b
UNIVERSITÄT
BERN

b UNIVERSITÄT BERN

Advanced Networking and Future Internet

II. Queue Management

Prof. Dr. Torsten Braun, Institut für Informatik

Bern, 21.09.2020

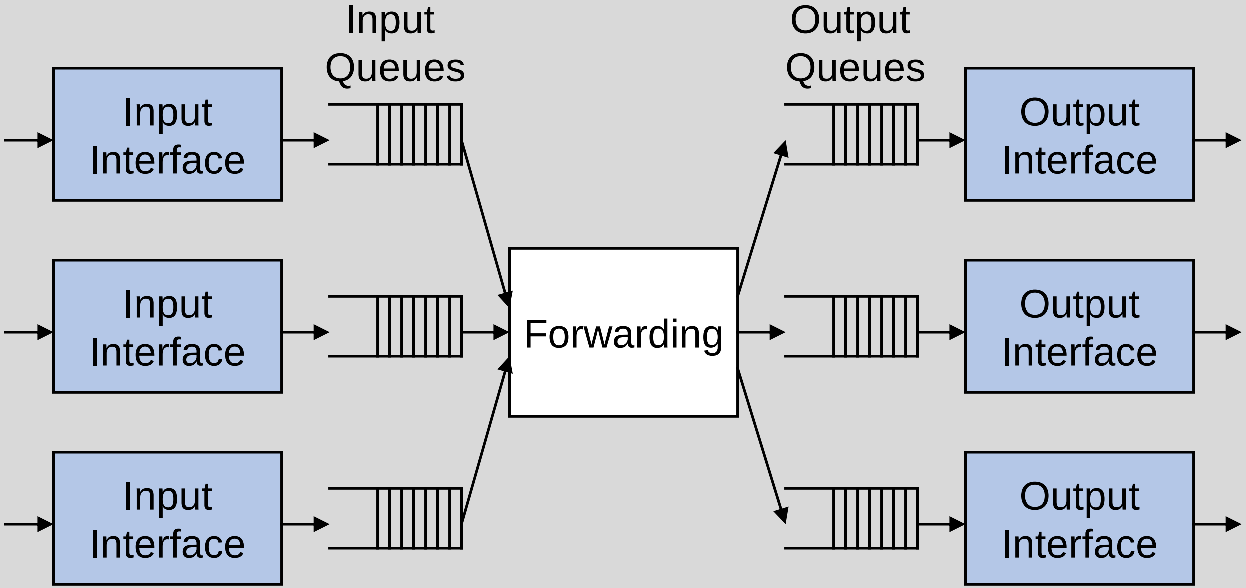
UNIVERSITÄT BERN

Advanced Networking and Future Internet: Introduction

Table of Contents

1. Introduction

1. Router Architecture

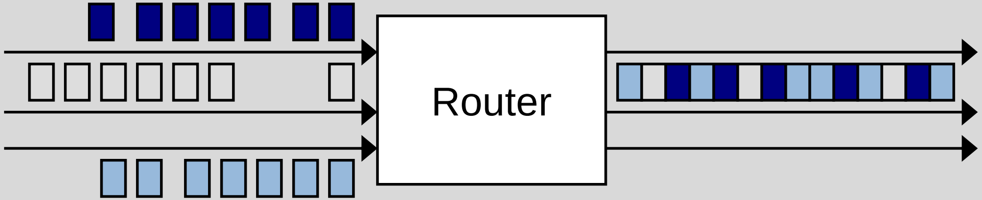


A packet scheduler at the output port must select one packet among those queued for transmission.

b UNIVERSITÄT BERN

1. Introduction

2. Congestion in the Internet



- Congestion in the Internet is still common.
  - Backbones, e.g., transatlantic links
  - Edge routers aggregating traffic
  - Access networks,
    e.g., modem lines, wireless links

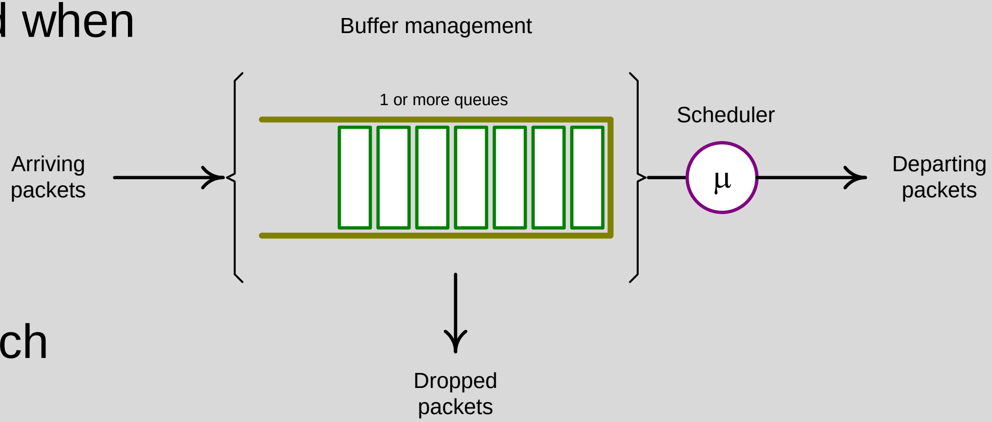
→ Output interfaces might be overloaded.

