CS 43: Computer Networks
Traffic Management

Kevin Webb

Swarthmore College

November 21, 2017

Overview

- We’ve seen the behavior of TCP/IP, and routers

- We’ve joked about the option of marking packets as “urgent”
  - As a lone user, your cries for urgency will likely be ignored by one or more ISPs on the Internet

- False implication: All traffic is treated equally.

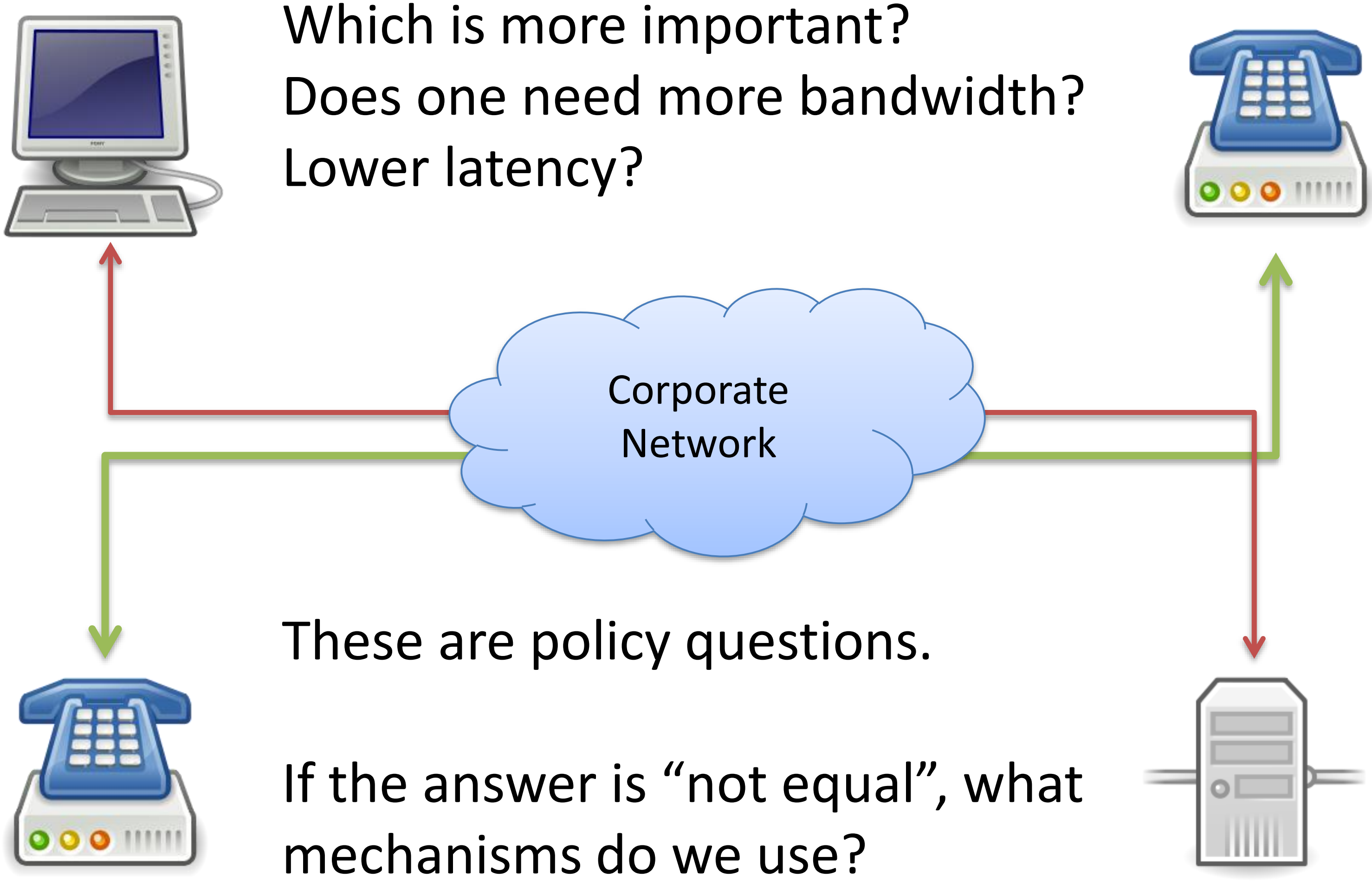
Scenarios

- Things we can do at the network layer to:
  - Treat traffic differently
  - Improve congestion control

- You own a private network
  - Corporate network
  - Data center
  - ISP

- You want to provide better performance to:
  - More important services
  - Customers who pay more

Example 1: Corporate Phones



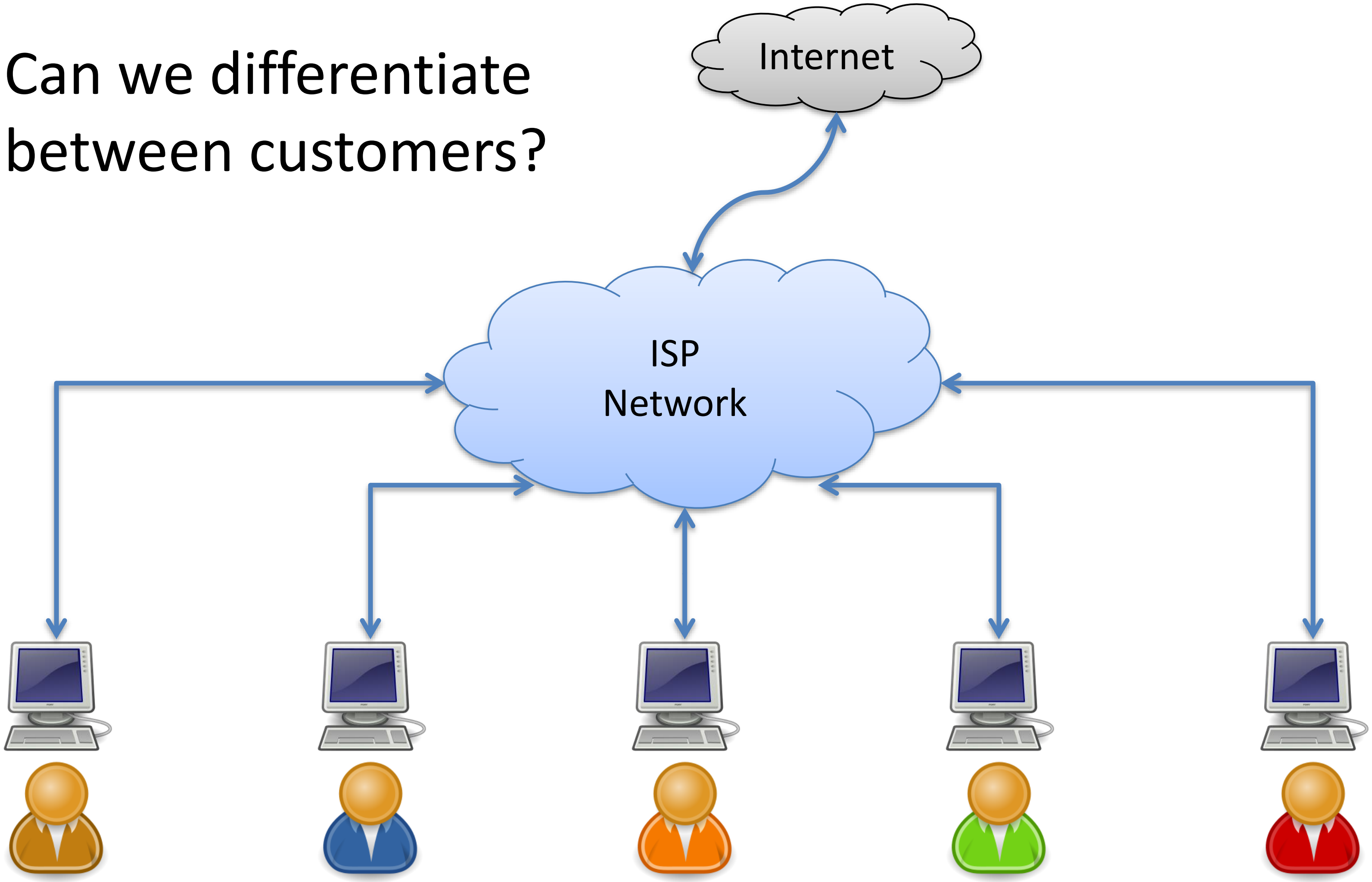
Which is more important?
Does one need more bandwidth?
Lower latency?

