Advanced Computer Networks

QoS in IP networks

Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

Contents

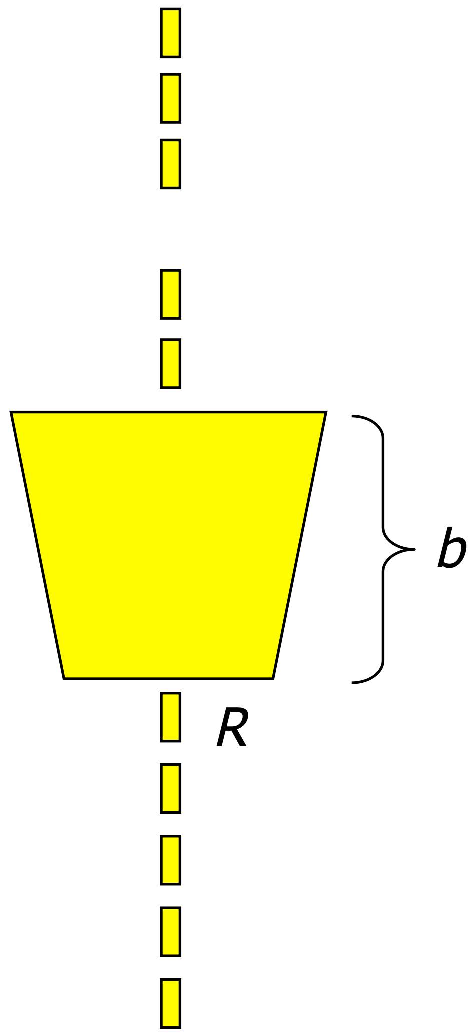
- Traffic shaping
  - leaky bucket
  - token bucket
- Scheduling
  - FIFO
  - Fair queueing
  - RED

Traffic shaping and QoS contract

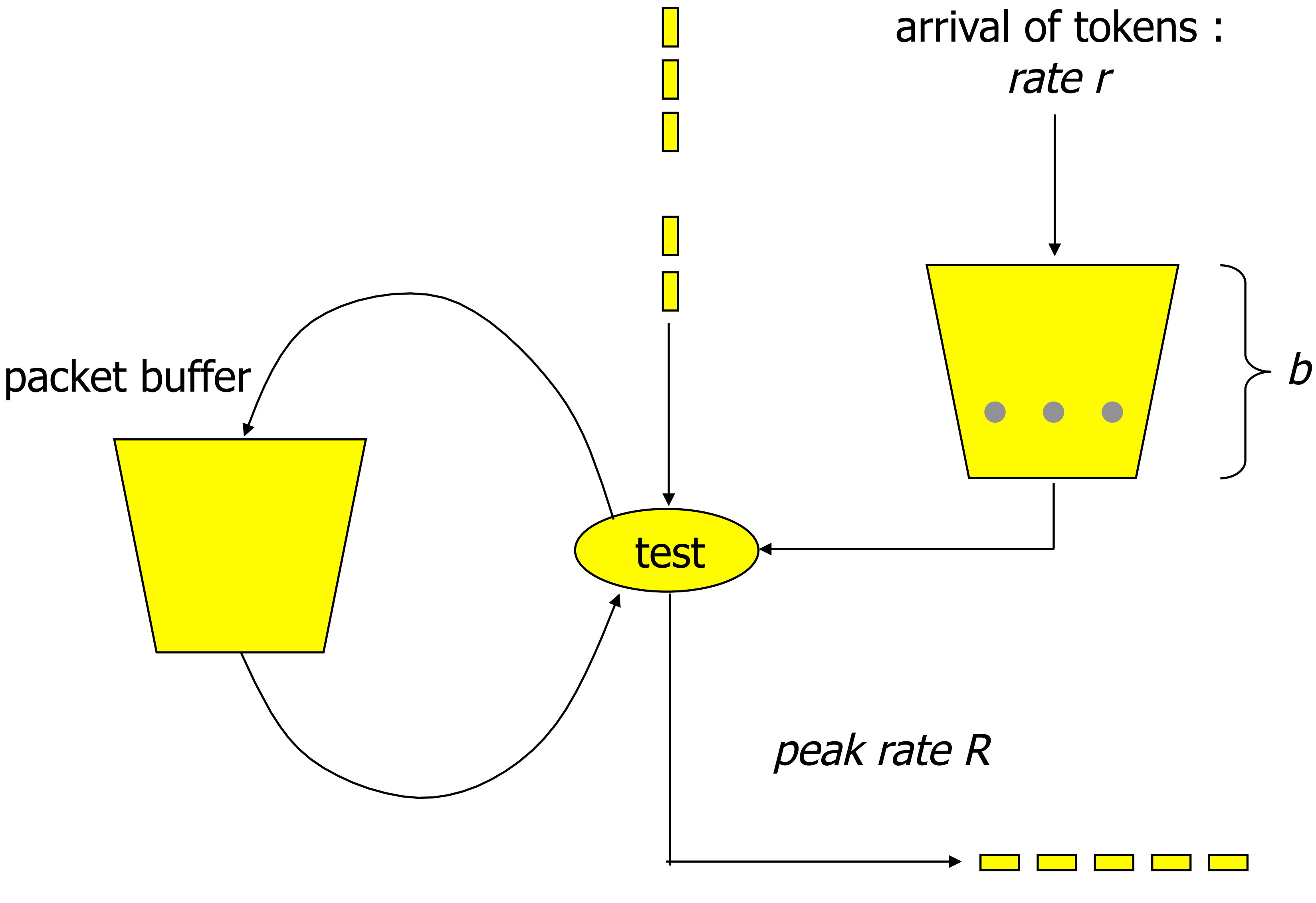
- Guaranteeing QoS
  - if the network knows the type of the traffic, it can reserve resources to support the traffic
- QoS contract between the source and the network
  - source: traffic description - leaky bucket, token bucket
  - network: QoS guarantee if the traffic conforms to the description
  - if the traffic is not conformant (leaky bucket, token bucket), penalty: reject a packet, no guarantees of the QoS *(traffic policing)*

Leaky bucket

- Limited size buffer with constant departure rate
  - *R* if buffer not empty
  - 0 if buffer empty
- Fixed size packets
  - one packet per clock tick
- Variable size packets
  - number of bytes per clock tick
- Packet loss if buffer filled

