Advanced Computer Networks

QoS in IP networks

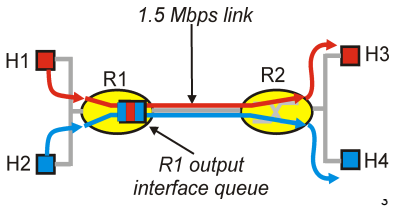
Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

Contents

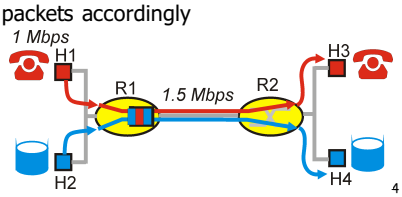
Improving QOS in IP Networks

- IETF groups are working on proposals to provide better QOS control in IP networks, i.e., going beyond best effort to provide some assurance for QOS
- Work in Progress includes Integrated Services, RSVP, and Differentiated Services
- Simple model for sharing and congestion studies:



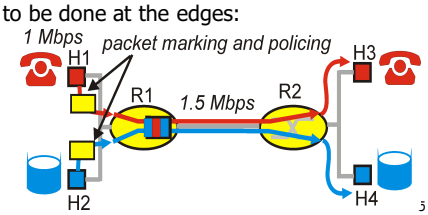
Principles for QOS Guarantees

- Consider a phone application at 1Mbps and an FTP application sharing a 1.5 Mbps link.
  - bursts of FTP can congest the router and cause audio packets to be dropped.
  - want to give priority to audio over FTP
- PRINCIPLE 1: Marking of packets is needed for router to distinguish between different classes; and new router policy to treat packets accordingly



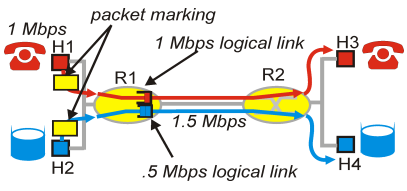
Principles for QOS Guarantees

- Applications misbehave (audio sends packets at a rate higher than 1Mbps assumed above);
- PRINCIPLE 2: provide protection (isolation) for one class from other classes
- Require Policing Mechanisms to ensure sources adhere to bandwidth requirements; Marking and Policing need to be done at the edges:



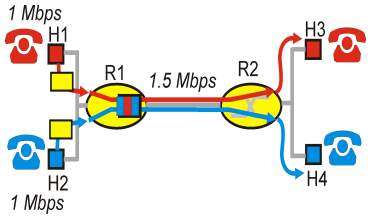
Principles for QOS Guarantees

- Alternative to Marking and Policing: allocate a set portion of bandwidth to each application flow; can lead to inefficient use of bandwidth if one of the flows does not use its allocation
- PRINCIPLE 3: While providing isolation, it is desirable to use resources as efficiently as possible



Principles for QOS Guarantees

- Cannot support traffic beyond link capacity
- PRINCIPLE 4: Need a Call Admission Process; application flow declares its needs, network may block call if it cannot satisfy the needs

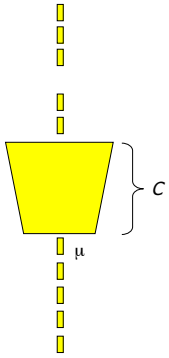


Traffic shaping

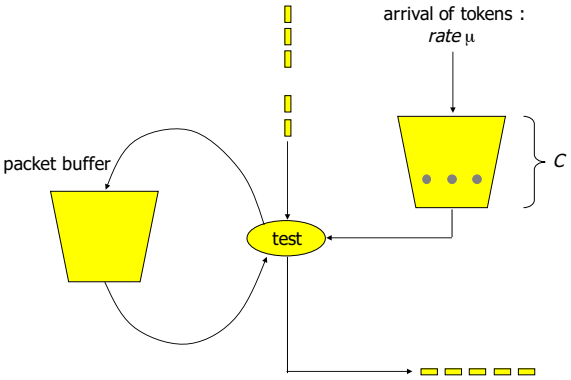
- How to prevent congestion?
  - it may result from burstiness
  - make arrivals more deterministic, obtain better performance
    - example : no. of clients in D/D/1 vs. G/D/1 or group arrivals vs. single arrivals
  - control the rate and burst size
    - traffic description - leaky bucket, token bucket
- Service contract
  - if the network knows the type of the traffic, it can reserve resources to support the traffic
  - 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
  - μ if buffer not empty
  - 0 if buffer empty
- Equivalent to the queue G/D/1/N
- Fixed size packets
  - one packet per clock tick
- Variable size packets
  - number of bytes per clock tick
- Packet loss if buffer filled