These are policy questions.

If the answer is “not equal”, what mechanisms do we use?

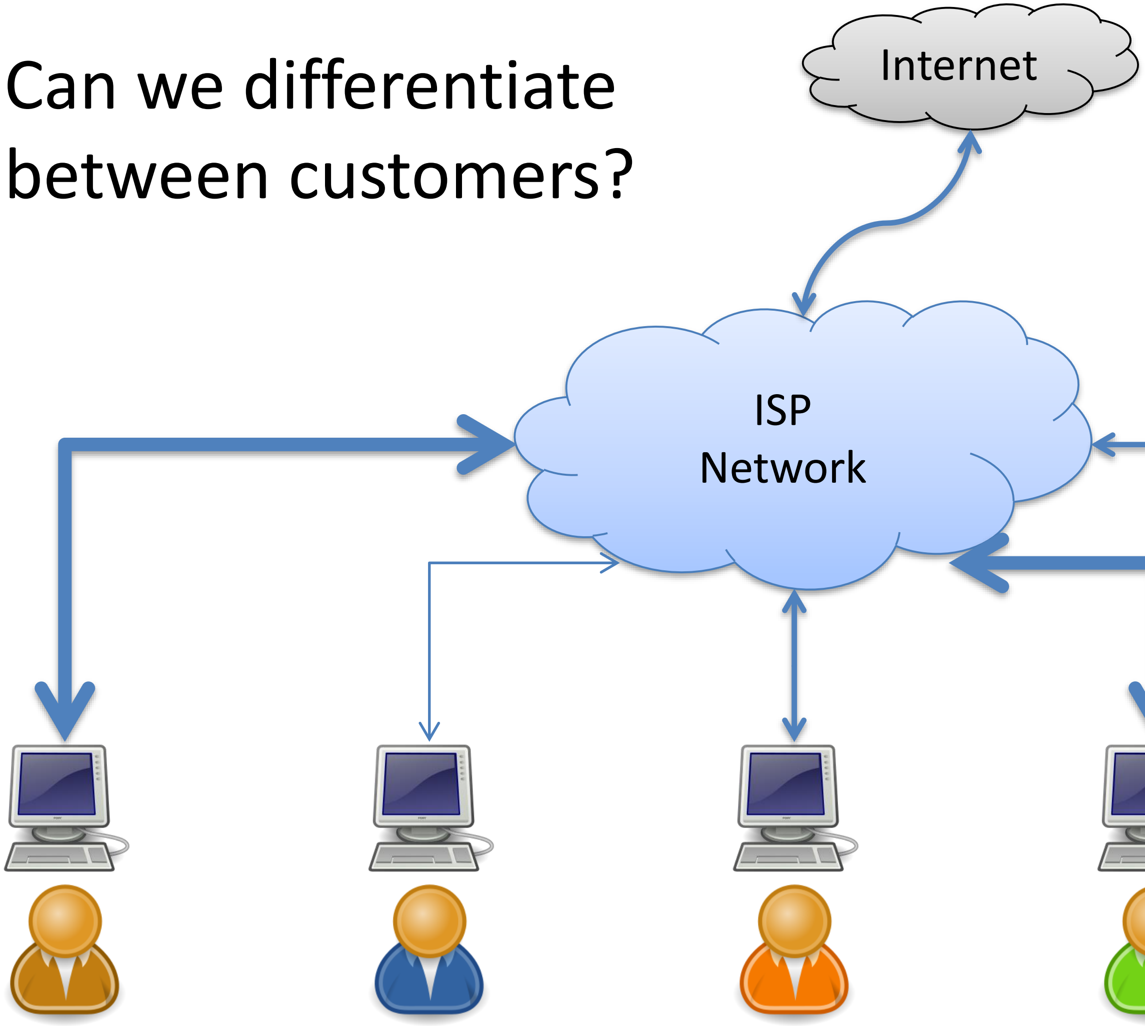
Example 2: ISP Customers

Can we differentiate between customers?



Example 2: ISP Customers

Can we differentiate between customers?



Common policy:

Pay more for faster service!

How might we enforce these types of policies?