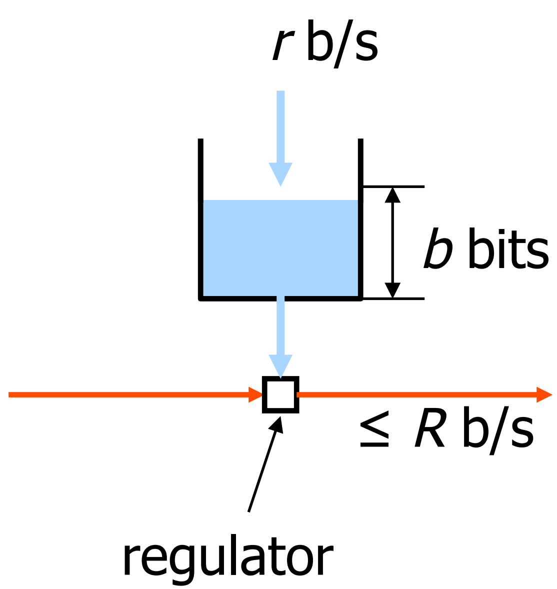


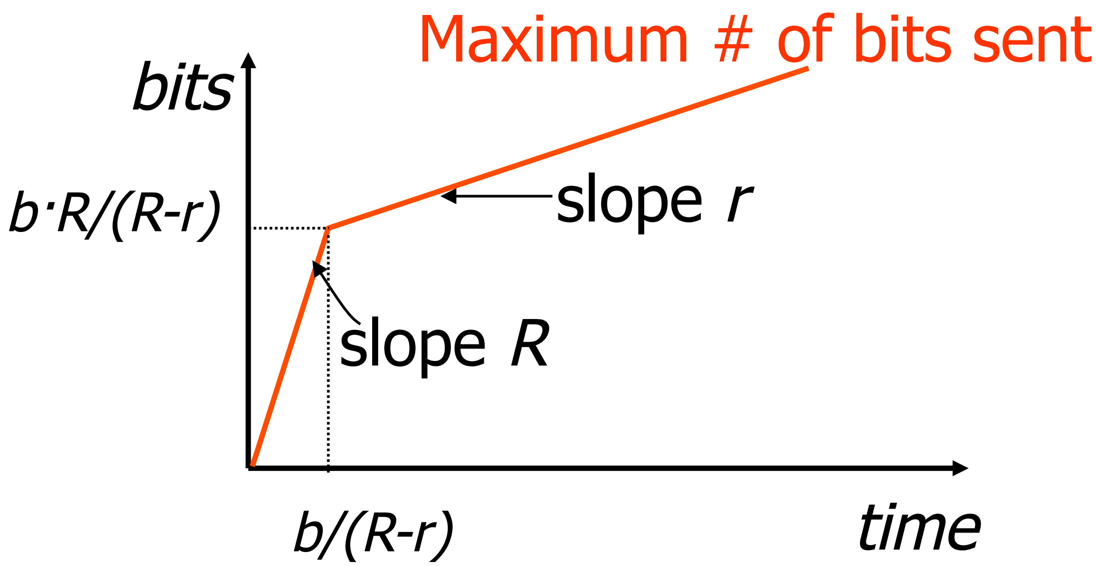
Token bucket



Characterizing Burstiness: Token Bucket

- 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
- A bit (packet) can be transmitted only when a token is available





Token bucket

- Tokens generated with rate *r*
  - 1 token : 1 packet or *k* bytes
- Packet must wait for a token before transmission
  - no losses
  - allows limited bursts (a little bit more than *b*)
- When packets are not generated, tokens accumulate
  - *n* tokens - burst of *n* packets
  - if bucket filled, tokens are lost
- Mean departure rate: *r*
- Delay limited by *b*/*r* (Little's formulae)

Example

- 25 MB/s link
- Network can support a peak rate R = 25 MB/s, but prefers sustained throughput of r = 2 MB/s
- Data generated
  - 1 MB each second, burst during 40 ms
- Example
  1. leaky bucket with b = 1 MB, R = 25 MB/s, r = 2 MB/s
  2. token bucket with b = 250 KB, R = 25 MB/s, r = 2 MB/s
  3. token bucket with b = 500 KB, R = 25 MB/s, r = 2 MB/s
  4. token bucket with b = 750 KB, R = 25 MB/s, r = 2 MB/s
  5. token bucket with b = 500 KB, R = 25 MB/s, r = 2 MB/s and leaky bucket with b = 1 MB, R = 10 MB/s