Token bucket



Token bucket

- Tokens generated with rate μ
  - 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 *C*)
- When packets are not generated, tokens accumulate
  - *n* tokens - burst of *n* packets
  - if bucket filled, tokens are lost
- Mean departure rate : μ
- Delay limited by C/μ (Little's formula)

Example

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

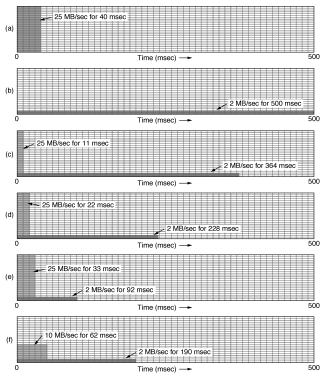
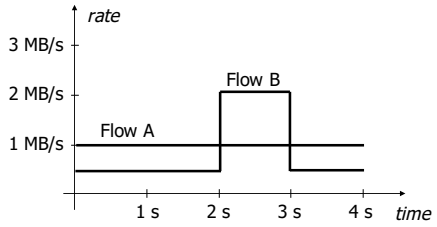


Fig. 5-25. (a) Input to a leaky bucket. (b) Output from a leaky bucket. (c) - (e) Output from a token bucket with capacities of 250KB, 500KB, and 750KB. (f) Output from a 500KB token bucket feeding a 10 MB/sec leaky bucket.

Burst duration

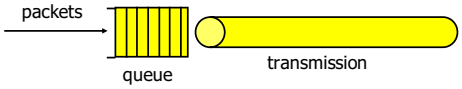
- Burst duration - S sec
- Size of the bucket - C bytes
- Maximal departure rate - p bytes/s
- Token arrival rate - μ bytes /s
  - burst of $C + \mu S$ bytes
  - burst of pS
  - $C + \mu S = pS \rightarrow S = C/(p - \mu)$
- Example
  - C = 250 Ko, p = 25 Mo/s, μ = 2 Mo/s
  - S = 11 ms

Traffic description



- Flow A : μ = 1 MB/s, C = 1 byte
- Flow B : μ = 1 MB/s, C = 1 Mbyte
  - during 2 s, the flow saves 2 s × 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

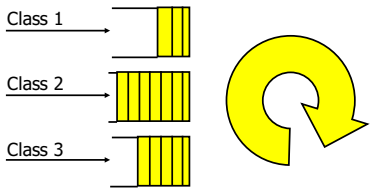
FIFO

- 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)

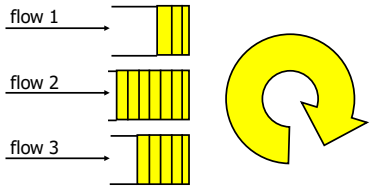


- Also called Custom Queueing (CISCO)
- Each queue serviced in round-robin order
- Dequeue a configured byte count from each queue in each cycle
- Each class obtains a configured proportion of link capacity

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

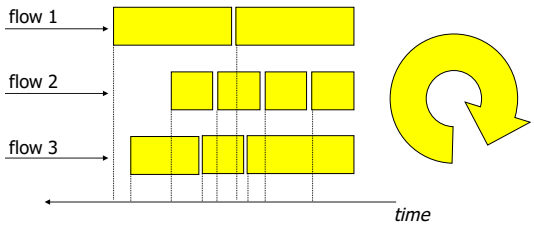


- Similar to Processor Sharing or Time Sharing
  - one queue per flow
  - cyclic service, one packet at a time

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

$$x_i = D\frac{w_i}{\sum w_i}$$ D : link capacity

Rate guarantee

- Weights expressed as proportions (w_i - guaranteed weight)

$$x_i = Dw_i, w_i \leq 1$$

  - If no packets of a given flow, unused capacity shared equally by other flows

$$x_i \geq D \times w_i$$

  - Weights to guarantee a given rate

$$w_i = x_i / D$$

Delay guarantee

- Flow constrained by a token bucket
  - rate μ, capacity C
  - delay limited by C/μ
- If $x_i > \mu$ (the rate obtained is sufficient for the flow)
  - delay limited by C/μ
  - total delay limited by the sum of all delays