A. Require that end-hosts police their traffic.

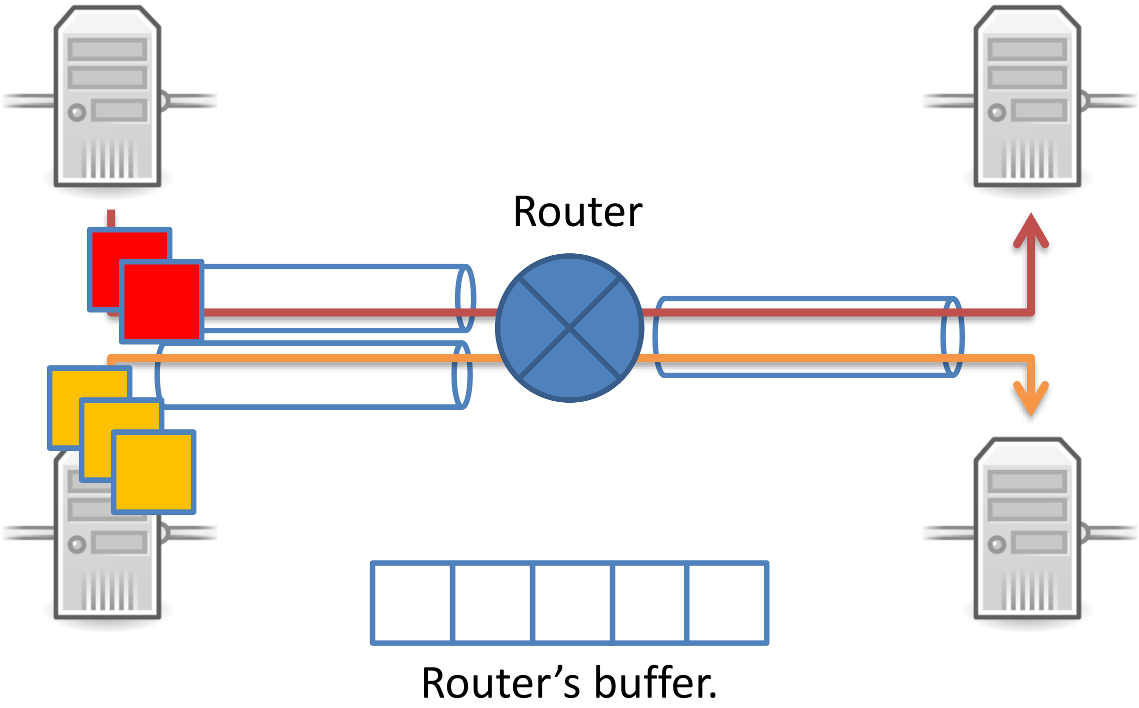
B. Change how routers queue traffic.

C. Ask users nicely to comply with policy.

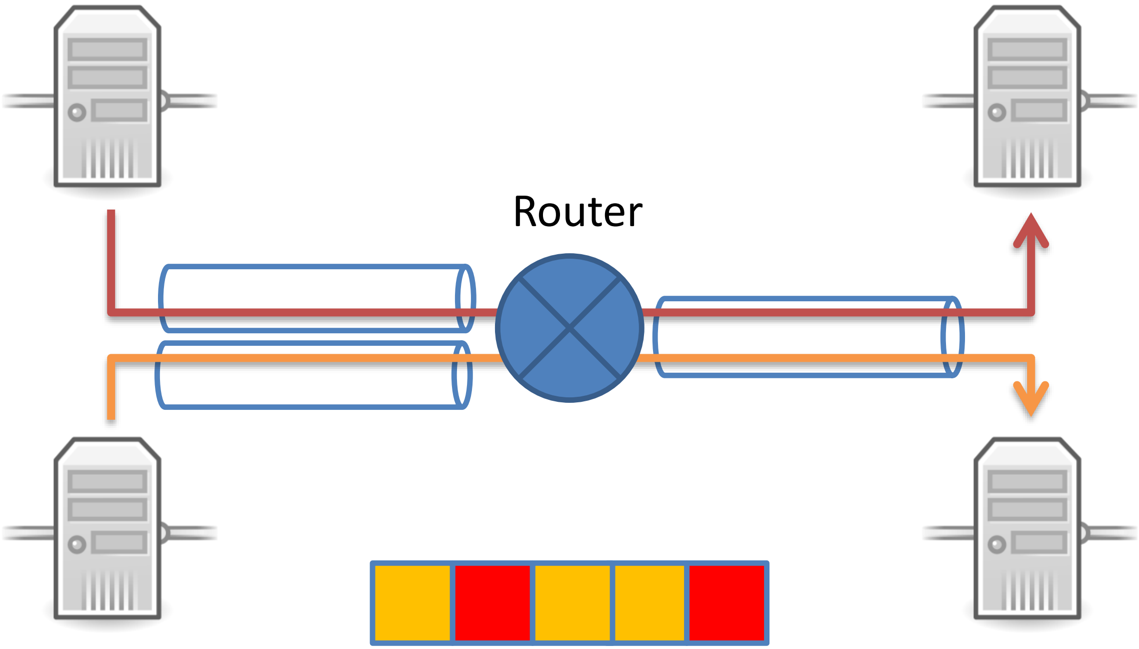
D. Enforce policies some other way.

E. There is nothing we can do.

Recall Queueing



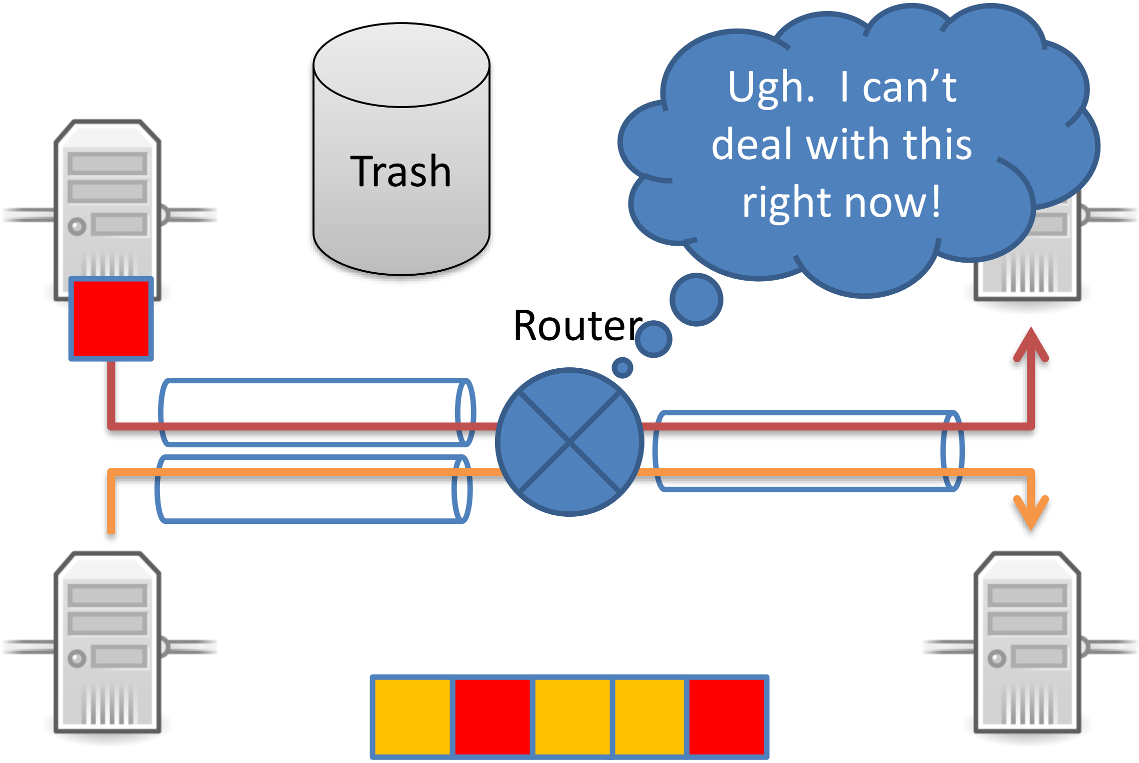
Recall Queueing



Router’s buffer.