- Congestion (not transmission errors) is the main reason for packet loss in the Internet.

2. Queue Management

Components

- Scheduling disciplines
  - Which packet to serve next, and when
- Dropping mechanism
  - How and when to drop packets under varying load conditions
- Buffer management
  - How many queues and how much buffer space to use



UNIVERSITÄT BERN

2. Queue Management

1. Why Queue Management in IP Routers?

Best-Effort (BE) applications

– e.g., elastic applications such as e-mail or file transfer that can adapt to available resources
– Resources (bandwidth, buffers etc.) should be shared in a fair way among users / applications.
– Protection of connections from misbehaving connections

Non-BE applications
(some form of QoS guarantee)

– e.g., real-time applications, e.g., audio conferencing, video retrieval, transactions, interactive games, cloud computing (partially), that might require performance bounds (bandwidth, delay, jitter)
– Performance guarantees for applications by resource reservations

2. Queue Management

2. Requirements for Queue Management

- Ease of implementation
- Fairness and Protection (for BE)
  - Fairness: The allocation of a link capacity satisfies the max-min fair share allocation criterion.
  - Protection: Misbehavior of one connection does not affect performance of other connections.
- Performance bounds (for non-BE)
  - Arbitrary connection performance bounds, limited by the conservation law only.
- Ease and efficiency of admission control (for non-BE)
  - Limitation of number of connections

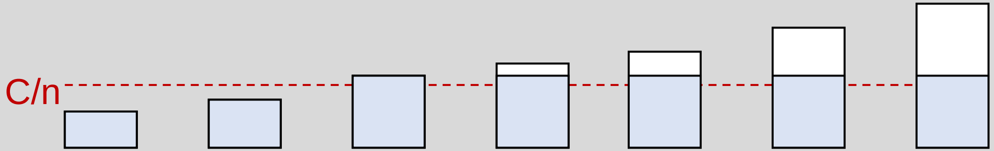
UNIVERSITÄT BERN

2. Queue Management

3. Max-Min Fair Share

Problem

- Dividing resources among a set of users
- Some users demand fewer resources than others.
- How to divide the resources left by these users ?



Max-Min Fair Share Allocation

- Resources are allocated in the order of increasing demand.
- No source gets a resource share larger than its demand.
- Sources with unsatisfied demands get an equal share of the resource.
- Allocation algorithm
  - Divide capacity C by n: C/n resources for each connection
  - Connection 1 needs x_1 resources ($x_1 < C/n$)
  - Distribute exceeding resources $C/n-x_1$ equally among other connections, so that each connection gets $C/n + (C/n - x_1)/(n-1)$ resources allocated
  - Continue process if resource allocation is larger than x_2

UNIVERSITÄT BERN

2. Queue Management

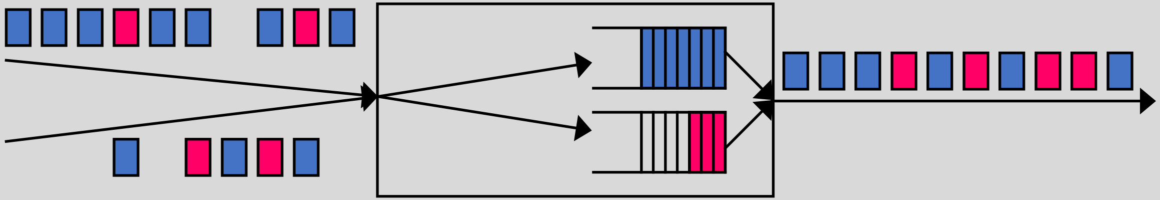
4. FIFO Scheduling

– First In First Out (FIFO) / First Come First Serve (FCFS)
– Packet transmission in the same order as packet arrival
– Packets are dropped if insufficient buffer space.
+ Simple implementation
+ Low packet processing delay

- No service differentiation
  - High-priority and/or delay-sensitive flows may be ‘trapped’ behind generic flows.
- Small packets can undergo large delays if they get behind big packets.
  - In general, flows with larger packets get better service.
- Greedy flows can overload the buffer.
  - Congestion-reactive flows get less and less bandwidth.

