Computer Networks WS14/15
Homework #10

QoS Building Blocks

- Q: Name three building blocks of QoS

QoS Building Blocks

- Q: Name three building blocks of QoS

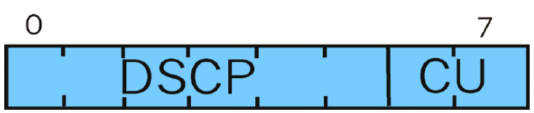
- Classification: distinguish one flow/class from another
- Policing: ensure traffic is conform to contracted parameters
- Scheduling: ensure QoS guarantees (bandwidth, delay)
- Admission control and resource reservation: check if resource requirements can be met and admit or deny flows accordingly

QoSBuilding Blocks

Classification: distinguish one flow/class from another

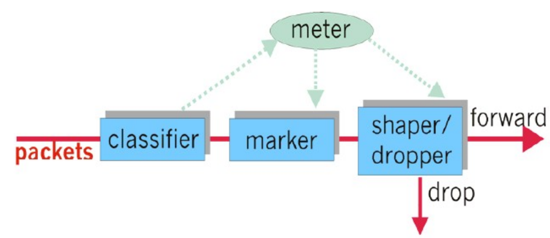
Classification and Conditioning

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB that the packet will receive
- 2 bits are currently unused



Classification and Conditioning

- may be desirable to limit traffic injection rate of some class:
  - user declares traffic profile (e.g., rate, burst size)
  - traffic metered, shaped if non-conforming



QoS Building Blocks

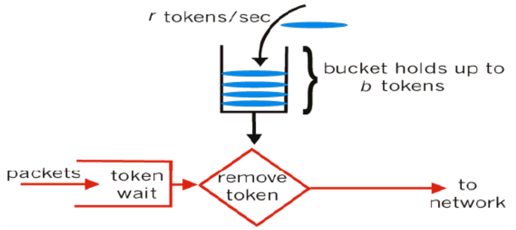
Policing: ensure traffic conforms contracted parameters

Policing Mechanisms

- Goal: limit traffic to not exceed declared parameters
- Three common-used criteria:
- (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!
- 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)

Policing Mechanisms

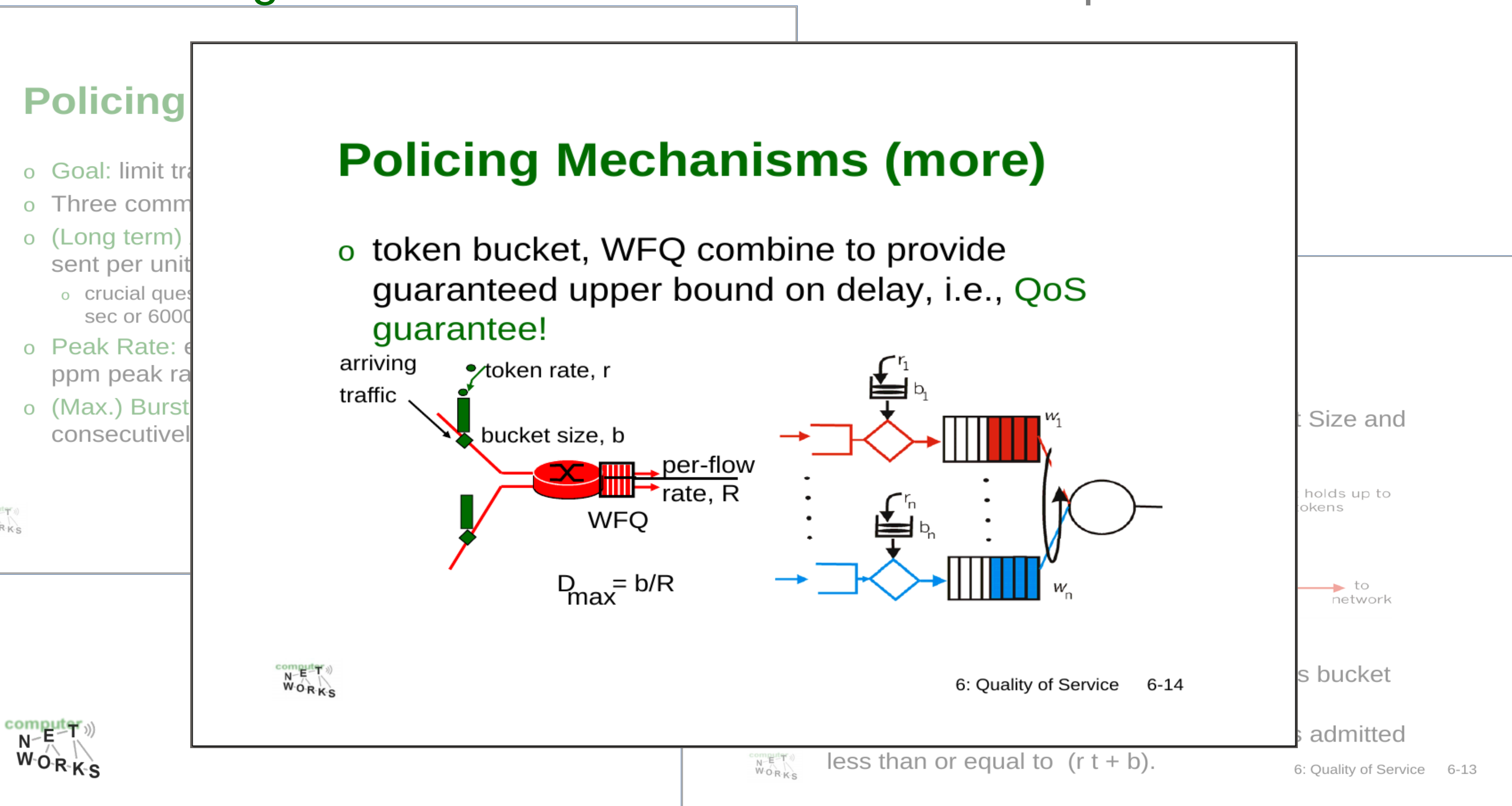
- Token Bucket: limit input to specified Burst Size and Average Rate.