Incoming rate is faster than
outgoing link can support.

Recall Queueing



Router's buffer.

Incoming rate is faster than outgoing link can support.

Basic Buffer Management

- FIFO + drop-tail
  - Simplest choice
  - Used widely in the Internet
- FIFO (first-in-first-out)
  - Traffic queued in first-come, first-served fashion
- Drop-tail
  - Arriving packets get dropped when queue is full
- Important distinction:
  - FIFO: queueing (scheduling) discipline
  - Drop-tail: drop policy

FIFO/Drop-Tail Problems

- Doesn’t differentiate between flows/users
- No policing: send more, get more service
- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Synchronization: hosts react to same events

FIFO/Drop-Tail Problems

- Doesn’t differentiate between flows/users
- No policing: send more, get more service

QoS

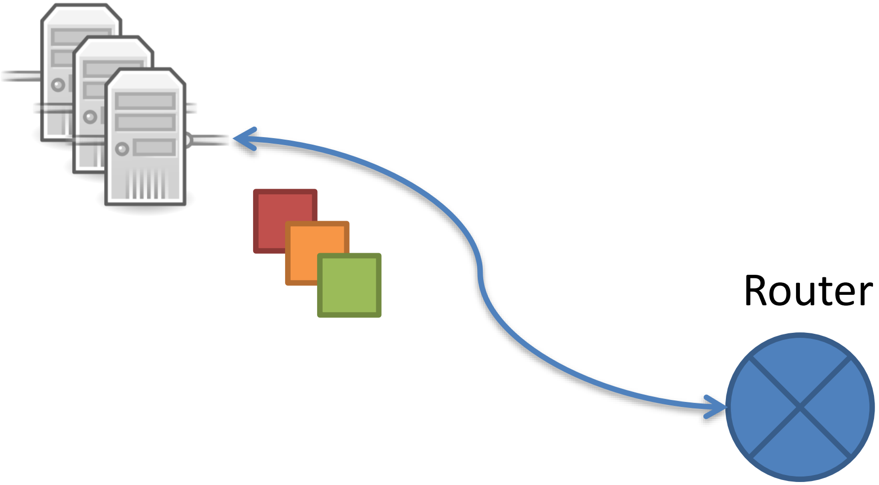
- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Synchronization: hosts react to same events

AQM

Quality of Service (QoS)

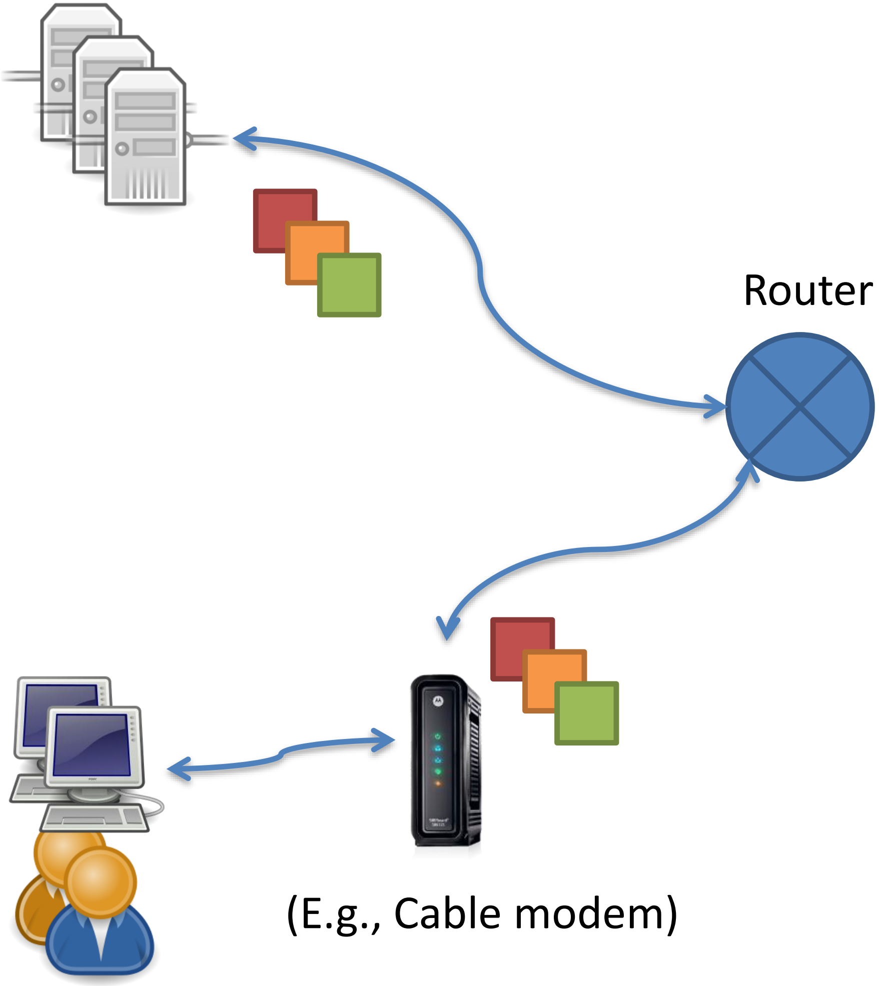
- QoS is a broad topic! We're going to discuss:
  - Mechanism for differentiating users/flows
  - Mechanism for enforcing rate limits
  - Mechanism for prioritizing traffic

Differentiating Users



- If you control end hosts:
  - Mark packets in OS according to policy.
- Take advantage of IP's class of service or options header fields