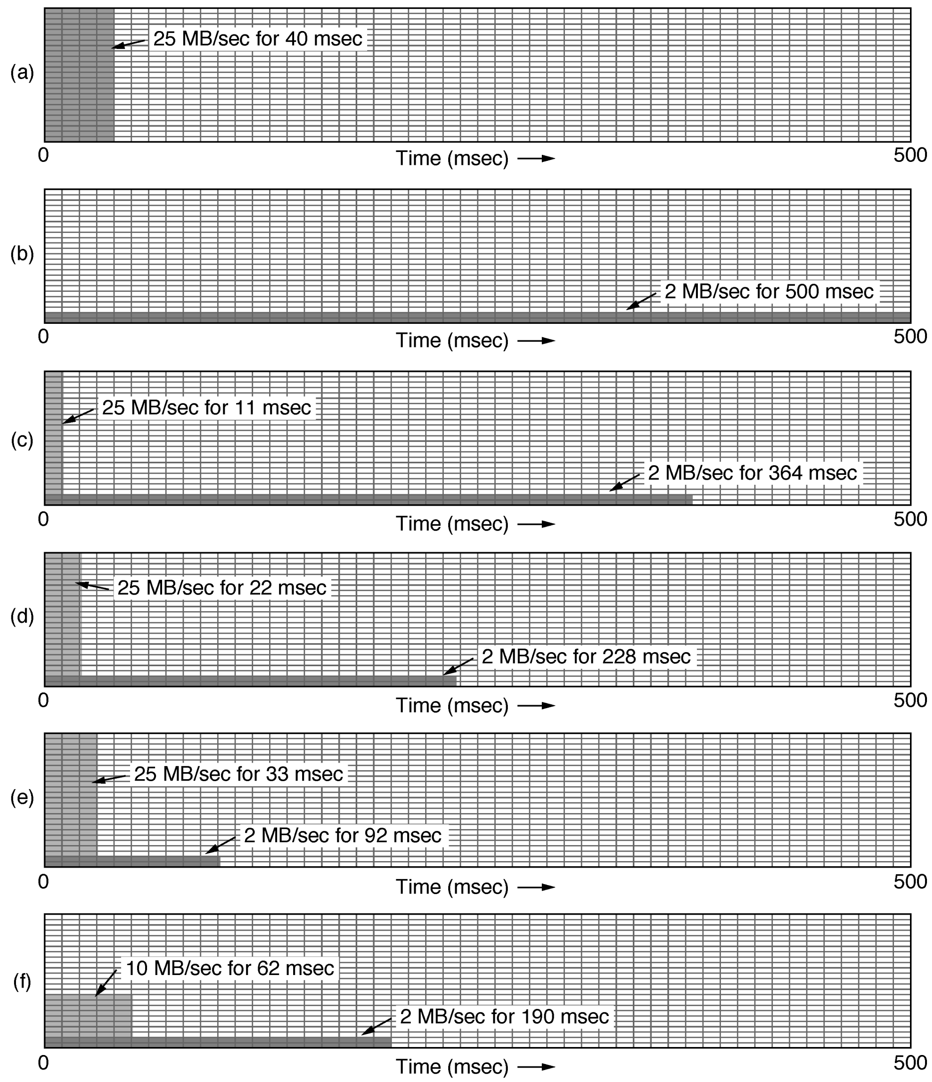


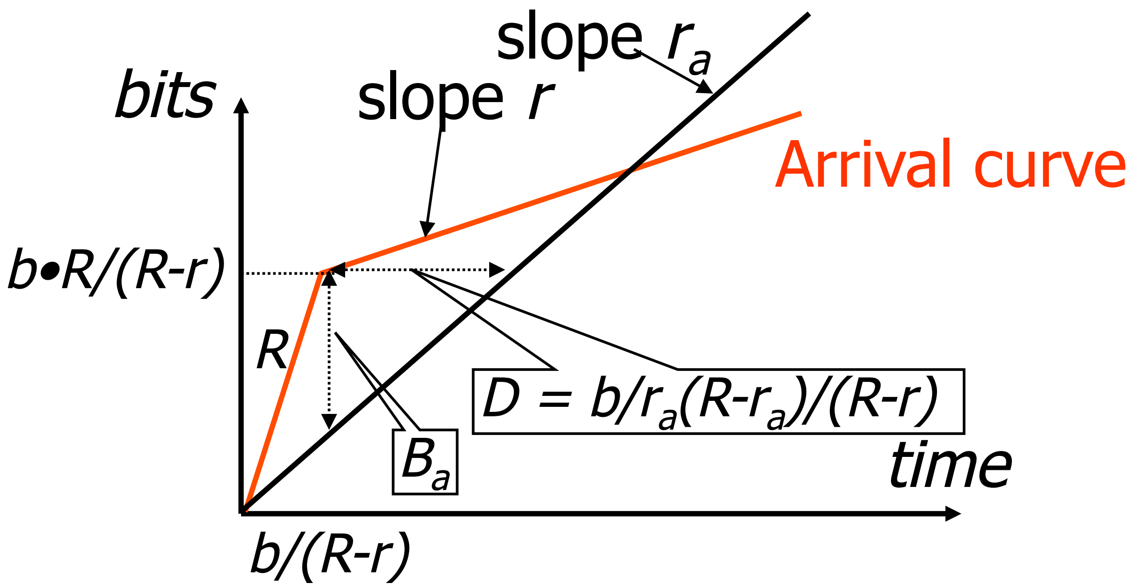
Fig. 5-25. (a) Input to a leaky bucket. (b) Output from a leaky bucket. (c) - (e) Output from a token bucket with capacities of 250KB, 500KB, and 750KB. (f) Output from a 500KB token bucket feeding a 10 MB/sec leaky bucket.

Burst duration

- Burst duration - S [s]
- Size of the bucket - b bits
- Maximal departure rate - R b/s
- Token arrival rate - r b/s
  - burst of $b + rS$ bits
  - burst of RS
  - $b + rS = RS \rightarrow S = b/(R - r)$
- Example
  - b = 250 KB, R = 25 MB/s, r = 2 MB/s
  - S = 11 ms

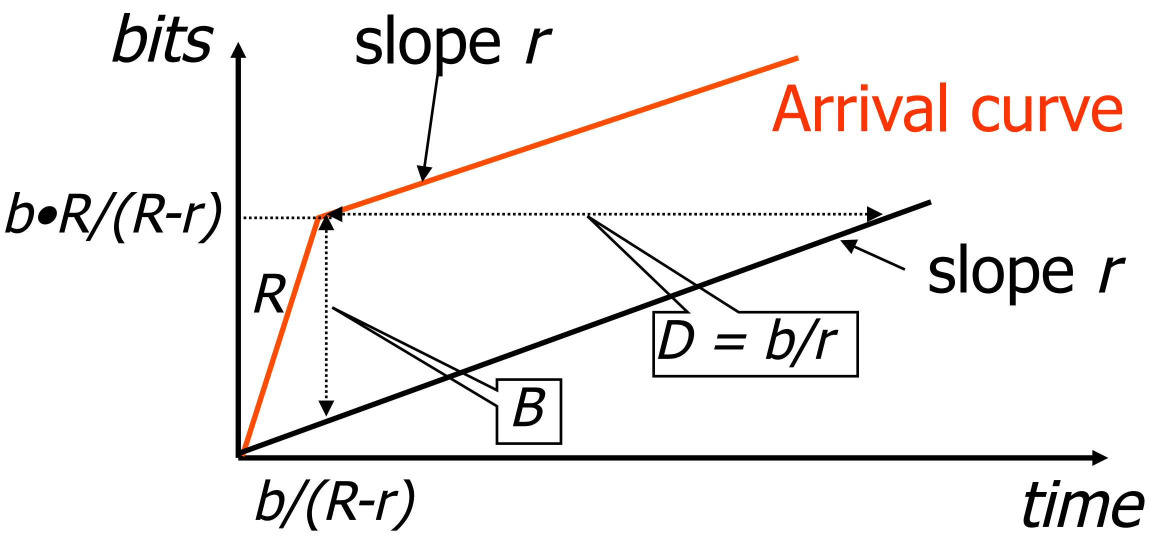
QoS Guarantees: Per-hop Reservation

- End-host: specify
  - arrival rate characterized by token bucket with parameters *(b, r, R)*
  - the maximum tolerable delay *D*, no losses
- Router: allocate bandwidth r_a, buffer space B_a such that
  - no packet is dropped
  - no packet experiences a delay larger than *D*