- bucket can hold b tokens
- tokens generated at rate r token/sec unless bucket full
- over interval of length t: number of packets admitted less than or equal to (r t + b).

QoS Building Blocks

Policing: ensure traffic conforms contracted parameters

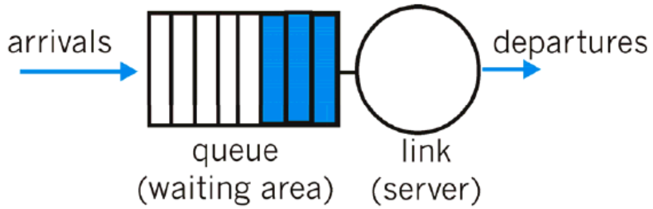


QoS Building Blocks

Scheduling: ensure QoS guarantees (bandwidth, delay)

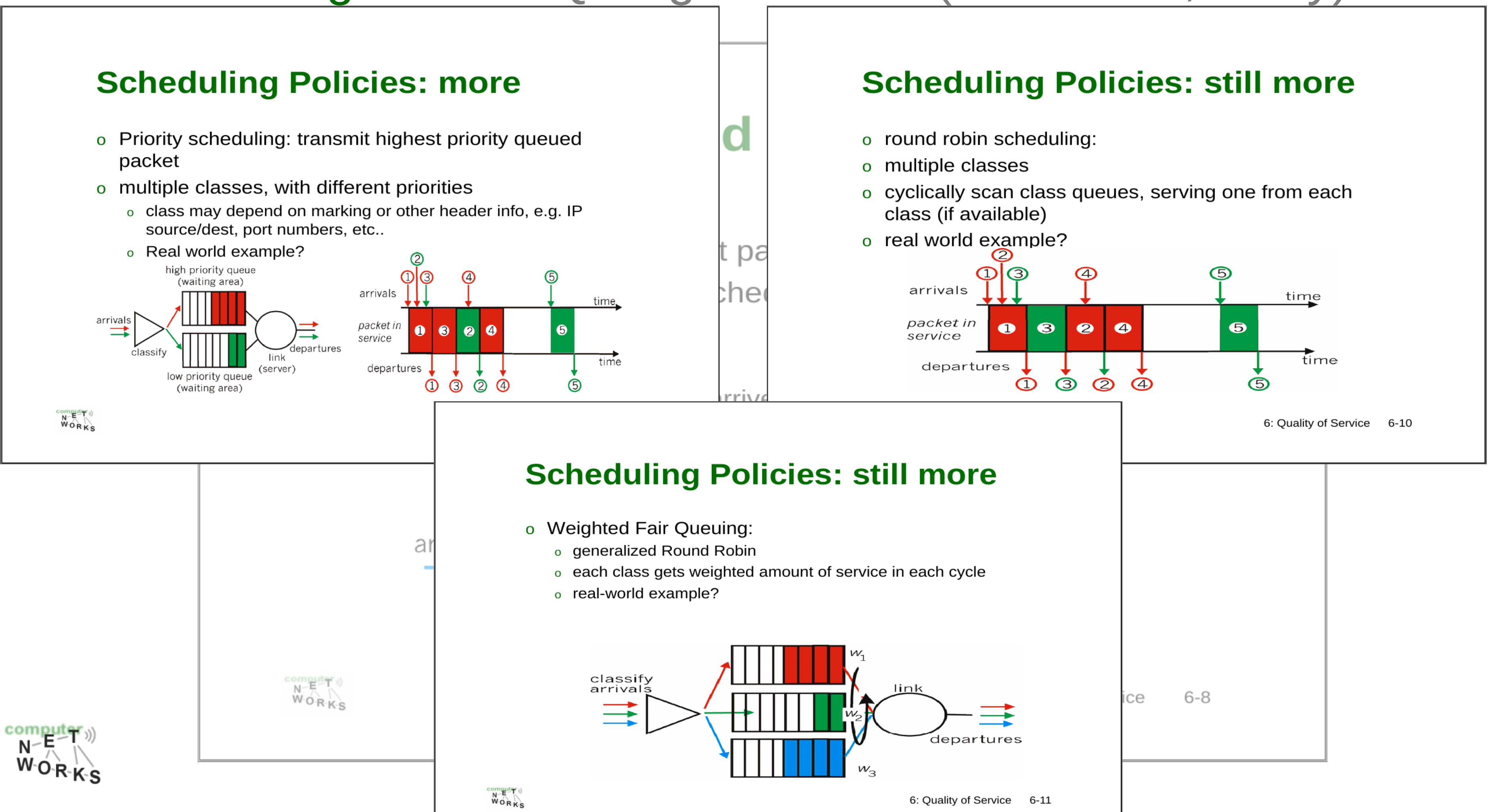
Scheduling And Policing Mechanisms

- scheduling: choose next packet to send on link
- FIFO (first in first out) scheduling: send in order of arrival to queue
  - real-world example?
  - discard policy: if packet arrives to full queue: who to discard?
    - Tail drop: drop arriving packet
    - priority: drop/remove on priority basis
    - random: drop/remove randomly



