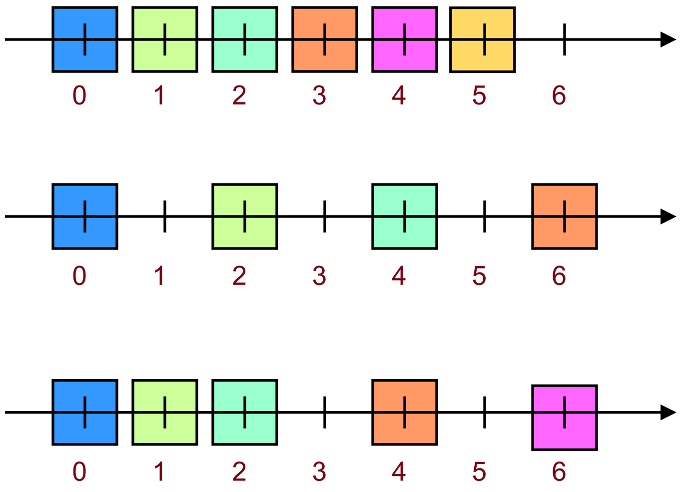
Departure time from a leaky bucket shaper
Leaky bucket rate = 1 packet in 2 time units
Assume packet buffer size is very large



Departure time from a token bucket
Token bucket fill rate = 1 tokens in 2 time units
Token bucket size = 2 tokens
Assume packet buffer size is very large

Token Bucket vs. Leaky Bucket

Case 2: Large burst of packet arrivals



Arrival time at bucket

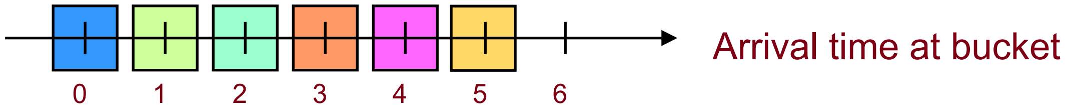
Departure time from a leaky bucket shaper
Leaky bucket rate = 1 packet in 2 time units
Assume packet buffer size is very large

Departure time from a token bucket shaper
Token bucket fill rate = 1 token in 2 time units
Token bucket size = 2 tokens
Assume packet buffer size is very large

??

Departure time from a token bucket policer
Token bucket fill rate = 1 token in 2 time units
Token bucket size = 2 tokens