2. Queue Management

5. Priority Scheduling



- Priority levels with buffers
- Priority may depend on protocol type, application, IP addresses etc.
- Scheduler selects packet with highest priority.
- Packets with lower priority are selected only if there are no packets with higher priority. → starvation of low priority flows

UNIVERSITÄT BERN

2. Queue Management

6.1 Conservation Law

- A work-conserving scheduler is idle only if its queue is empty.
- FCFS is work-conserving
- Conservation Law:

$$\sum_{i=1}^{N} \rho_i \cdot q_i = const$$

- ρ_i: mean utilization of a link due to connection i
- q_i: connection i's mean waiting time at the queue
- If a particular connection receives a lower delay than with FCFS, it must be at the expense of another connection.

b UNIVERSITÄT BERN

2. Queue Management

6.2 Example: Conservation Law

2 connections A (10 Mbps) and B (25 Mbps) share a 155 Mbps link and each having a mean queuing delay with FCFS: 0.5 ms

New scheduling mechanism reduces A‘s queuing delay to 0.1 ms

- $10 / 155 \cdot 0.5 + 25 / 155 \cdot 0.5 = 10 / 155 \cdot 0.1 + 25 / 155 \cdot B$
- B = 0.66 ms

UNIVERSITÄT BERN

2. Queue Management

7. (Non-)Work-Conserving Scheduling Disciplines

- Non-work-conserving scheduling disciplines may be idle, even if there are packets to serve.
- A packet is only sent if it is eligible. Otherwise it is delayed until it becomes eligible.

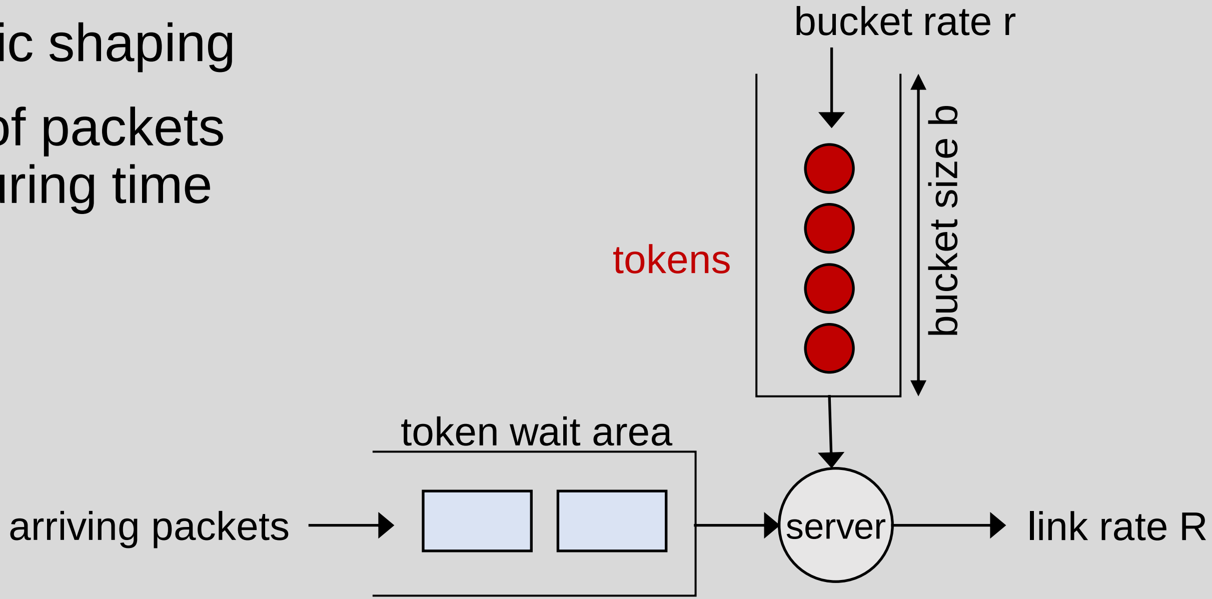
- Are non-work-conserving scheduling disciplines useful ?
  - Downstream traffic can be made more predictable reducing buffer sizes (important in routers) and jitter.
  - Bursts are eliminated and traffic becomes smoother.
    But jitter can also be compensated by playback buffers in end systems!
- Today‘s routers implement work-conserving schedulers mainly.

b UNIVERSITÄT BERN

2. Queue Management

8. Token Bucket

- Mechanism for traffic shaping
- Maximum number of packets entering network during time interval t: $r \cdot t + b$



b UNIVERSITÄT BERN

3. Scheduling Disciplines

For Best-Effort

- Generalized Processor Sharing (GPS)
- Round-Robin (RR)
- Weighted Round-Robin (WRR)
- Deficit Round-Robin (DRR)
- Weighted Fair Queuing (WFQ)

For Guaranteed Service

- Weighted Fair Queuing (WFQ)
- Delay Earliest Due Date (EDD)
- Jitter Earliest Due Date (J-EDD)
- Rate-Controlled Scheduling

(non-work-conserving)

UNIVERSITÄT BERN

3. Scheduling Disciplines

1. Generalized Processor Sharing

– serves packets as if they are in separate logical queues.