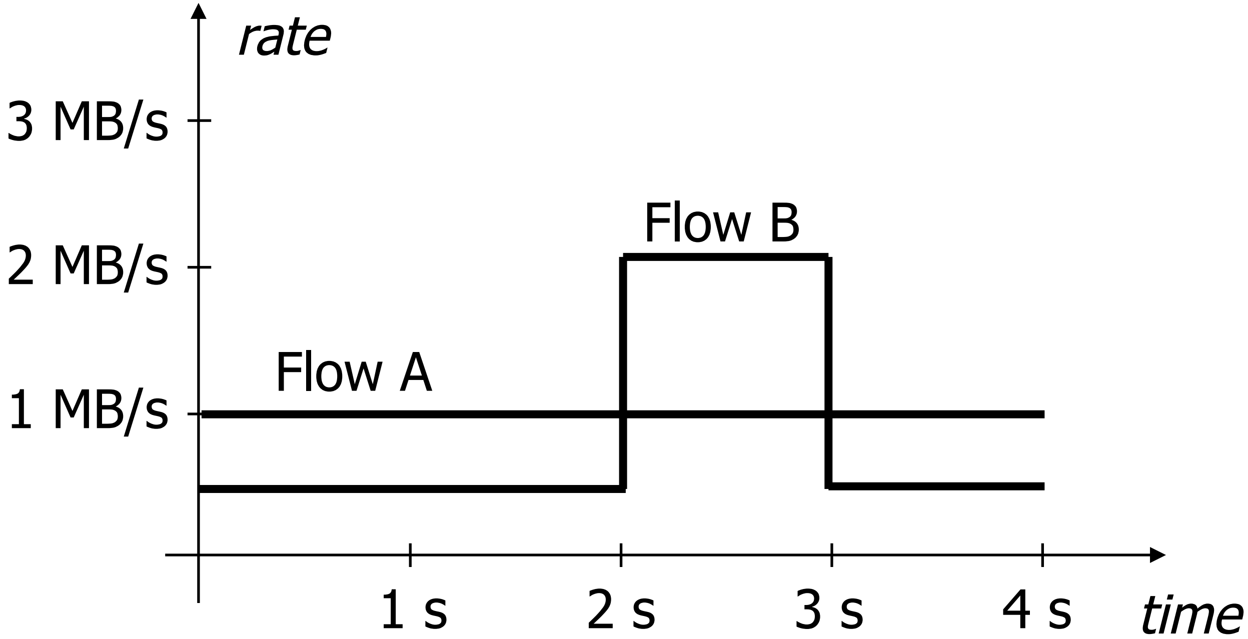


QoS Guarantees: Per-hop Reservation

- Router: if allocated bandwidth $r_a = r$, buffer space *B* such that
  - no packet is dropped
  - no packet experiences a delay larger than $D = b/r$

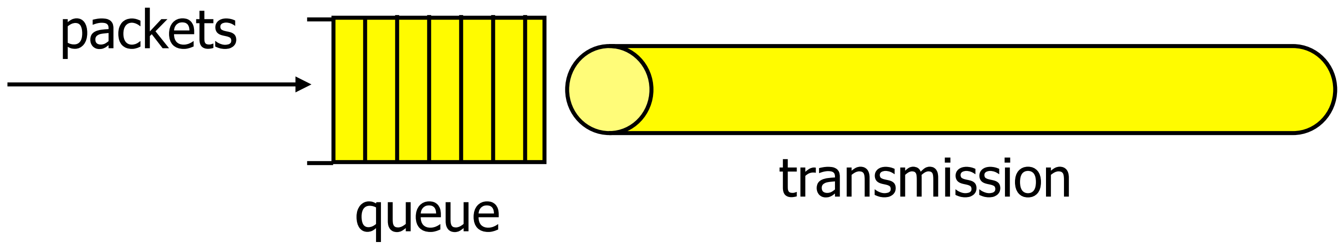


Traffic description



- Flow A : r = 1 MB/s, b = 1 B
- Flow B : r = 1 MB/s, b = 1 MB
  - during 2 s, the flow saves 2 s at 0.5 MB/s = 1 MB

Scheduling strategies



- Scheduler
  - defines the order of packet transmission
- Allocation strategy
  - throughput
    - which packet to choose for transmission
    - when chosen, packet benefits from a given throughput
  - buffers
    - which packet to drop, when no buffers

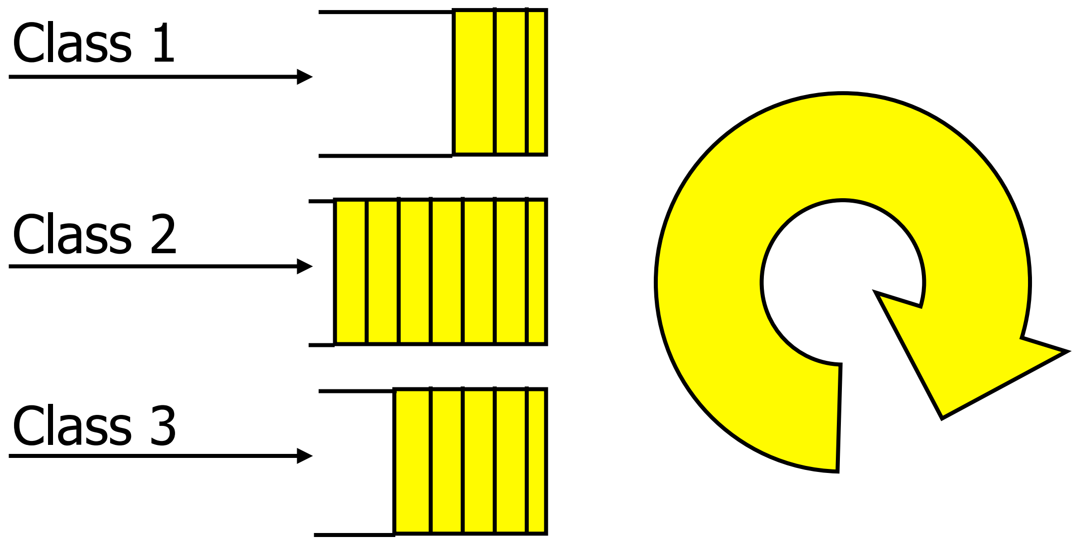
FIFO

- Current state of Internet routers
- Allows to share bandwidth
  - proportionally to the offered load
- No isolation
  - elastic flows (rate controlled by the source eg. TCP) may suffer from other flows
    - a greedy UDP flow may obtain an important part of the capacity
    - real time flows may suffer from long delays
- Last packets are dropped - tail drop
  - TCP adapt bandwidth based on losses
- RED (Random Early Detection) techniques
  - choose a packet randomly before congestion and drop it

Priority Queue

- Several queues of different priority
  - source may mark paquets with priority
    - eg. ToS field of IP
  - packets of the same priority served FIFO
  - non-preemptive
- Problems
  - starvation - high priority source prevents less priority sources from transmitting
  - TOS field in IP - 3 bits of priority
  - how to avoid everybody sending high priority packets?