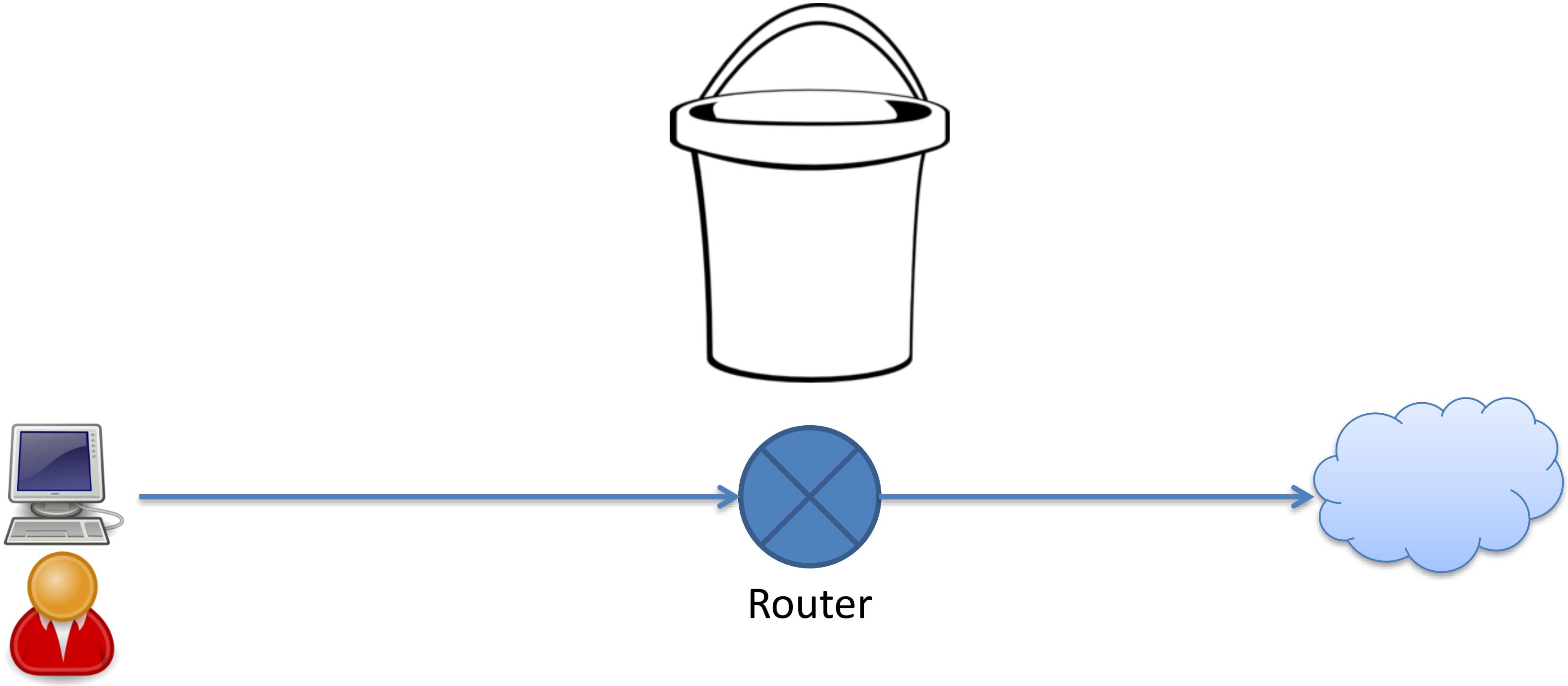
Differentiating Users



- If you control end hosts:
  - Mark packets in OS according to policy.
- Take advantage of IP's class of service or options header fields
- Otherwise:
  - Introduce an intermediate device you trust.