– visits each non-empty queue in turn.

– serves an infinitesimally small amount of data in each queue so that it can visit each queue at least once in any finite time interval.

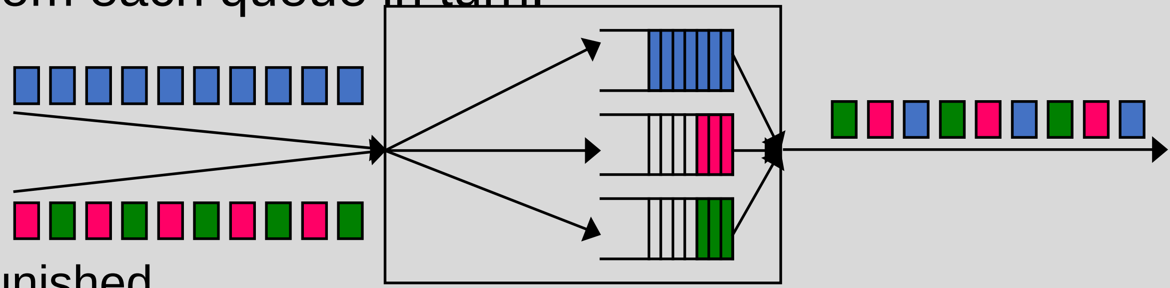
Characteristics

+ Fair
+ Ideal
- Not implementable
- Work-conserving

3. Scheduling Disciplines

2. (Weighted) Round Robin

- Several queues per output, e.g., 1 queue per sender or flow
- RR: Packets are taken from each queue in turn.
- WRR: Queues have different weights.



- Characteristics
  - \+ Aggressive flows get punished.
  - – Mean packet sizes must be known in advance.
  - – Fairness over a time interval much longer than a round time