Class Based Queueing (CBQ)

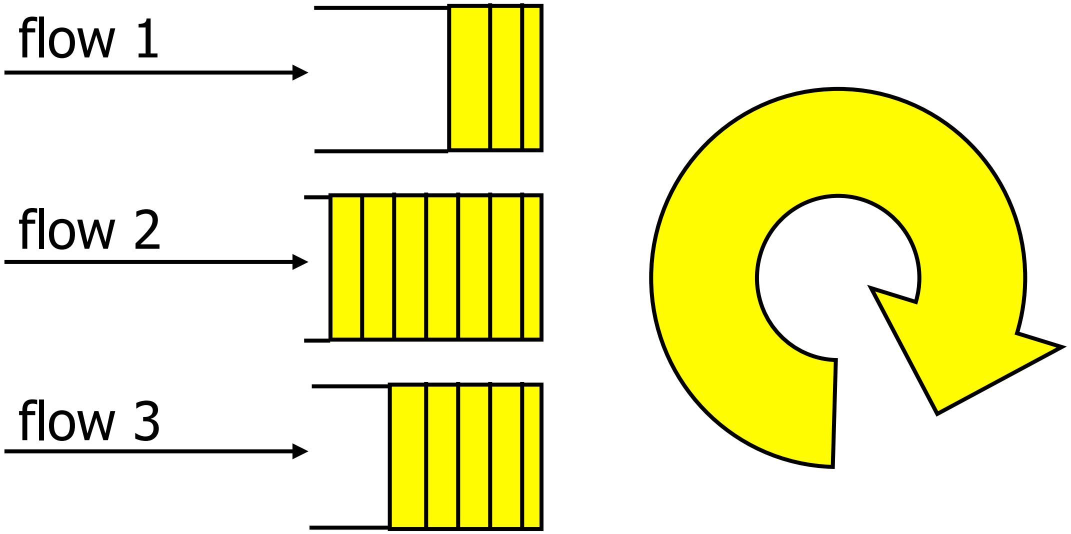


- Also called Custom Queueing (CISCO)
- Each queue serviced in round-robin order
- Dequeue a configured byte count from each queue in each cycle
- Each class obtains a configured proportion of link capacity

Characteristics

- Limited number of queues (CISCO - 16)
- Link sharing for Classes of Service (CoS)
  - based on protocols, addresses, ports
- Method for service differentiation
  - assign different proportions of capacity to different classes
  - not so drastic as Priority Queueing
- Avoids starvation

Per Flow Round Robin

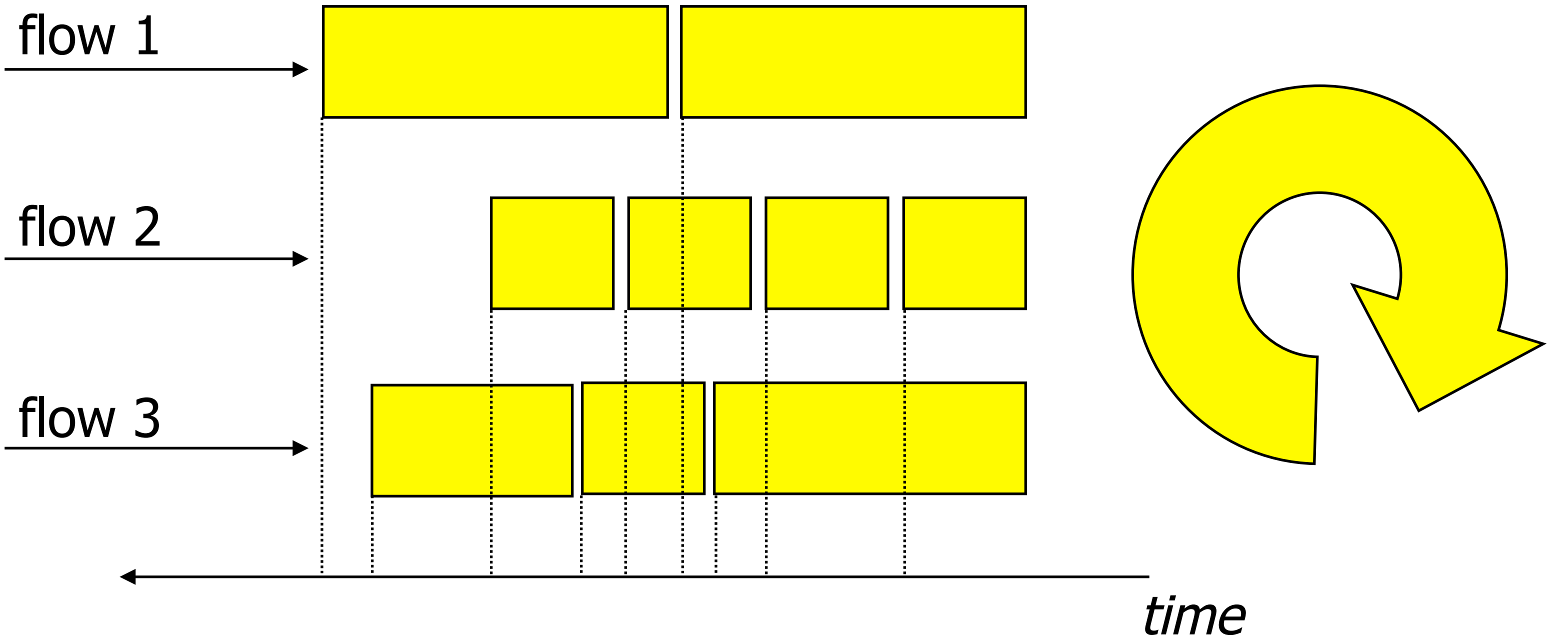


- Similar to Processor Sharing or Time Sharing
  - one queue per flow
  - cyclic service, one packet at a time

Characteristics

- It modifies the optimal strategy of sources
  - FIFO: be greedy - send as much as possible
  - RR: use your part the best
    - a greedy source will experience high delays and losses
- Isolation
  - good sources protected from bad ones
- Problems
  - flows sending large packets get more
  - cost of flow classification

Fair Queueing



- Round robin "bit per bit"
  - each packet marked with the transmission instant of the last bit
  - served in the order of the instants

