Advanced Computer Networks

QoS in IP networks

Prof. Andrzej Duda
duda@imag.fr

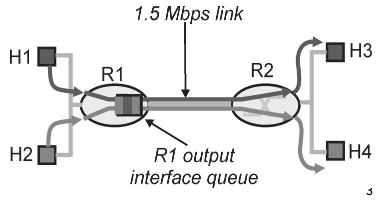
http://duda.imag.fr

Contents

- QoS principles
- Traffic shaping
  - leaky bucket
  - token bucket
- Scheduling
  - FIFO
  - Fair queueing
  - RED
- IntServ
- DiffServ

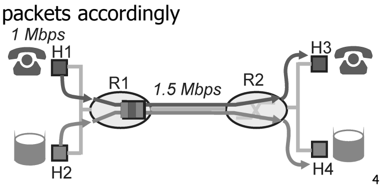
Improving QOS in IP Networks

- 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:



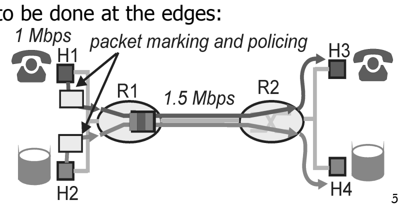
Principles for QOS Guarantees

- 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



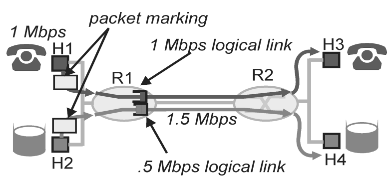
Principles for QOS Guarantees

- 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:



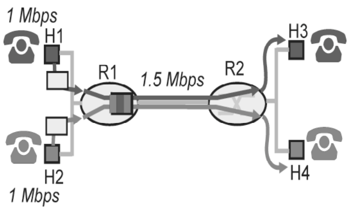
Principles for QOS Guarantees

- 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



7

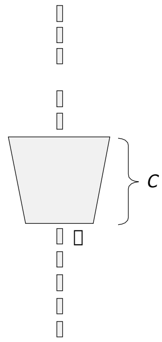
Traffic shaping

- 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)*

8

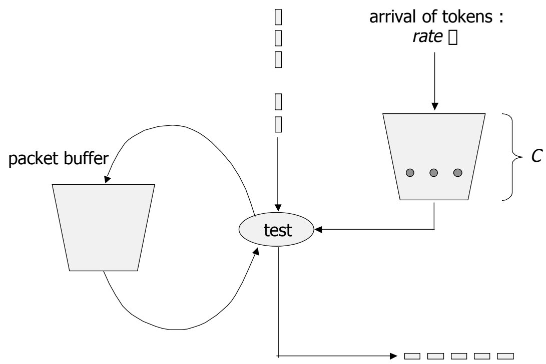
Leaky bucket

- Limited size buffer with constant departure rate
  - □ 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



9

Token bucket



10

Token bucket

- Tokens generated with rate □
  - 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 : □
- Delay limited by *C*/□ (Little's formulae)

11

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, □ = 2 MB/s
  2. token bucket with C = 250 KB, p = 25 MB/s, □ = 2 MB/s
  3. token bucket with C = 500 KB, p = 25 MB/s, □ = 2 MB/s
  4. token bucket with C = 750 KB, p = 25 MB/s, □ = 2 MB/s
  5. token bucket with C = 500 KB, p = 25 MB/s, □ = 2 MB/s and leaky bucket with C = 1 MB, p = 10 MB/s, □ = 2 MB/s