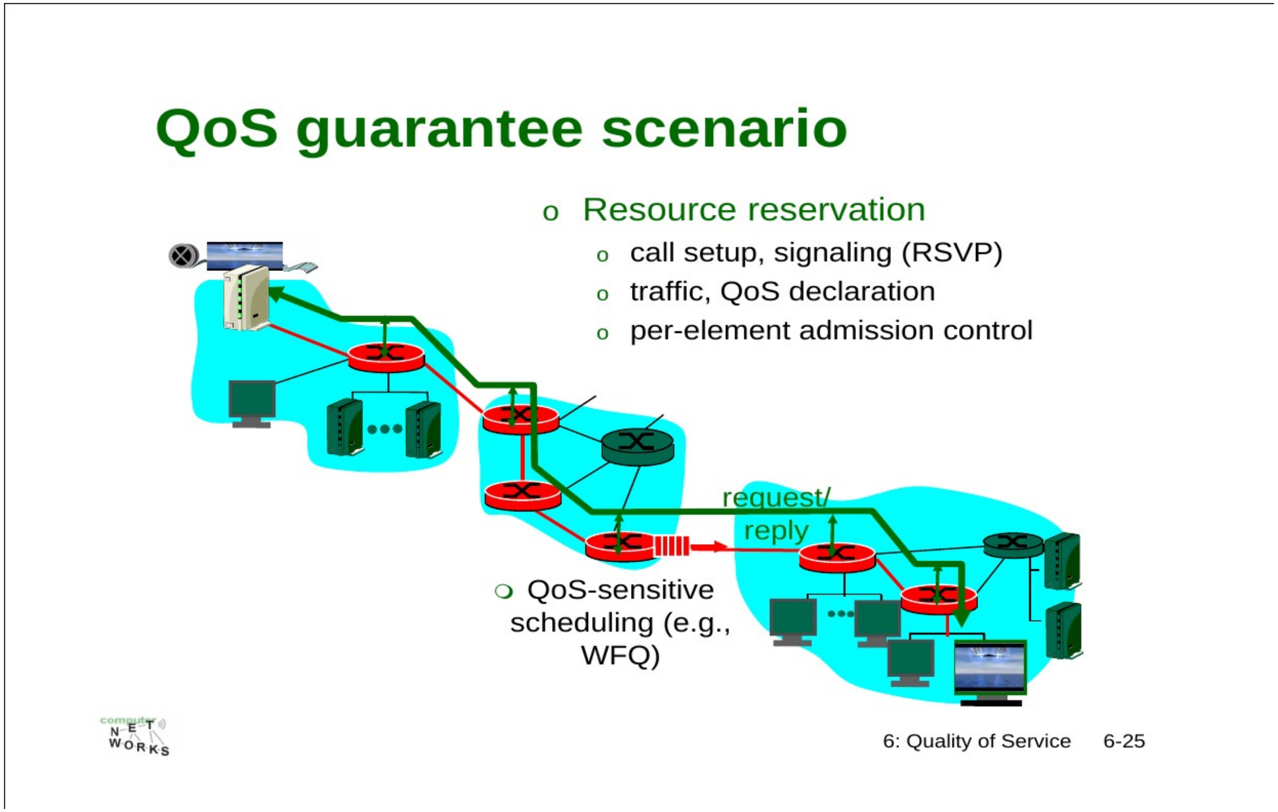
QoS Building Blocks

Scheduling: ensure QoS guarantees (bandwidth, delay)



QoS Building Blocks

Admission control and resource reservation: check if resource requirements can be met and admit or deny flows accordingly



Scheduling Policies

- Q: Name and characterize four scheduling policies that were introduced in the lecture.

Scheduling Policies

First In First Out (FIFO)

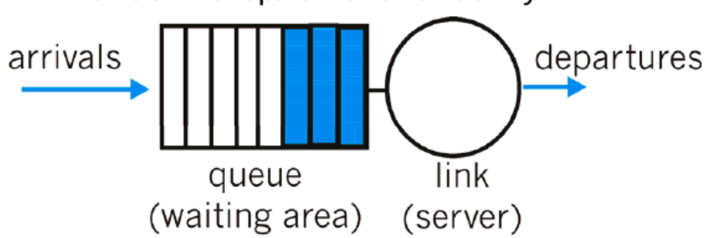
- Packets are sent in order of arrival
- If queue is full, discard either
  - Newly arriving packets
  - Packets based on priority
  - Random packets

Scheduling Policies

First In First Out (FIFO)

Scheduling And Policing Mechanisms