Weighted Fair Queueing

- Fair queueing
  - equal parts : $1/n$
- Weighted fair queueing
  - each flow may send different number of bits
- Example - weights w_i

flow 1	flow 2	flow 3
1/3	1/6	1/2

$x_i = C\, w_i$, C: link capacity

Rate guarantee

- Weights expressed as proportions (w_i - guaranteed weight)
  - If no packets of a given flow, unused capacity shared equally by other flows

$$x_i >= C\, w_i$$

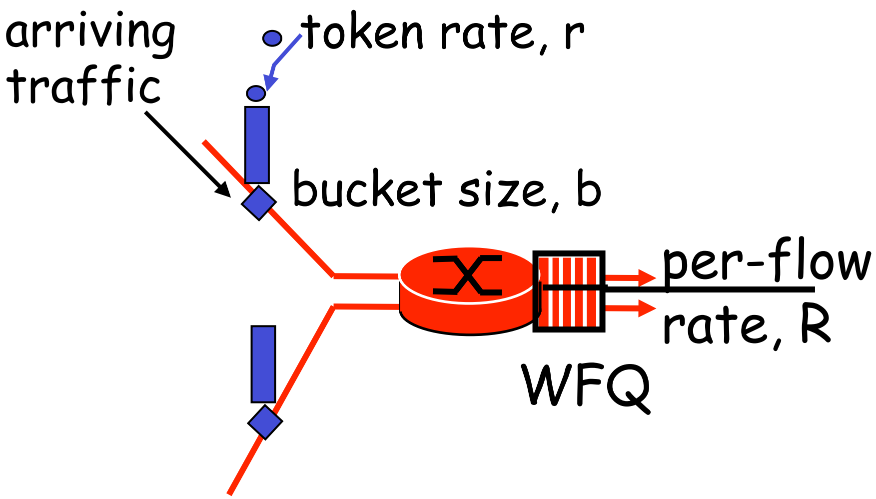
  - Weights to guarantee a given rate

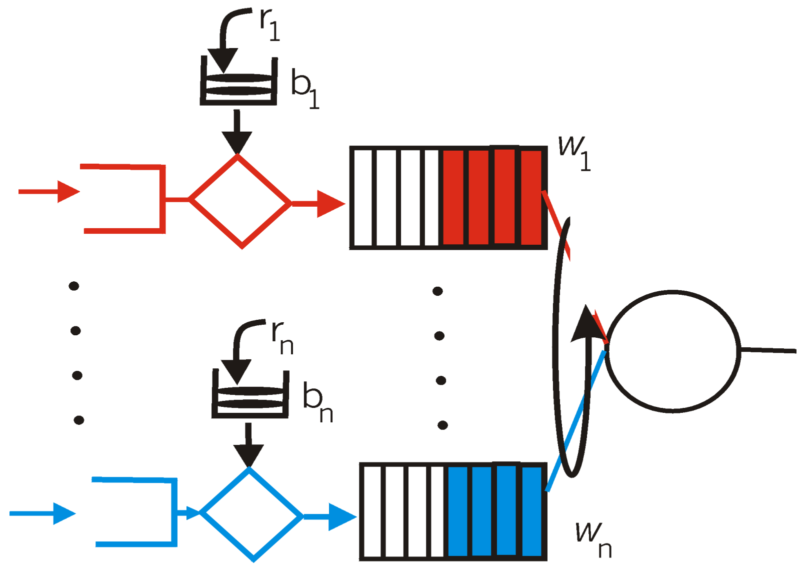
$$w_i = x_i / C$$

Delay guarantee

- Flow constrained by a token bucket
  - rate *r*, buffer of *b*
  - delay limited by *b/r*
- If $r_i >= r$ (the rate obtained is sufficient for the flow)
  - delay limited by b/r_i
  - total delay limited by the sum of all delays

Delay guarantee



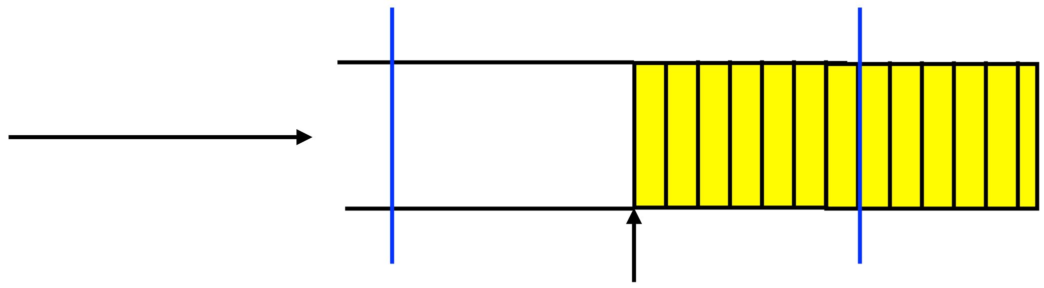


$$D_{max} = b/R$$

Random Early Detection

- Family of techniques used to detect congestion and notify sources
  - when a queue is saturated, packets are dropped
  - losses interpreted as congestion signals → decrease rate
- Idea
  - act before congestion and reduce the rate of sources
  - threshold for starting to drop packets
- Losses are inefficient
  - result in retransmissions, dropped packets should be retransmitted - enter Slow Start
- Synchronization of TCP sources
  - several packets dropped
  - several sources detect congestion and enter slow start at the same instant

RED



- Estimation of the average queue length
  - **`average ← q × measure + (1 - q) × average`**
- If **`average ≤ th-min`**
  - accept the packet
- If **`th-min < average < th-max`**
  - drop with probability **`p`**
- If **`th-max ≤ average`**
  - drop the packet