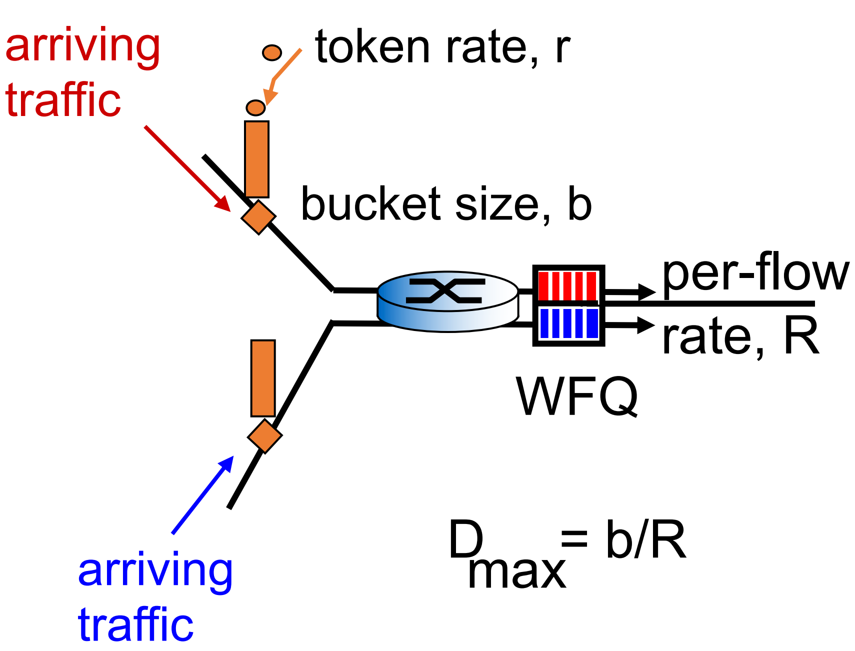
Poll #4



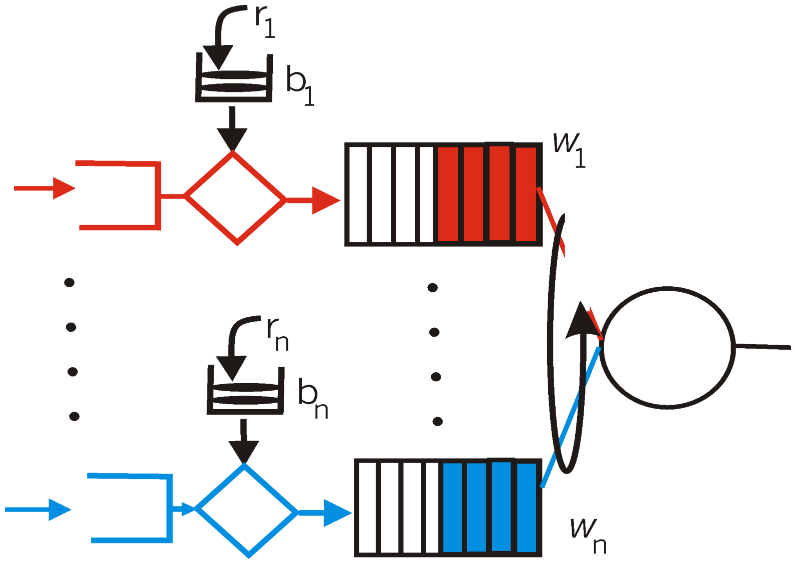
- Suppose the packets above arrived at a token bucket policer with a fill rate of 1 token per 2 time units, and a token bucket size of 2 tokens. What happens to packet 3 above as it arrives?
  - (a) Buffered into the packet buffer
  - (b) Transmitted out of the token bucket
  - (c) Packet is dropped
  - (d) None of the above

Token buckets can be used to provide delay QoS guarantees

- token bucket + WFQ combine to provide a guaranteed upper bound on delay



$D_{max} = b/R$



Impact of Token Bucket Policers

The Internet has tons of them, and they affect application performance

Google study from 2016

Region	Policed segments (among lossy)	Policed segments (overall)	Loss rate (policed)	Loss rate (non-pol.)
India	6.8%	1.4%	28.2%	3.9%
Africa	6.2%	1.3%	27.5%	4.1%
Asia (w/o India)	6.5%	1.2%	22.8%	2.3%
South America	4.1%	0.7%	22.8%	2.3%
Europe	5.0%	0.7%	20.4%	1.3%
Australia	2.0%	0.4%	21.0%	1.8%
North America	2.6%	0.2%	22.5%	1.0%

Table 2: % segments policed among lossy segments ($\geq$ 15 losses, the threshold to trigger the policing detector), and overall. Avg. loss rates for policed and unpoliced segments.

Flach et al., An Internet-Wide Analysis of Traffic Policing, SIGCOMM 2016

Impact on TCP

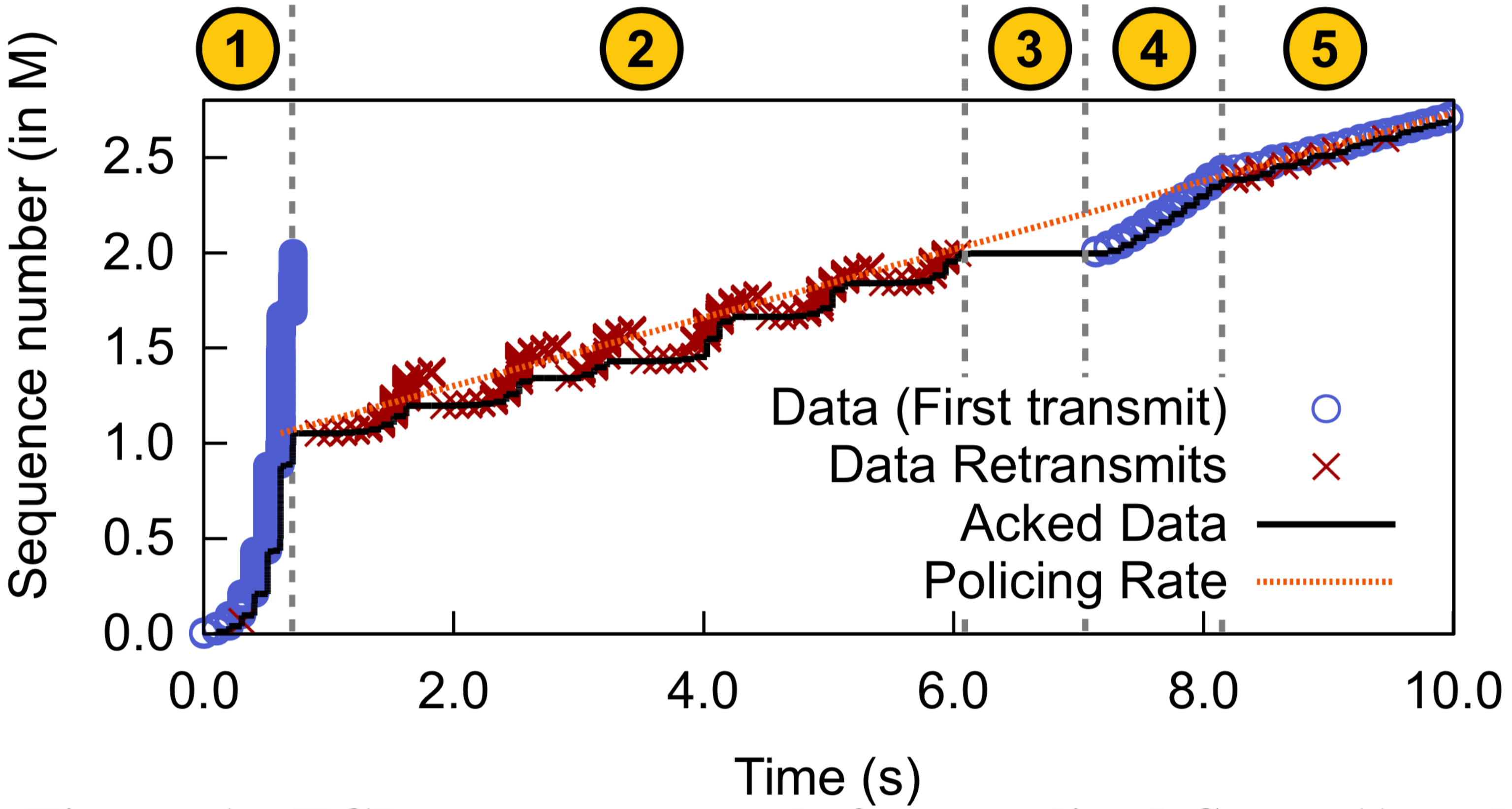
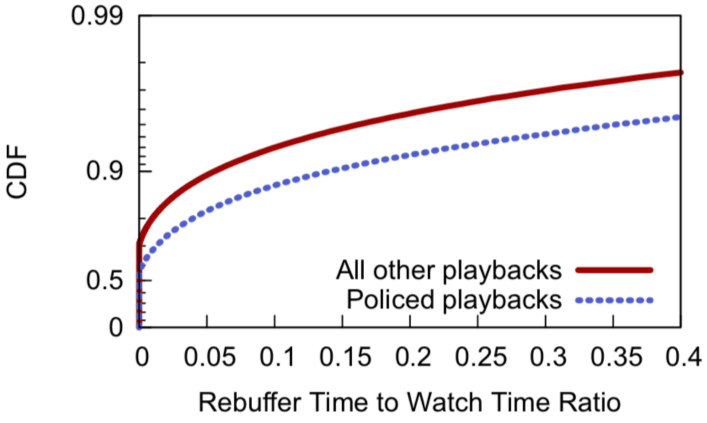