3. Scheduling Disciplines

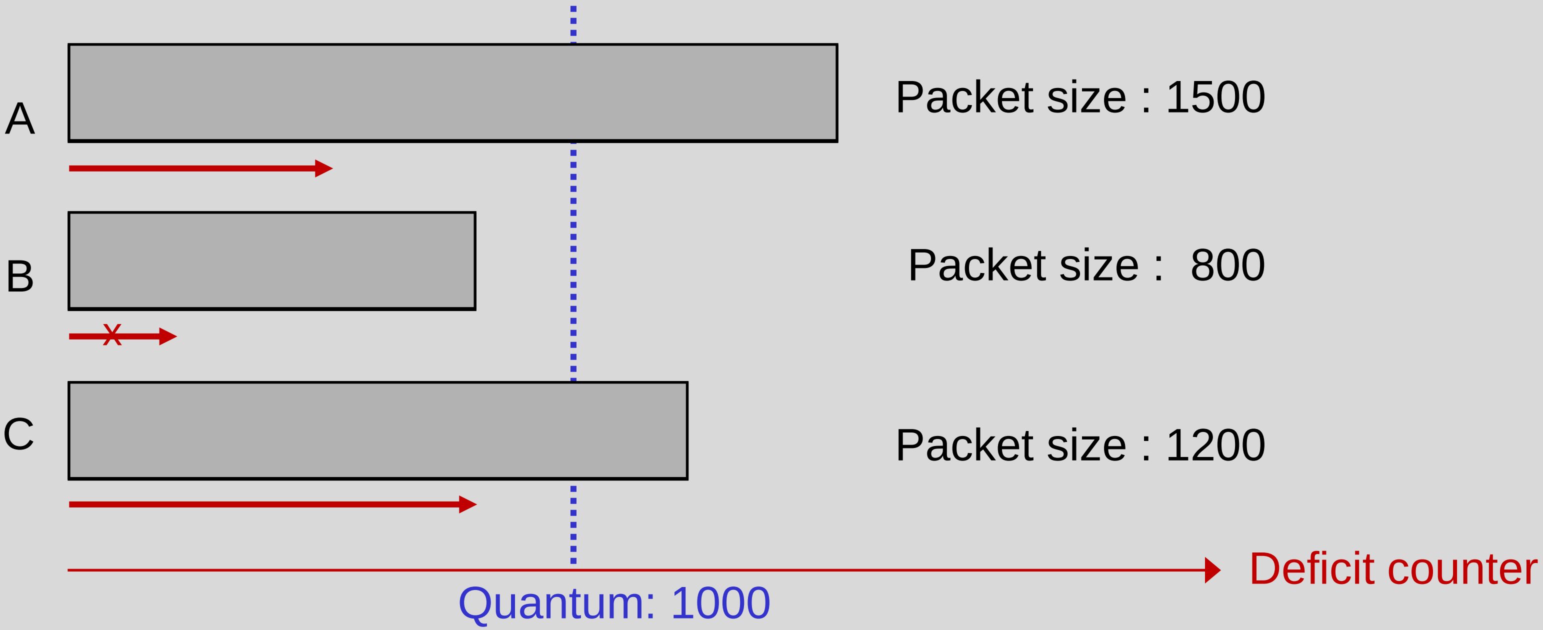
3.1 Deficit Round Robin

- DRR supports variable packet sizes.
- Each queue has a deficit counter initialized by 0.
- Scheduler visits each queue in turn and tries to serve 1 quantum of bits.

```
if (there is a packet to be served)
   if (packet_length <
   (quantum + deficit_counter)){
          serve packet;
          deficit_counter +=
              (quantum - packet_length)}
   else
          deficit_counter += quantum;
else
   deficit_counter := 0;
```

3. Scheduling Disciplines

3.2 Example: DRR



3. Scheduling Disciplines

4. Weighted Fair Queuing

Approximation of GPS

- emulates a weighted bit-by-bit round-robin discipline, i.e., generalised processor sharing
- Virtual starting and finishing ‘times’ are calculated as if GPS would be used.
- The next packet to be served is the one with the smallest finishing time (at the time of calculation).

Implementable GPS version providing different amounts of capacity to different flows.

- WFQ closely approximates the properties of GPS.
- WFQ provides uniform and appropriate level of service to each flow.

3. Scheduling Disciplines

4.1 WFQ: Finish Number

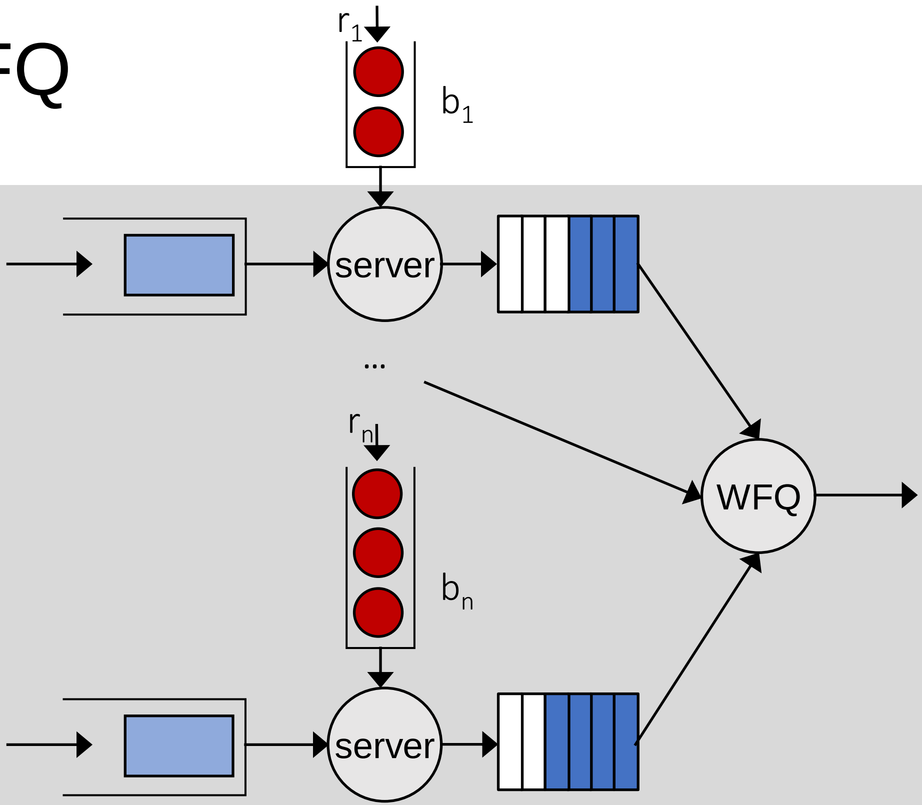
- Calculation of a finish number ($\neq$ finish time !) indicating the order, in which packets are served assuming a bit-by-bit service
- Round number = number of rounds a bit-by-bit RR scheduler has completed at a given time (variable duration of rounds !)
- A connection is active if (largest finish number in the connection‘s queue or of the packet last served) > current round number