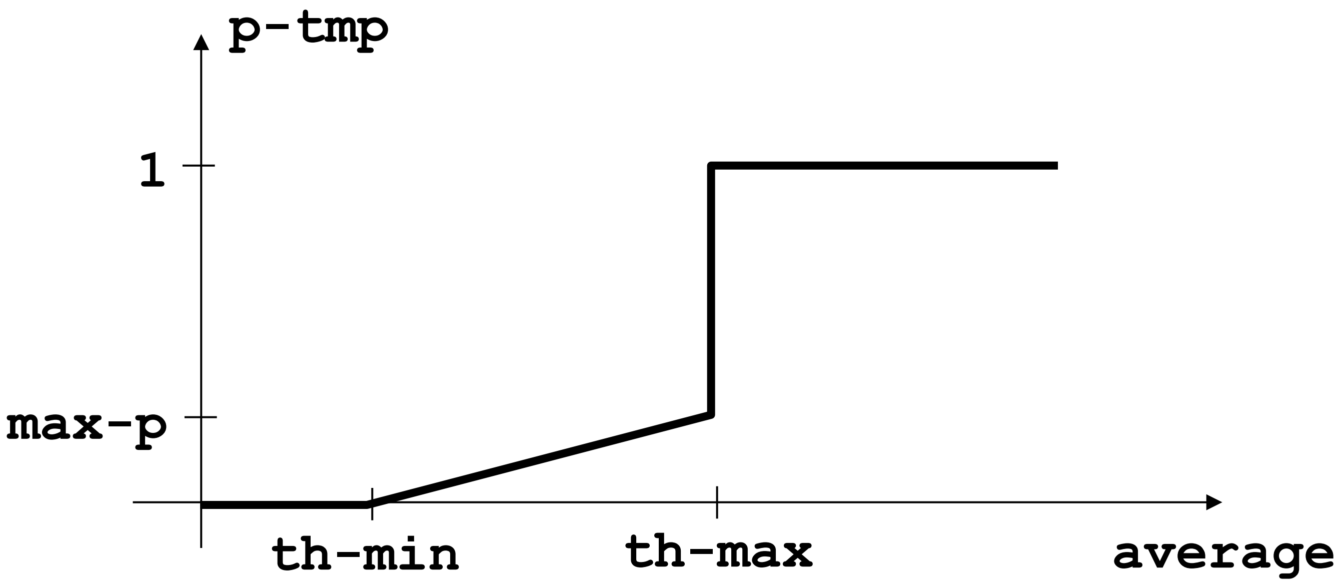
RED Characteristics

- Tends to keep the queue reasonably short
  - low delay
- Suitable for TCP
  - single loss recovered by Fast Retransmit
- Probability **p** of choosing a given flow is proportional to the rate of the flow
  - more packets of that flow, higher probability of choosing one of its packet

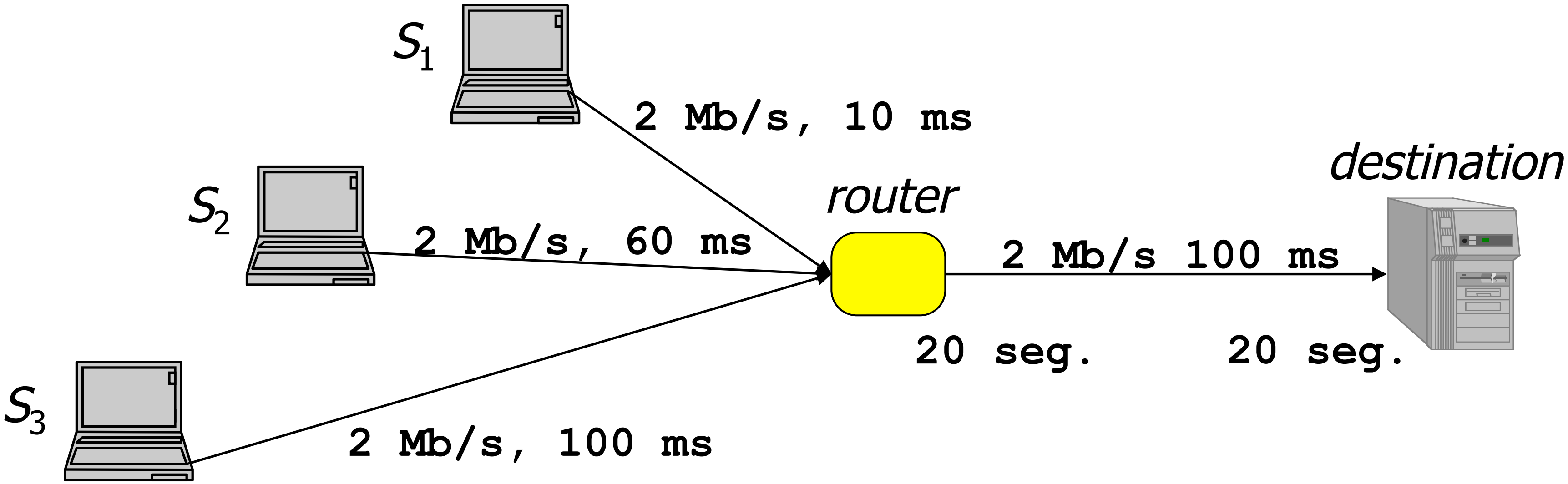
RED Characteristics

- Dynamic probability **`p`**
  - **`p-tmp = max-p × (average - th-min)/`**
    `(th-max - th-min)`
  - **`max-p`**: maximal drop probablility when the queue attains **`th-max`** threshold
  - **`p = p-tmp/(1 - nb-packets × p-tmp)`**
  - **`nb-packets`**: how many packets have been accepted since the last drop
  - **`p`** increases slowly with **`nb-packets`**
  - drops are spaced in time
- Recommended values
  - **`max-p`** = 0.02
  - if **`average`** in the middle of two thresholds, 1 drop in 50