- scheduling: choose next packet to send on link
- FIFO (first in first out) scheduling: send in order of arrival to queue
  - real-world example?
  - discard policy: if packet arrives to full queue: who to discard?
    - Tail drop: drop arriving packet
    - priority: drop/remove on priority basis
    - random: drop/remove randomly



Scheduling Policies (cont'd)

Priority Scheduling

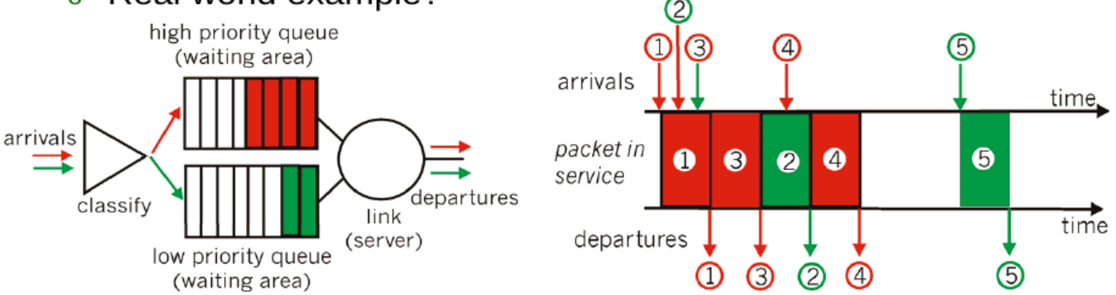
- Packets are sent in order of priority

Scheduling Policies (cont'd)

Priority Scheduling

Scheduling Policies: more

- Priority scheduling: transmit highest priority queued packet
- multiple classes, with different priorities
  - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc..
  - Real world example?