12

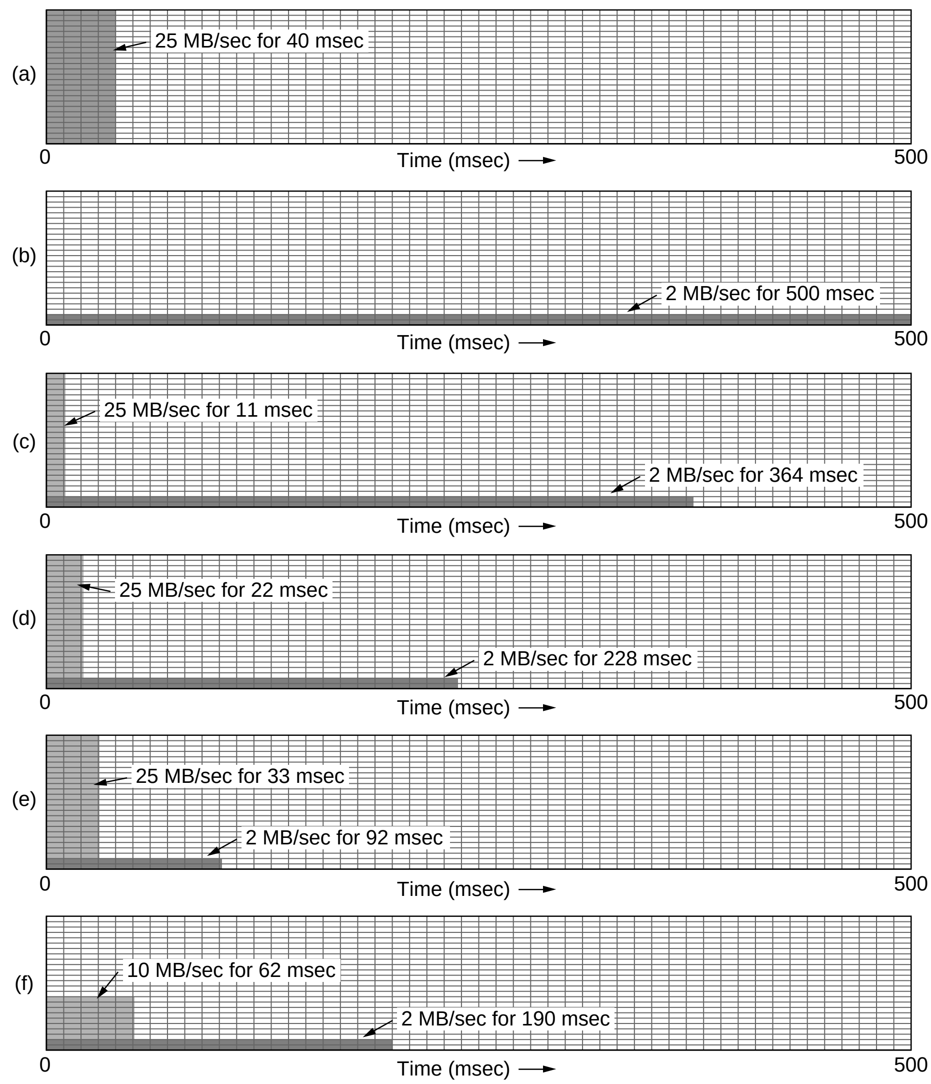


Fig. 5-25. (a) Input to a leaky bucket. (b) Output from a leaky bucket. (c) - (e) Output from a token bucket with capacities of 250KB, 500KB, and 750KB. (f) Output from a 500KB token bucket feeding a 10 MB/sec leaky bucket.

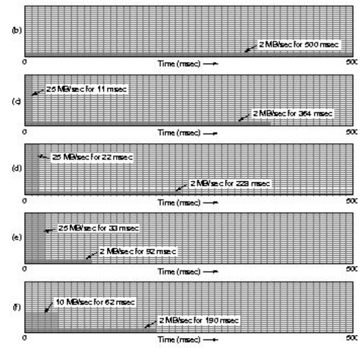


Fig. 5-25. (a) Input to a leaky bucket. (b) Output from a leak bucket. (c) - (e) Output from a token bucket with capacities 250KB, 500KB, and 750KB. (f) Output from a 500KB toke bucket feeding a 10 MB/sec leaky bucket.

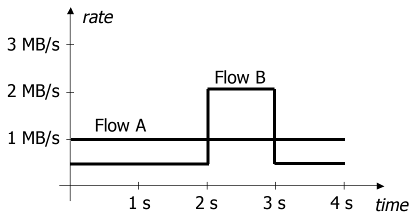
13

Burst duration

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

14

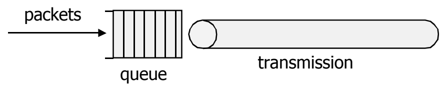
Traffic description



- Flow A : □ = 1 MB/s, *C* = 1 byte
- Flow B : □ = 1 MB/s, *C* = 1 Mbyte
  - during 2 s, the flow saves 2 s □ 0.5 MB/s = 1 MB

15

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

16

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

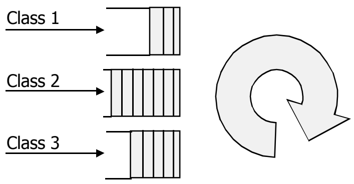
17

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?