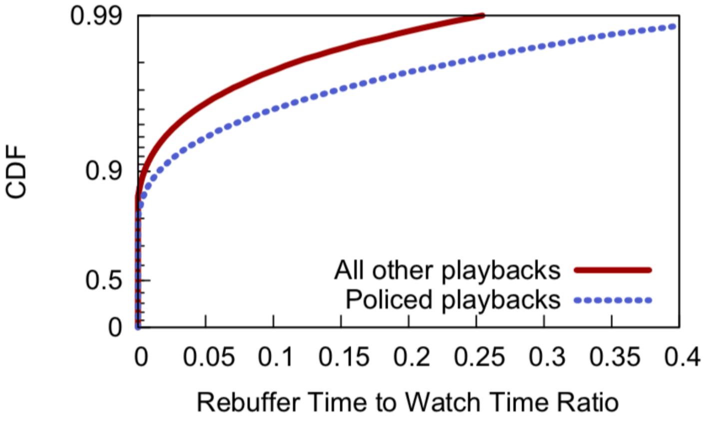
Figure 1: TCP sequence graph for a policed flow: (1 and 4) high throughput until token bucket empties, (2 and 5) multiple rounds of retransmissions to adjust to the policing rate, (3) idle period between chunks pushed by the application.

Policing losses impact applications

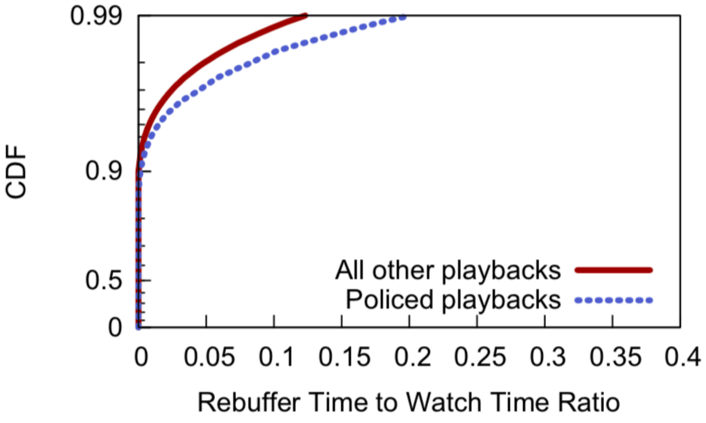
- Video rebuffer rate: rebuffer time / overall watch time



(a) 300 kbps



(b) 1.5 Mbps



(c) 3 Mbps

Figure 9: Rebuffer to watch time ratios for video playbacks. Each had at least one chunk with a goodput of 300 kbps, 1.5 Mbps, or 3 Mbps ($\pm 15\%$).