Scheduling Policies (cont'd)

Round Robin

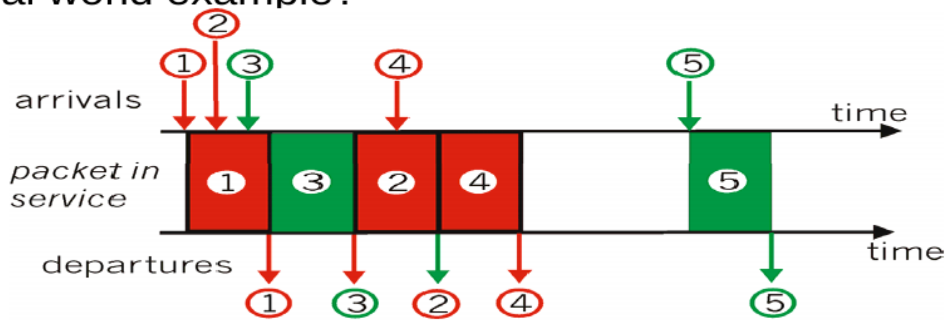
- Packets are sent alternating between different classes

Scheduling Policies (cont’d)

Round Robin

Scheduling Policies: still more

- round robin scheduling:
- multiple classes
- cyclically scan class queues, serving one from each class (if available)
- real world example?



Scheduling Policies (cont'd)

Weighted Fair Queuing

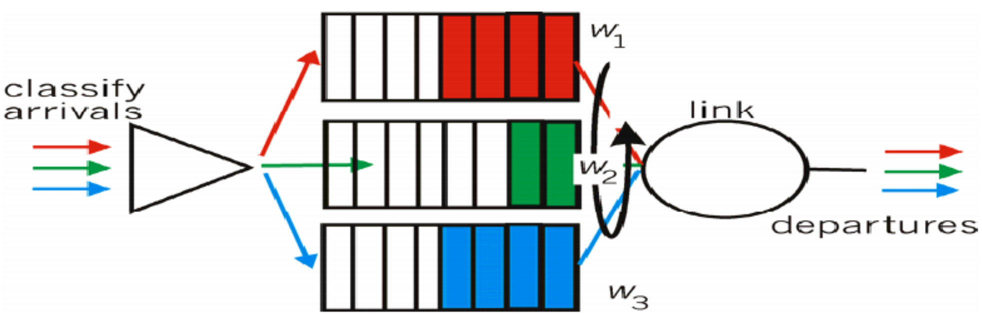
- Combination of Priority Scheduling and Round Robin
- Weighted amount of packets sent from each class every cycle

Scheduling Policies (cont’d)

Weighted Fair Queuing

Scheduling Policies: still more

- Weighted Fair Queuing:
  - generalized Round Robin
  - each class gets weighted amount of service in each cycle
  - real-world example?



Policing

- Q: Which are the criteria that policing mechanisms can use to control a data stream?

Policing

- Q: Which are the criteria that policing mechanisms can use to control a data stream?

- (Long term) Average Rate
  - E.g. 6,000 packets per min (ppm)
- Peak Rate
  - E.g. 6,000 ppm on avg but limited to 1,500 packets per sec (pps) peak rate
- (Max.) Burst Size
  - E.g. 500 packets

Policing

- Q: Which are the criteria that policing mechanisms can use to control a data stream?
- (
-
- (

Policing Mechanisms

- Goal: limit traffic to not exceed declared parameters
- Three common-used criteria:
- (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!
- 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)

Packet Scheduling

- Q: Assume three different queues (a, b, c).
- Illustrate example sequences in which they can be sent according to WFQ scheduling.
- Suppose that the weights for the queues are
  - a: 0.25,
  - b: 0.25, and
  - c: 0.5.

Packet Scheduling

- Q: Assume three different queues (a, b, c).
- Illustrate example sequences in which they can be sent according to WFQ scheduling.
- Suppose that the weights for the queues are
  - a: 0.25,
  - b: 0.25, and
  - c: 0.5.