18

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

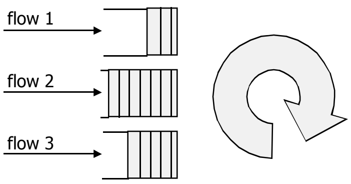
19

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

20

Per Flow Round Robin



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

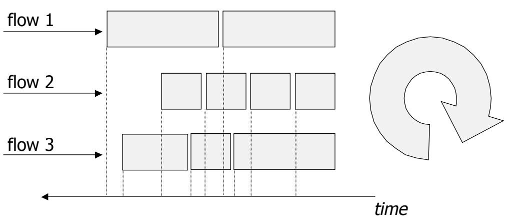
21

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

22

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

23

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

24

Rate guarantee

- Weights expressed as proportions (w_i - guaranteed weight)

$$x_i = D w_i, w_i \square 1$$

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

$$x_i \geq D \square w_i$$

  - Weights to guarantee a given rate

$$w_i = x_i / D$$

25

Delay guarantee

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

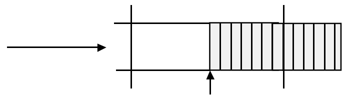
26

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 packet dropped
  - several sources detect congestion and enter slow start at the same instant

27

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

28

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

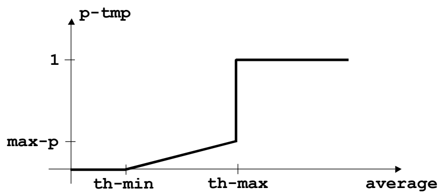
29

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

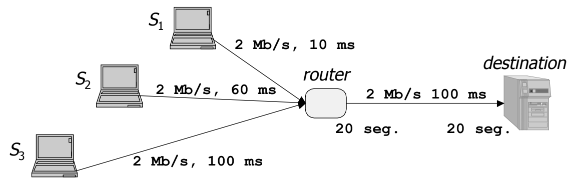
30

Drop probability



31

Example network for RED



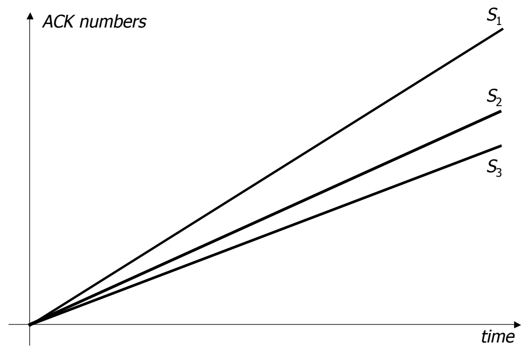
- Example network with three TCP sources
  - different link delays
  - limited queues on the link (20 packets)

32

Throughput in time

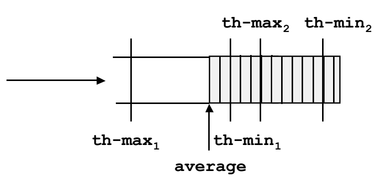
33

Throughput in time with RED



34

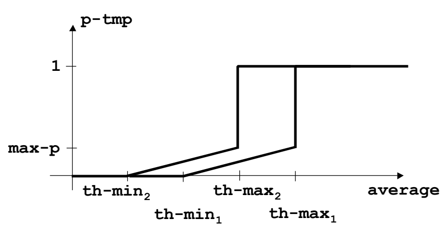
WRED



- Weighted RED
- Different thresholds for different classes
  - higher priority class - higher thresholds
    - lower drop probability
  - lower priority class - lower thresholds
    - greater drop probability
- Method for service differentiation

35

Different drop probabilities



36

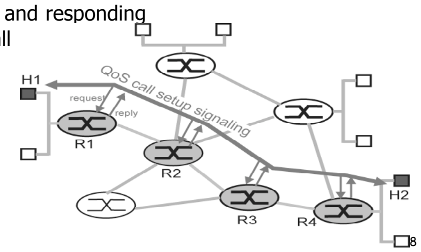
QoS architectures

- Integrated Services (IntServ)
  - per flow reservation at routers (RSVP protocol for reservation)
  - per flow scheduling
- Differentiated Services (DiffServ)
  - no reservation
  - classification at the border
  - scheduling per aggregated classes in the backbone

37

Integrated Services

- An architecture for providing QOS guarantees in IP networks for individual application sessions
- Relies on resource reservation, and routers need to maintain state info, maintaining records of allocated resources and responding to new Call setup requests on that basis



38

Flow Admission

- Session must first declare its QOS requirement (T-spec) and characterize the traffic it will send through the network
- Routers check for resources and reserve them
- A signaling protocol is needed to carry QOS requirement to the routers where reservation is required

39

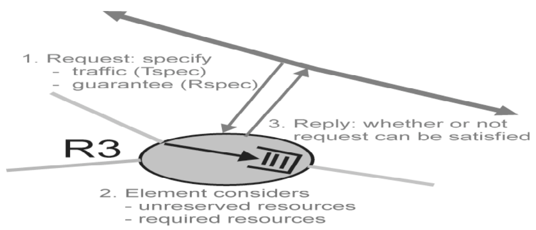
RSVP (Reservation Protocol)