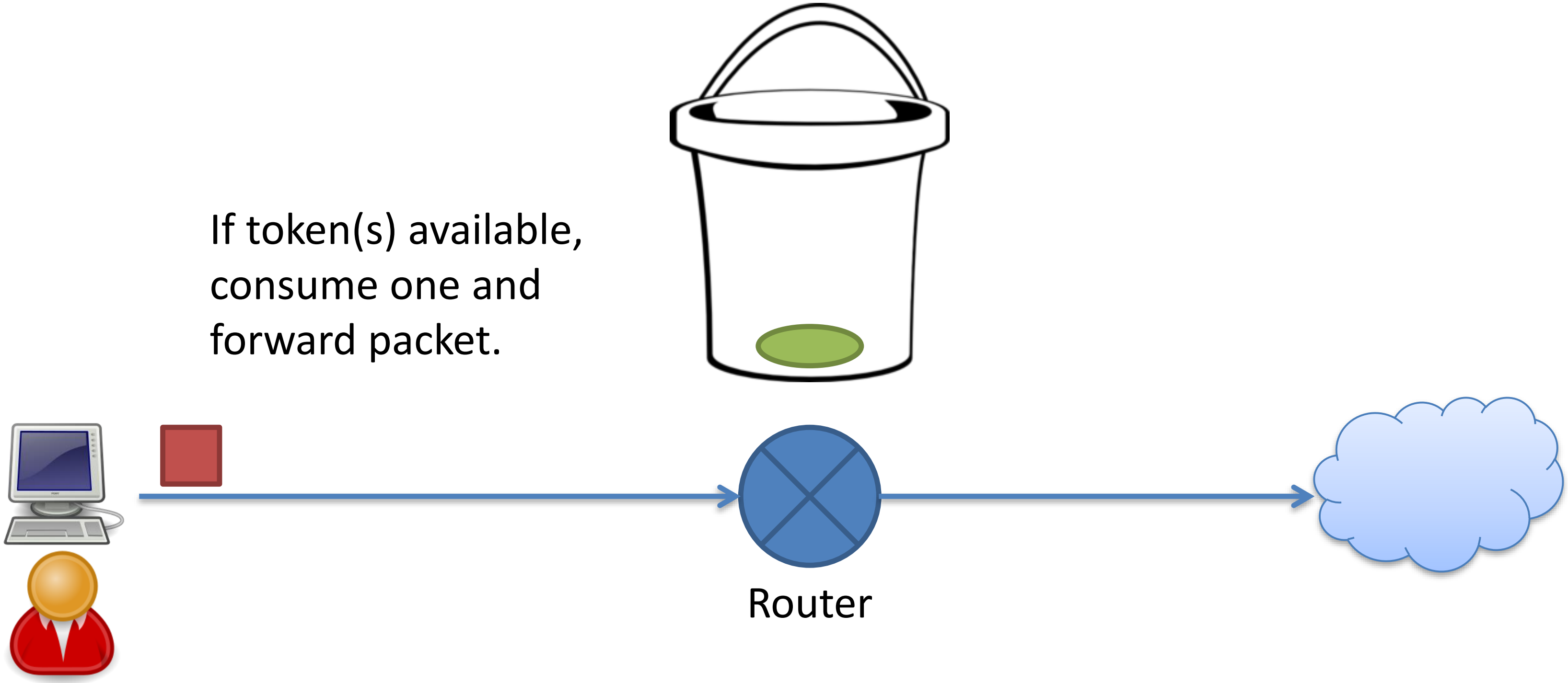
Enforcing (Policing) Rate Limits

- Example: the red user gets at most 10 Mbps
- Solution: Token bucket



Enforcing (Policing) Rate Limits

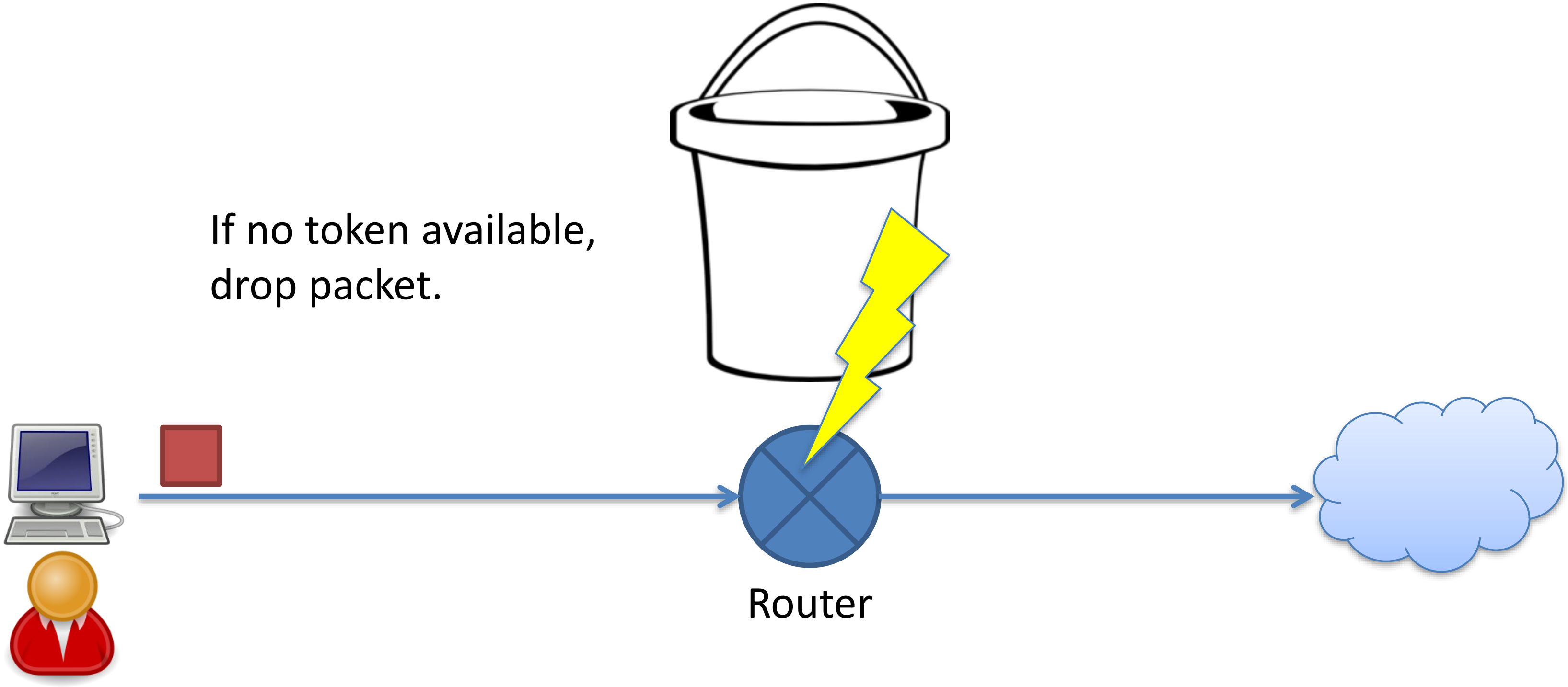
- Example: the red user gets at most 10 Mbps
- Solution: Token bucket



If token(s) available, consume one and forward packet.

Enforcing (Policing) Rate Limits

- Example: the red user gets at most 10 Mbps
- Solution: Token bucket



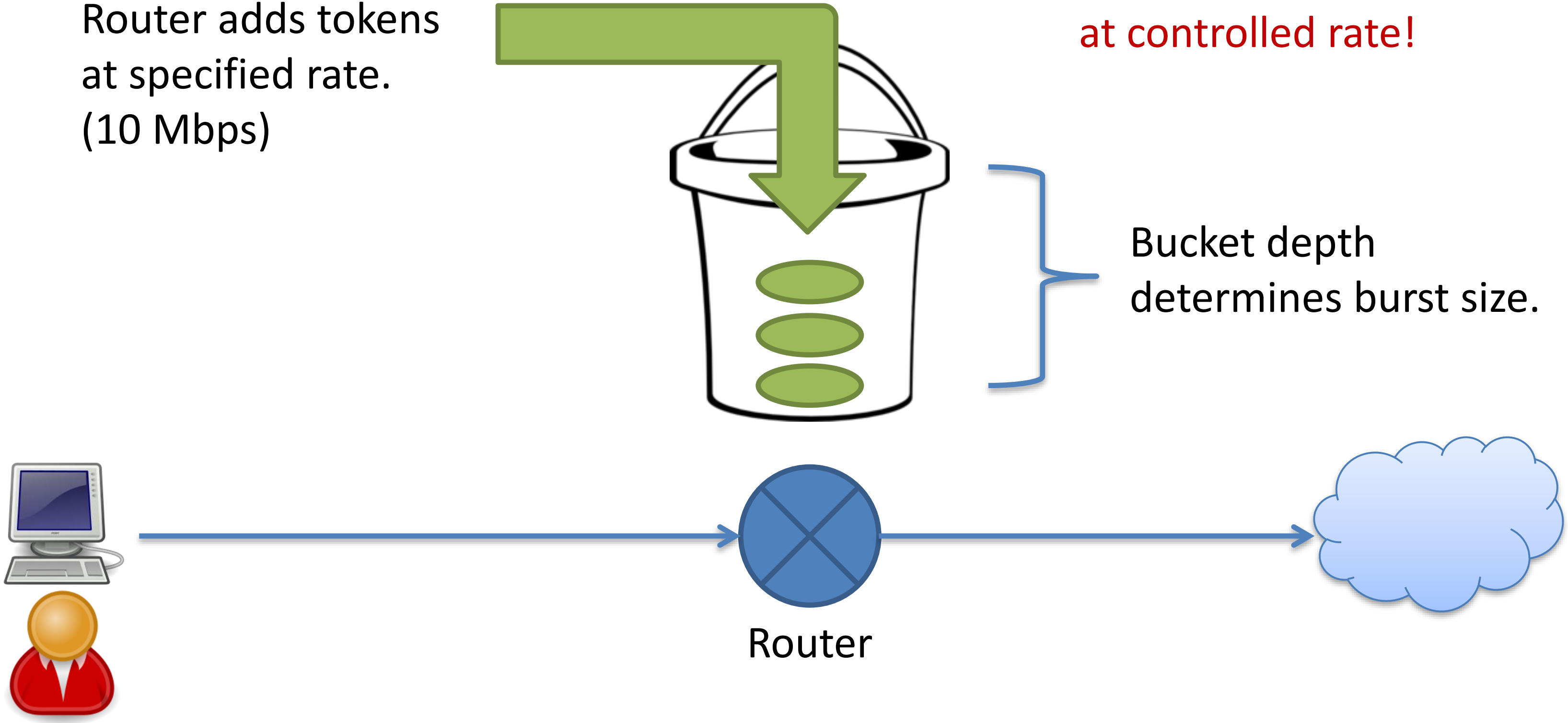
If no token available, drop packet.