→ abcc abcc abcc -or- acbc acbc acbc

Packet Scheduling

- Q: Assume three different queues (a, b, c).
- Illustrate example sequences in which they can be sent according to WFQ scheduling.
- Suppose that the weights for the queues are
  - a: 0.25,
  - b: 0.25, and
  - c: 0.5.
- How could the sequence look like if there are no packets in queue 'c'?

→ abcc abcc abcc -or- acbc acbc acbc

Packet Scheduling

- Q: Assume three different queues (a, b, c).
- Illustrate example sequences in which they can be sent according to WFQ scheduling.
- Suppose that the weights for the queues are
  - a: 0.25,
  - b: 0.25, and
  - c: 0.5.
- How could the sequence look like if there are no packets in queue 'c'?

→ abcc abcc abcc -or- acbc acbc acbc

→ abab abab abab

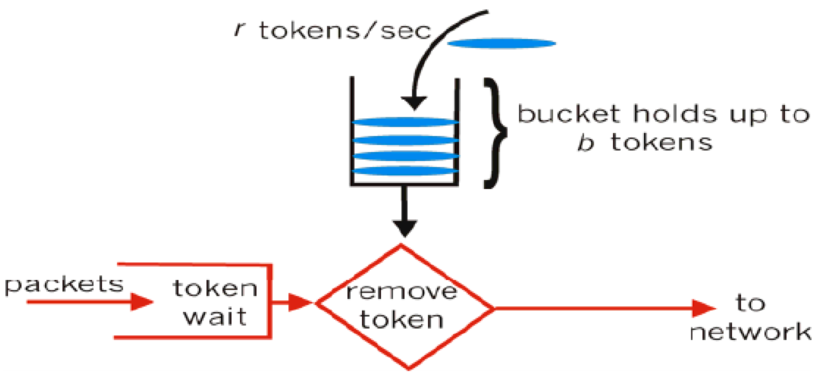
Token Bucket

- Q: Consider a token bucket filled with a rate of 1,000 tokens/min and a size of 500 tokens.
- Each bit of data consumes one token.
- Does a flow of 20bps conform to this bucket?

Token Bucket

Policing Mechanisms

- Token Bucket: limit input to specified Burst Size and Average Rate.



- bucket can hold b tokens
- tokens generated at rate r token/sec unless bucket full
- over interval of length t: number of packets admitted less than or equal to (r t + b).

Token Bucket

- Q: Consider a token bucket filled with a rate of 1,000 tokens/min and a size of 500 tokens.
- Each bit of data consumes one token.
- Does a flow of 20bps conform to this bucket?

- 20 bit per second = 1,200 bit per minute
  → excess of 200 bit per minute
- => The flow does not conform to this bucket
- However: If the bucket is full, flow has 2.5 min to adjust (500 tokens / 200 tokens/min)

Delay Guarantees

- Q: Assume a router maintains QoS garantees for two data flows A and B.
- A is policed by a token bucket with a capacity of 100 tokens and a fill rate of 20 token/sec.
- B is policed by a token bucket with a capacity of 200 tokens and a fill rate of 12 token/sec.
- The router uses weighted fair queuing to schedule the packets with a weight of 3 for flow A and a weight of 2 for flow B. It's sending rate is 40 packets/sec.
- What is the upper bound delay that the router can guarantee for both flows?

Delay Guarantees (sol.)

- Flow A
  - 20 tokens/sec plus 100 tokens if filled
  - Worst case: 100 tokens worth of traffic instantaneously => Queue for A fills up with 100 packets
  - Outgoing rate is 40 packets/sec of which A is allocated at least 3/5 => 24 packets/sec
  - Worst case: A has average (incoming) flow rate of 20 packets/sec => only at least 4 packets/sec can be removed from the queue
  - Last packet in the queue will be delivered after 100/4 = 25 sec

Delay Guarantees (sol.)