- Finish number
  - of a packet arriving at an inactive connection = current round number + packet size [bits]
  - of a packet arriving at an active connection = largest finish number in the queue or of the packet last served + packet size [bits]
  - $F(i, k, t) = \max \{F(i, k - 1, t'), R(t)\} + P(i , k, t) / \phi(i)$
    - $F(i, k, t)$: finish number of k^{th} packet of connection i arriving at time t
    - $P(i, k, t)$: packet length of k^{th} packet of connection i arriving at time t
    - $R(t)$: number of the round at time t
    - $\phi(i)$: weight of connection i
- Packets are ordered by their finish number and are served in that order.

b UNIVERSITÄT BERN

3. Scheduling Disciplines

4.2 Token Buckets and WFQ



UNIVERSITÄT BERN

3. Scheduling Disciplines

4.3 WFQ: Delay Bound

Minimum throughput for flow i by WFQ = $R \cdot \phi(i) / \Sigma\, \phi(j)$

Maximum delay for flow i?

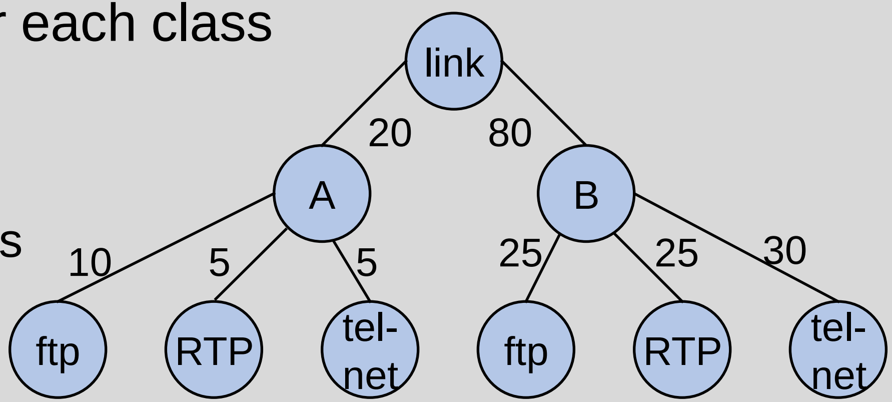
- Assumption: token bucket i is full.
- Then, burst of b_i packets arrives and consumes all tokens.
- b_i packets are served with rate of at least $R \cdot \phi(i) / \Sigma\, \phi(j)$.
- Last packet will experience maximum delay = $b_i / (R \cdot \phi(i) / \Sigma\, \phi(j))$.

UNIVERSITÄT BERN

3. Scheduling Disciplines

5. Class-Based Queuing

- Hierarchical Link Sharing
- Implementation by WFQ, DRR
- Definition of bandwidth available for each class
- Class =
  - Protocol / application
  - Traffic from / to certain sub-networks



3. Scheduling Disciplines

6. Earliest Due Date

- Assignment of a deadline (= arrival time + period of traffic stream) to each packet
- Scheduler selects packets in the order of their deadlines.
- Packets missing their deadline are discarded.

b
UNIVERSITÄT
BERN

3. Scheduling Disciplines

6.1 Delay-EDD

- Negotiation of a service contract (in addition to EDD): If a source obeys a peak rate, the delay is less than some delay bound.
- Deadline =
  arrival time + delay bound

- Delay-EDD requires
  - admission control and
  - bandwidth reservation at peak rate.
- Example
  - One packet arrives every 0.2 s with a delay bound of 1 s
  - Deadline of k^{th} packet = (0.2k+1) s

UNIVERSITÄT BERN

3. Scheduling Disciplines

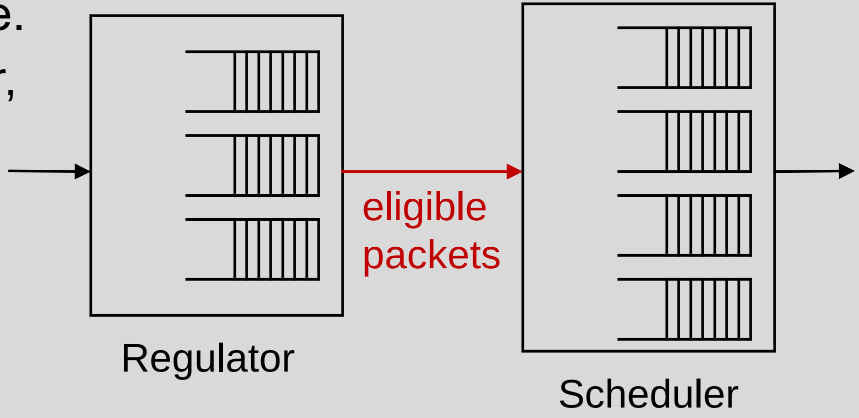
6.2 Jitter-EDD

- Extension of Delay-EDD: Delay-jitter-regulator precedes EDD scheduler.
- Packets get timestamp = deadline - finishing time.
- Next server delays packet by timestamp value.
- All packets receive the same delay at every hop (except at the last hop).