Enforcing (Policing) Rate Limits

- Example: the red user gets at most 10 Mbps
- Solution: Token bucket

No matter how fast user sends, limited by number of tokens, which replenish at controlled rate!

Router adds tokens at specified rate. (10 Mbps)



Bucket depth determines burst size.

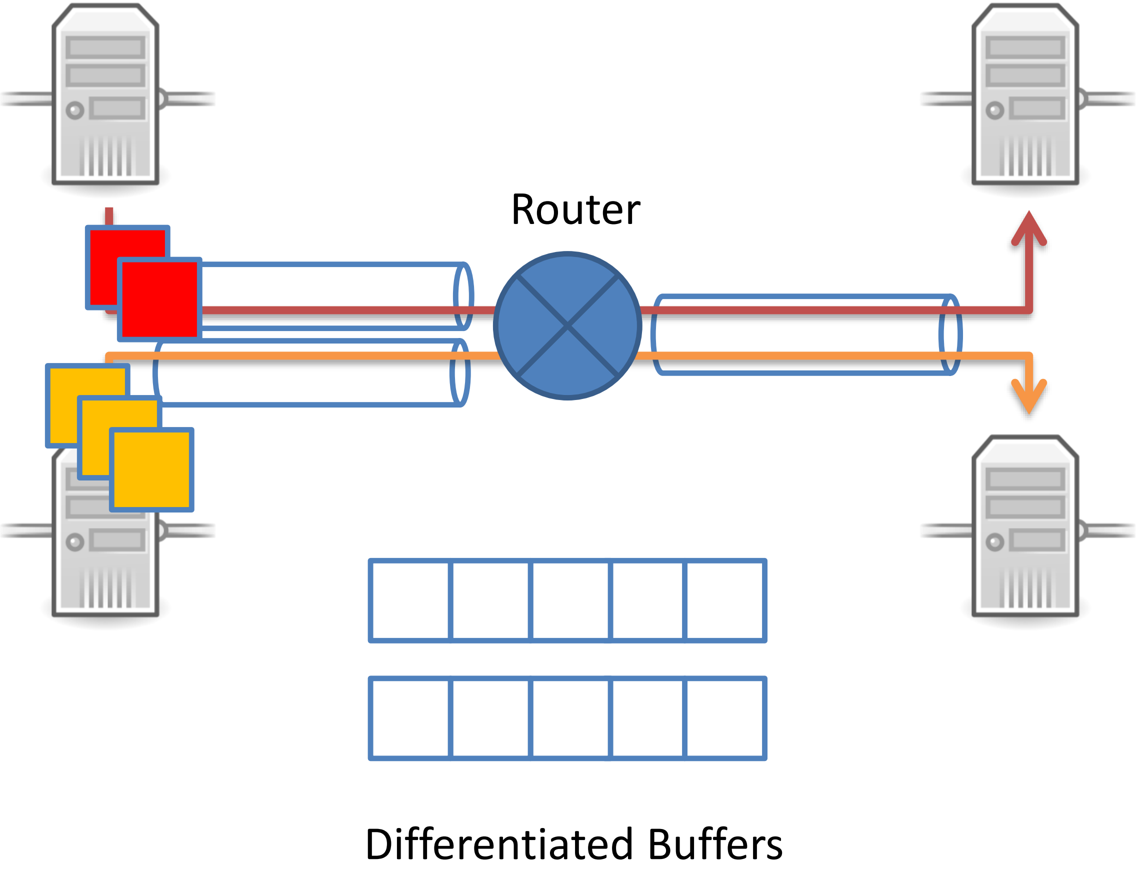
Router

Prioritizing

- Been to a theme park recently?

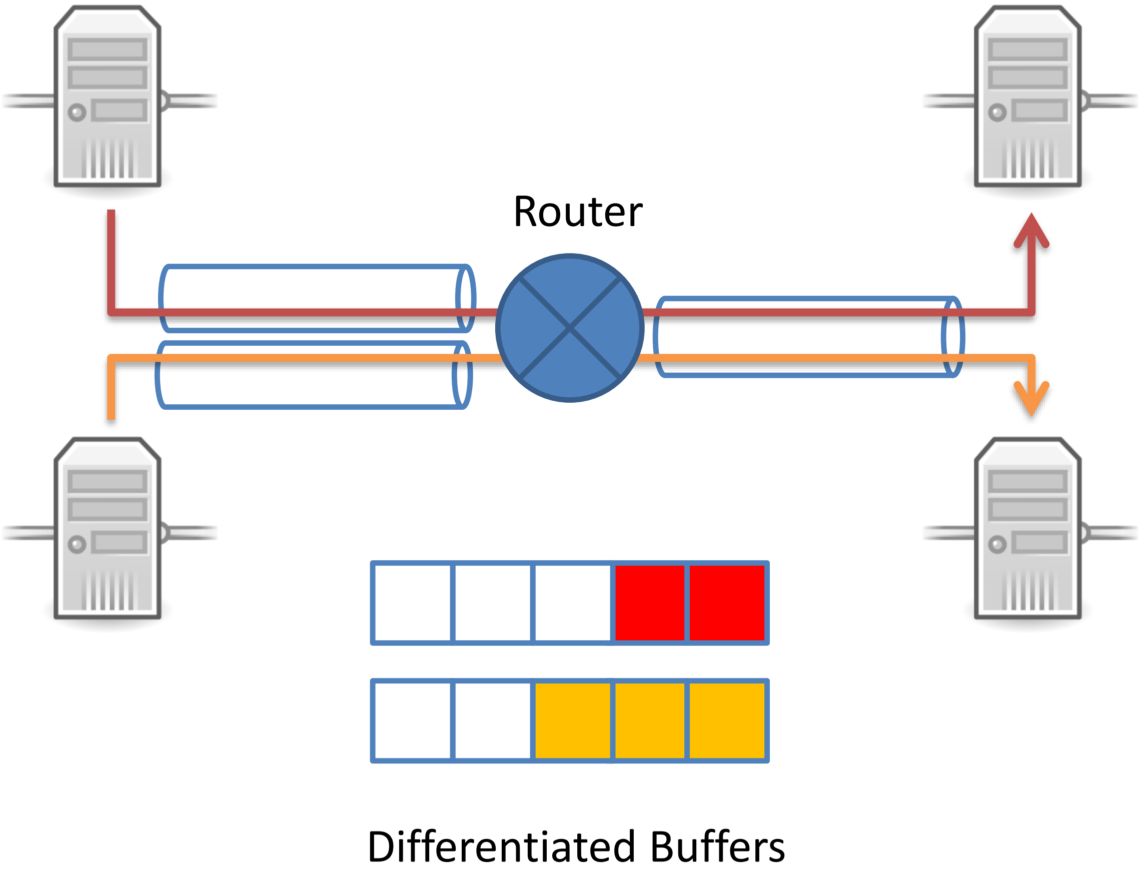
Prioritizing Traffic

- Designate multiple classes of traffic.