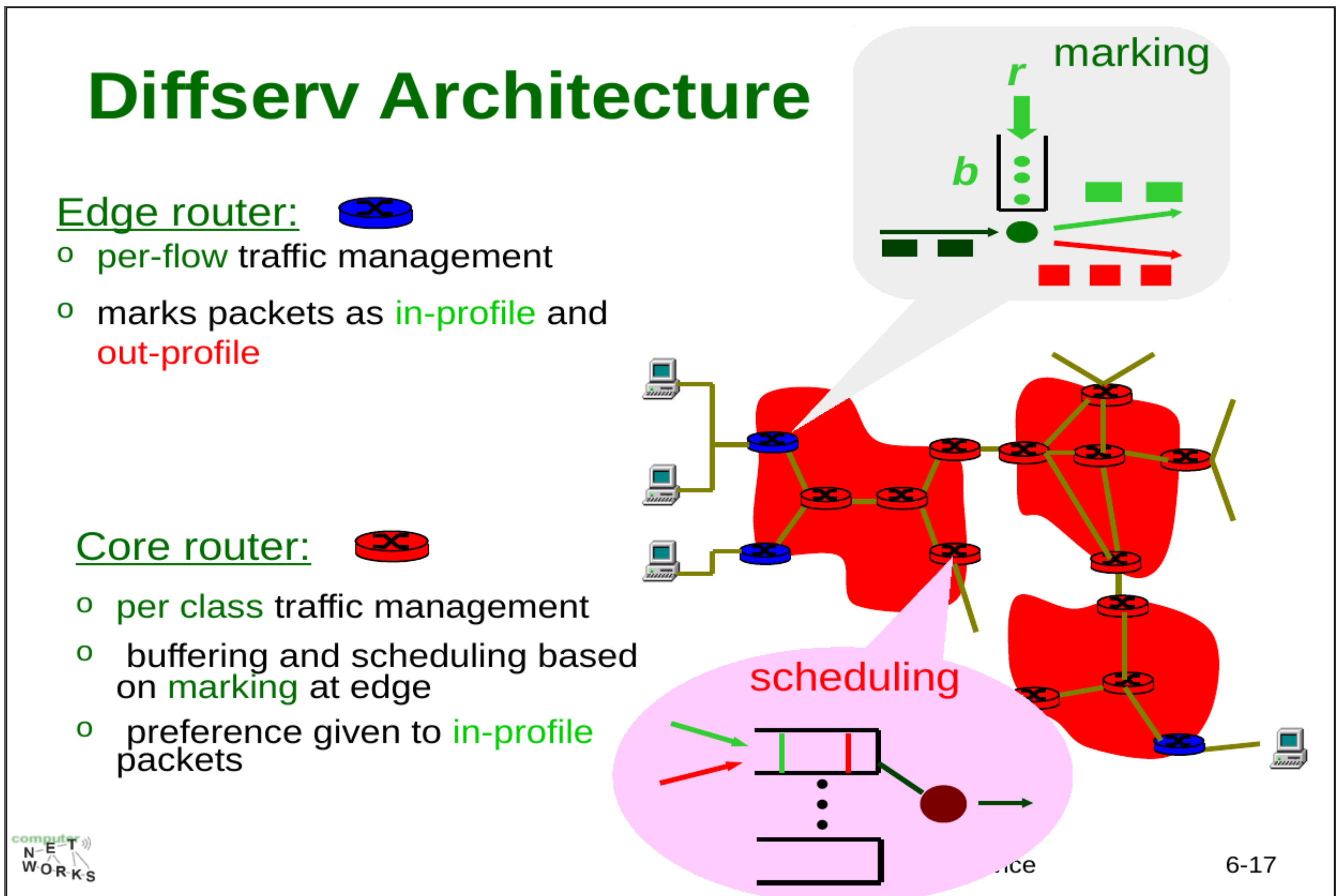
- Flow B
  - 12 tokens/sec plus 200 tokens if filled
  - Worst case: 200 tokens worth of traffic instantaneously => Queue for B fills up with 200 packets
  - Outgoing rate is 40 packets/sec of which A is allocated at least 2/5 => 16 packets/sec
  - Worst case: B has average (incoming) flow rate of 12 packets/sec => only at least 4 packets/sec can be removed from the queue
  - Last packet in the queue will be delivered after 200/4 = 50 sec

IntServ and DiffServ

- Q: Briefly compare the IntServ to the DiffServ architecture.