3. Scheduling Disciplines

7. Rate-Controlled Scheduling

- can provide bandwidth and delay bounds to connections
- Regulator
  - determines the packet‘s eligibility time.
  - forwards eligible packets to scheduler, e.g., FIFO, WFQ, EDD.

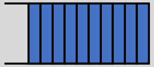


4. Packet Dropping

1. Issues

Limited queue length requires packet dropping by scheduler in overload conditions.

- Degree of aggregation
  - Protection versus state information
- Marking of several drop priorities
  - Packets with high drop priorities are dropped first.
  - Either applications or routers must be able to mark packets.
  - Low dropping priority for packets that traveled very long



- Late / early dropping
  - Early drop: Packets are dropped even in case of non-full queues. Cooperative sources experience low packet loss.
- Drop position
  - Tail: easy to implement, may be unfair
  - Head: more complex, sources notice dropping earlier
  - Random: very complex, fair

UNIVERSITÄT BERN

4. Packet Dropping

2. Late and Early Dropping

Late Dropping

- Arriving packets are discarded, if queue is full.
- Certain flows can monopolize the queue.
- Higher delays
- Queues can not absorb bursts.
- Reaction onto congestion, but no congestion avoidance
- Global synchronisation
  - When queue is full, all packets from all flows are dropped.
  - All flows back-off simultaneously.
  - Periods of congestion followed by periods of low network utilisation

Early Dropping

- Flows (e.g., TCP connection) reduce rate before queue becomes overloaded.
- Examples
  - Early Random Drop:
    Arriving packet is dropped with fixed probability, if queue level exceeds a certain limit.
  - Random Early Detection

UNIVERSITÄT BERN

4. Packet Dropping

3.1 Random Early Detection

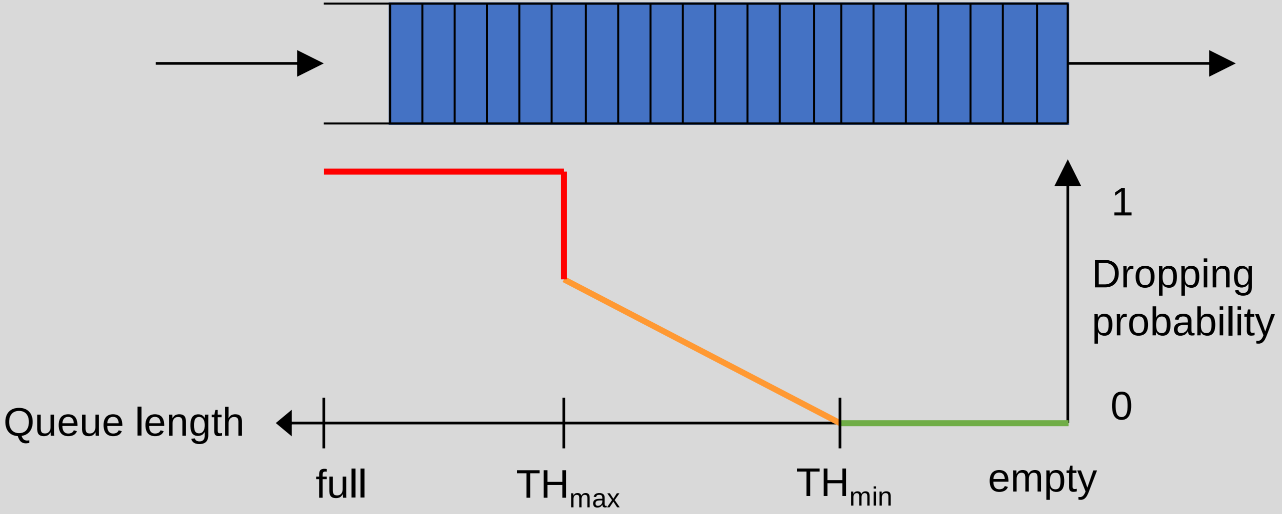
```
avg := exponential average of queue length;
if (avg < TH_min)
   store packet in queue
else if (TH_min ≤ avg < TH_max){
   P := calculate dropping probability;
   discard packet with probability P;
   store packet with probability 1-P}
else if (avg ≥ TH_max)
   discard packet;
```

b UNIVERSITÄT BERN

4. Packet Dropping

3.2 Random Early Detection

Dropping probability increases with the exponential average queue length.