Drop probability

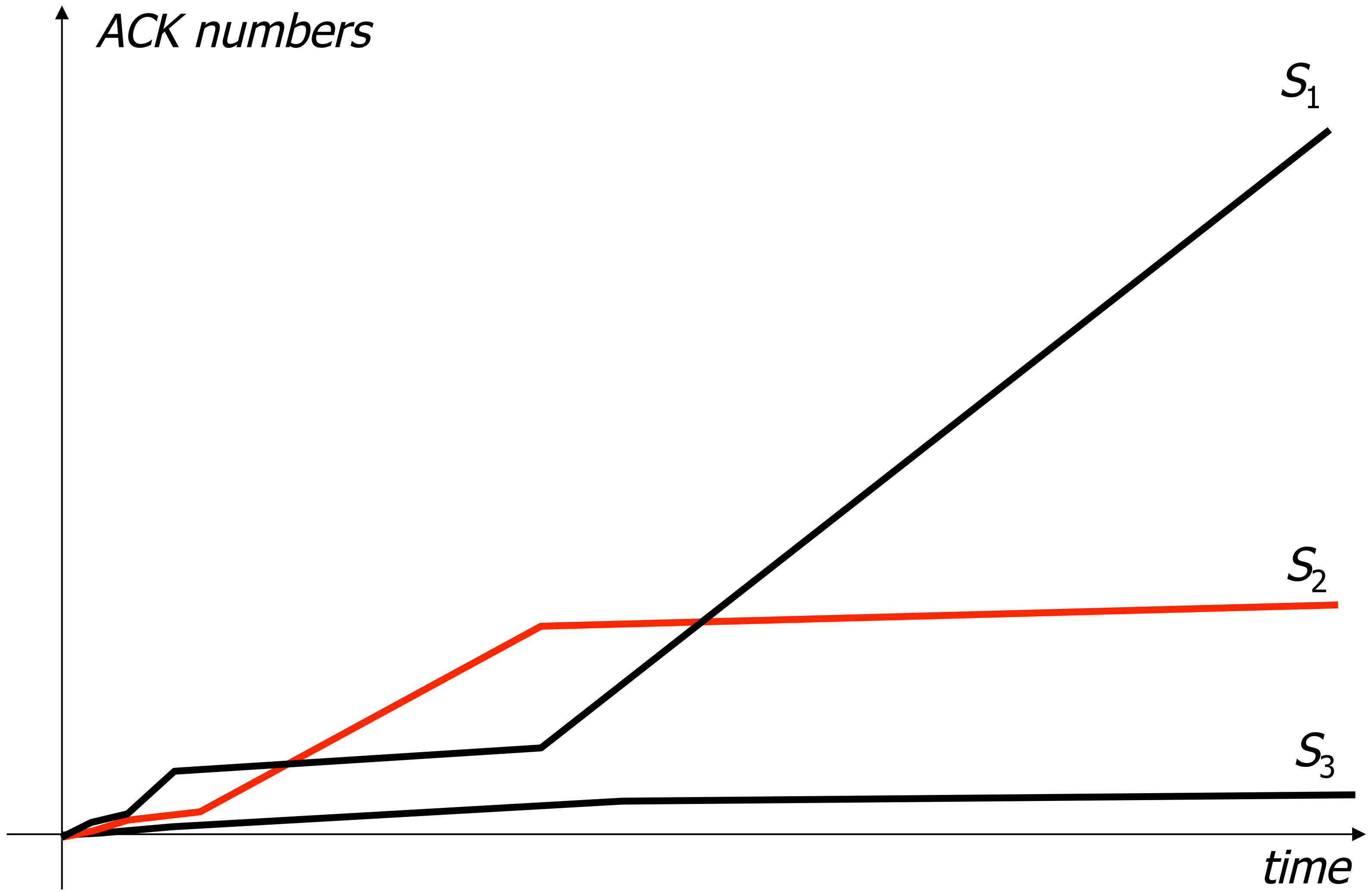


Example network for RED

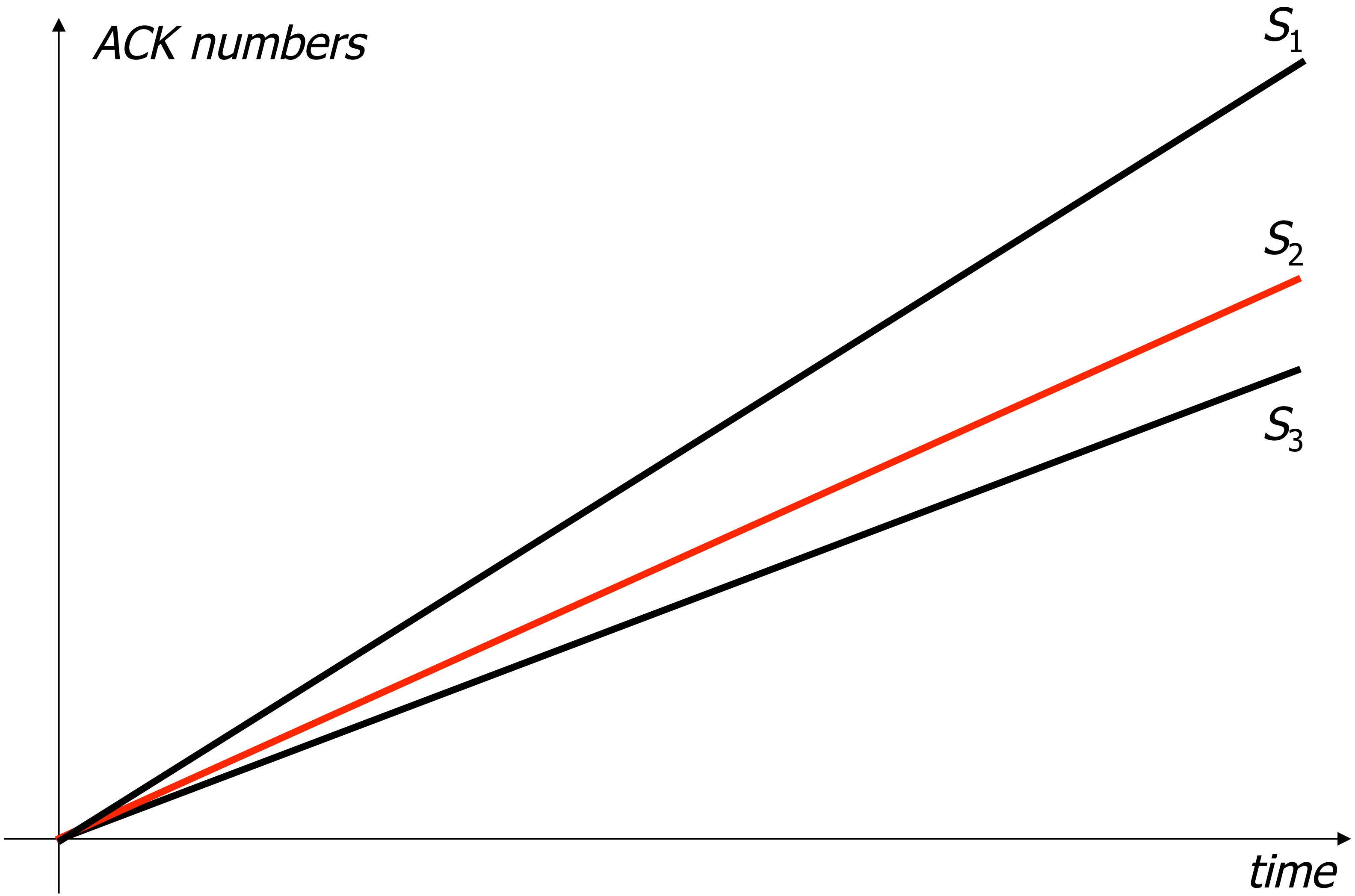


- Example network with three TCP sources
  - different link delays
  - limited queues on the link (20 packets)

Throughput in time



Throughput in time with RED



Facts to remember

- QoS in packet networks based on
  - scheduling algorithms
  - buffer management policies
- Traffic shaping helps to deal with QoS
  - limiting bursts
  - traffic description
  - traffic policing
- Used in
  - *IntServ, DiffServ*