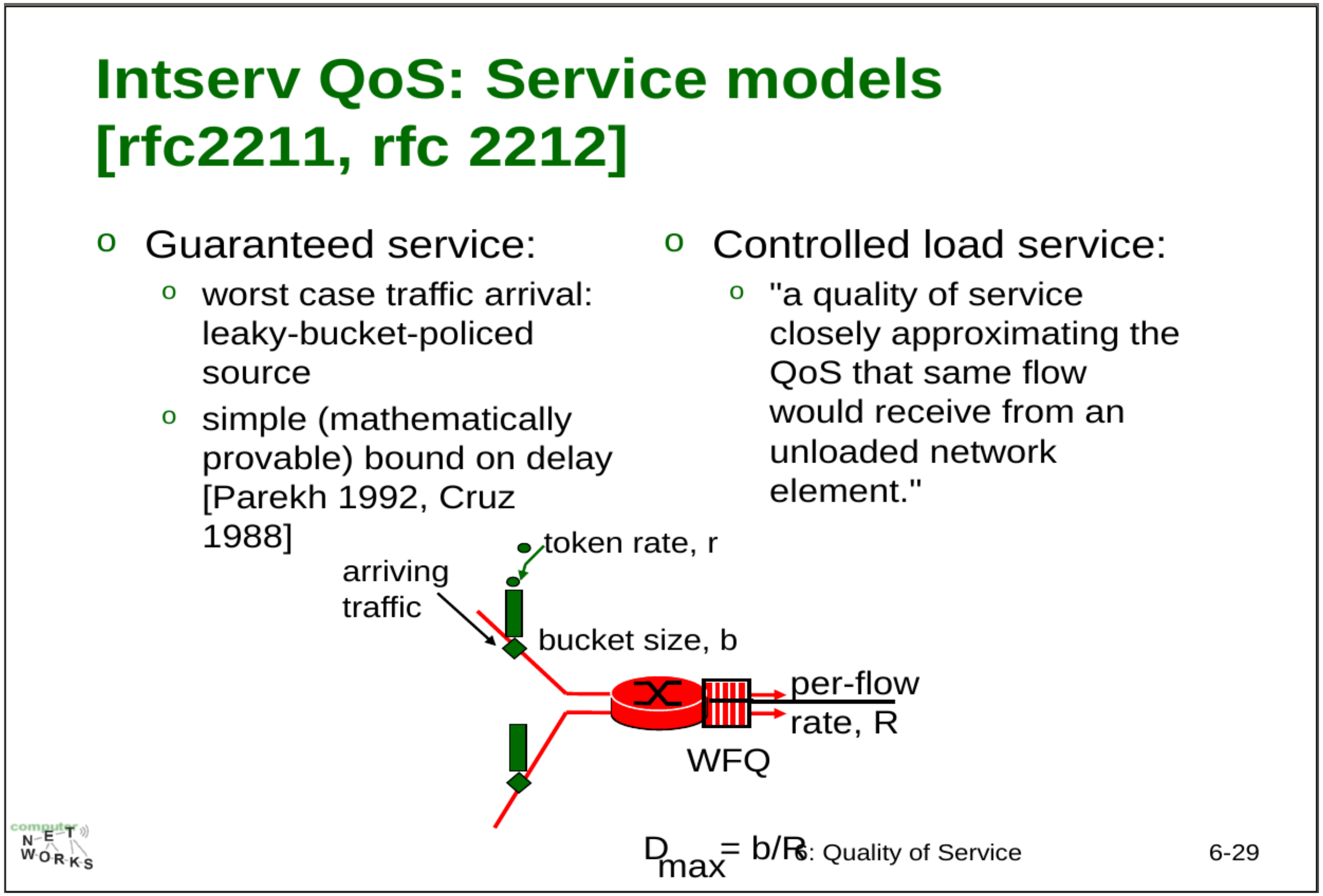
IntServ and DiffServ

- Q: Briefly compare the IntServ to the DiffServ architecture.



IntServ and DiffServ

- Q: Briefly compare the IntServ to the DiffServ architecture.



IntServ and DiffServ

- Q: Briefly compare the IntServ to the DiffServ architecture.

Criteria	DiffServ	IntServ
Based on	Aggregated service classes	Single flow
Level of control	Coarse-grained	Fine-grained
Basic principle	Traffic classification	Resource reservation
States	Edge: per flow Core: none	Every node: per flow
Setup	Per class (long-term)	Per flow
Scalability	Average	Bad

Thank you

Any questions?