Prioritizing Traffic

- Weight queues differently.



Weighted Fair Queueing

- Suppose orange is more important than red.

- Policy: Always empty orange's queue first.
  - Problem: Red might starve!

- Policy: Always allow 1 red packet for every N orange packets.
  - Ratio is known as weight.

FIFO/Drop-Tail Problems

- Doesn't differentiate between flows/users
- No policing: send more, get more service

QoS

- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Synchronization: hosts react to same events

AQM

Active Queue Management

- Design active router queue management to aid congestion control

- Why?
  - TCP at end hosts have limited vantage point
  - Routers see actual queue occupancy

- “Hint”: TCP will still do congestion control
  - We can try to help it out in the network!

How might we take advantage of TCP’s behavior to help it discover congestion in the network?

A. Drop packets, even when they could be sent.

B. Hold packets in the queue, even when they could be sent.

C. Send a congestion notification back to the sender.

D. Send a congestion notification to the receiver.

E. Some other mechanism.

Random Early Detection (RED)

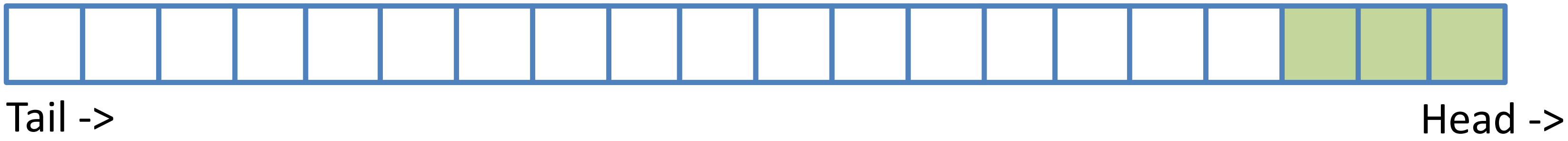
- Goal: Prevent congestion before it’s a problem
- Assume hosts respond to lost packets
- Avoid window synchronization
  - Randomly mark packets
- Avoid bias against bursty traffic

RED Algorithm

- Maintain running average of queue length
- If avg < min_{th} do nothing
  - Low queuing, send packets through
- If avg > max_{th}, drop packet
  - Protection from misbehaving sources
- Else drop/mark packet in a manner proportional to queue length
  - Notify sources of incipient congestion

RED

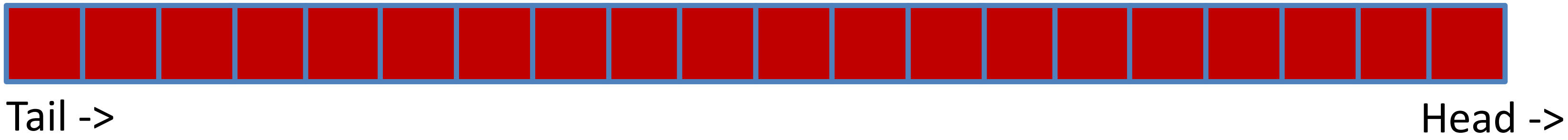
- Router queue:



- Mostly empty? Don’t drop.

RED

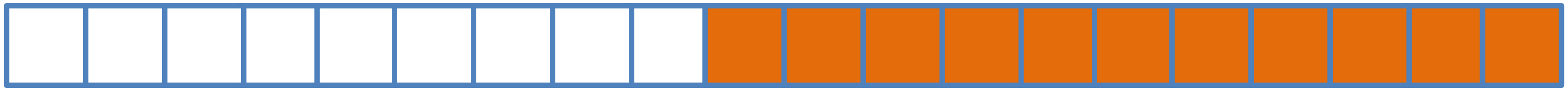
- Router queue:



- Mostly full? Drop new packets.

RED

- Router queue:



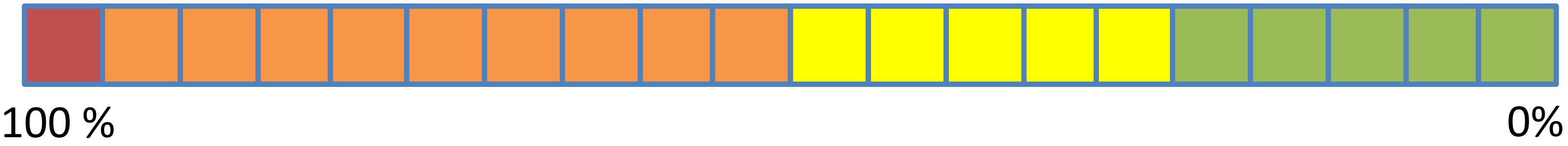
Tail ->

Head ->

- In the middle? Drop proportionally to how full the queue is!

RED

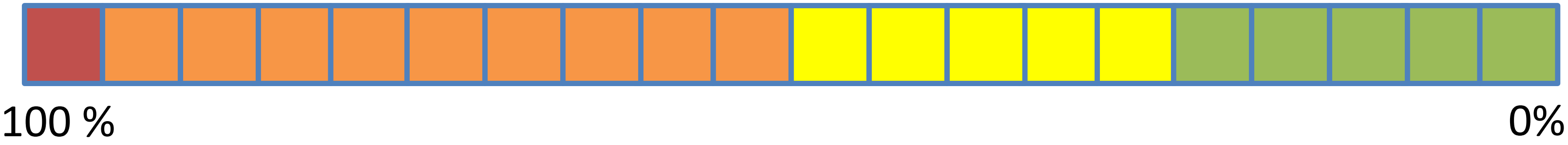
- Drop probability:



- In the middle? Drop proportionally to how full the queue is!

ECN

- ~~Drop~~ Mark probability:



- Explicit congestion notification: Instead of dropping, set a header field, which gets returned to sender in ACK.

- Treat marked packets as “congestion events”

Summary

- Not all traffic is (should be?) treated equally
  - We can differentiate by marking traffic

- Routers exert power by managing their queue
  - Queueing disciplines: WFQ, RED
  - Can impose other mechanisms (token bucket)

“Net Neutrality”

- Big “Tier one” ISPs probably don’t care much about what you do, but your local ISP might.
- Example: Comcast didn’t like BitTorrent, started injecting RSTs into user TCP streams.
- Scarier example: You like Netflix, but your ISP has their own video service. They degrade (or block) Netflix service unless you pay $$$.

“Net Neutrality”

- Neutrality: Call for legislation to prevent ISPs from imposing arbitrary restrictions on the types of data users can transmit.

“Net Neutrality”

Cases for:

- End to end principle
- Prevent customer extortion
- Allow for innovation

Google, Microsoft, Yahoo, Amazon, eBay

Cases against:

- ISP owns their network
- Asymmetric application bandwidth usage
- We shouldn’t legislate the Internet, it moves too fast

Cisco, many ISPs