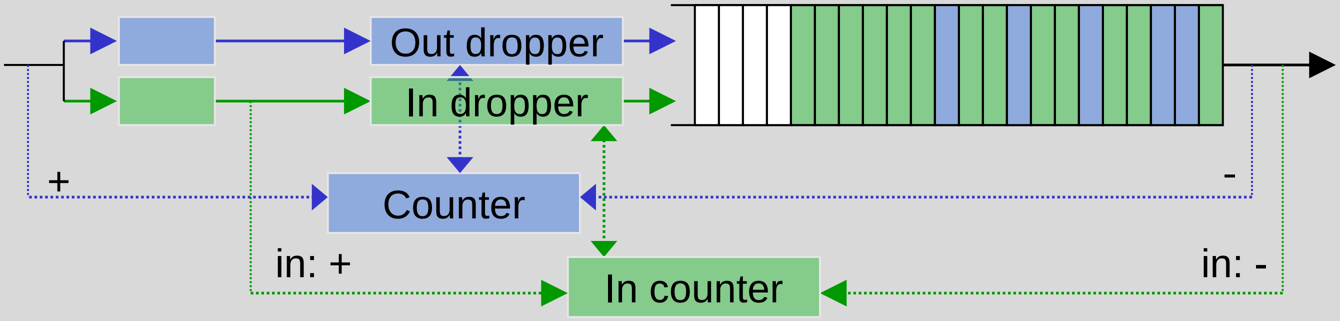
4. Packet Dropping

4.1 RED with In and Out (RIO)

Different droppers for high / low priority (in-service / out-of-service) packets

– Out dropper drops earlier and more aggressively.

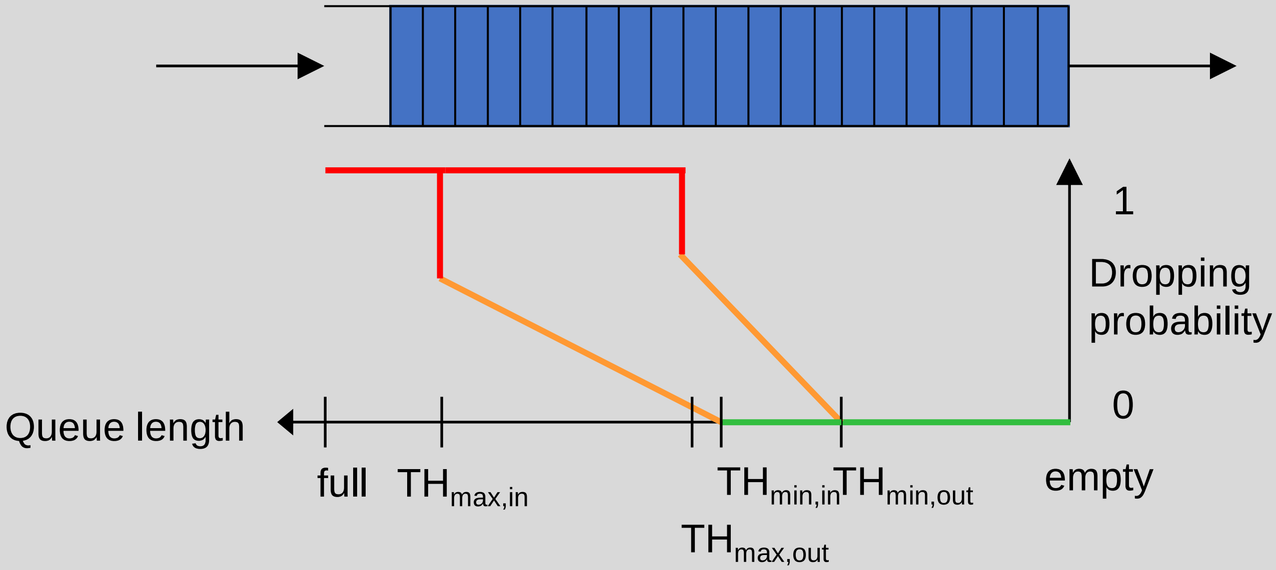
– In dropper considers high-priority packets only.



u^b UNIVERSITÄT BERN

4. Packet Dropping

4.2 RIO



Thanks

for Your Attention

UNIVERSITÄT BERN

Prof. Dr. Torsten Braun, Institut für Informatik

Bern, 21.09.2020