- PATH message
  - **T-spec** - source traffic description
    - defines the traffic characteristics
    - token bucket: rate, capacity, and peak rate
  - packet from source to destination determines the return route
- RESV message
  - **R-spec**: if receiver wants to have better QoS (e.g. higher rate and jitter)
  - packet from destination to source follows the route established by PATH
  - reservations are done upon receiving this message

40

Flow Admission

- Flow Admission: routers will admit flows based on their T-spec and R-spec and base on the current resource allocated at the routers to other flows



41

Integrated Services: Classes

- **Guaranteed QOS**: this class is provided with <u>firm bounds</u> on queuing delay at a router; envisioned for hard real-time applications that are highly sensitive to end-to-end delay expectation and variance
  - rate and delay
- **Controlled Load**: this class is provided a QOS <u>closely approximating</u> an unloaded network; envisioned for today's IP network real-time applications which perform well in an unloaded network
  - rate

42

Differentiated Services

- Intended to address the following difficulties with Intserv and RSVP
  - **Scalability**: maintaining states by routers in high speed networks is difficult due to the very large number of flows
  - **Flexible Service Models**: IntServ has only two classes, want to provide more classes - *relative* service distinction (Platinum, Gold, Silver, …)
  - **Simpler signaling**: (than RSVP) many applications and users may only want to specify a more qualitative notion of service

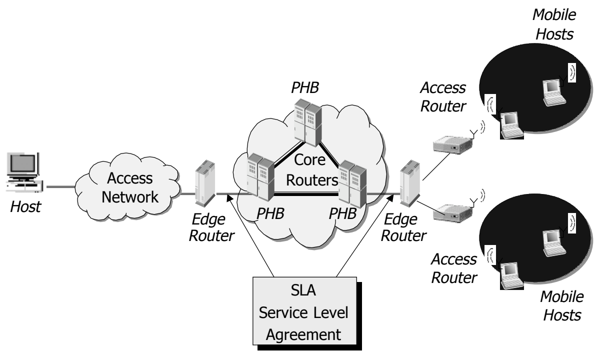
43

Differentiated Services

- Approach:
  - Only simple functions in the core, and relatively complex functions at edge routers (or hosts)
  - Do not define service classes, instead provide functional components with which service classes can be built

44

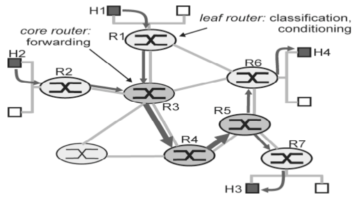
End-to-end *DiffServ* architecture



45

Edge Functions

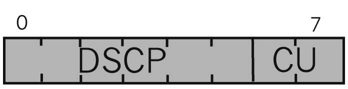
- At DS-capable host or first DS-capable router
- **Classification**: edge node marks packets according to classification rules to be specified (manually by admin, or by some TBD protocol)
- **Traffic Conditioning**: edge node may delay and then forward or may discard



46

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



47

Core Functions

- **Forwarding**: according to "Per-Hop-Behavior" or PHB specified for the particular packet class; such PHB is strictly based on class marking (no other header fields can be used to influence PHB)
- QoS, if sufficient provisioning

- BIG ADVANTAGE:

  No state info to be maintained by routers!

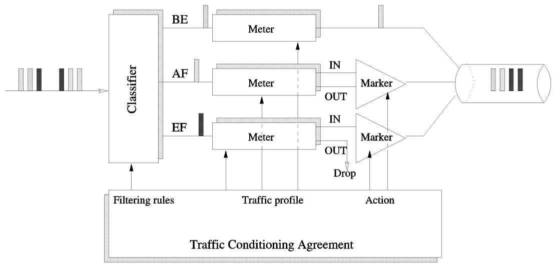
48

DiffServ service classes

- Two main types of application
  - interactive (games, interactive distributed simulations, VoIP, device control)
    - delay, jitter
  - elastic (data transfer)
    - sustained throughput
- Traffic classes
  - EF (Expedited Forwarding)
    - short delay, small jitter
  - AF (Assured Forwarding)
    - minimal sustained throughput
    - 4 subclasses with 3 different drop probabilities (12 subclasses in total)
  - BE (Best Effort)

49

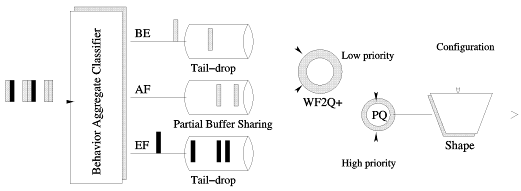
DiffServ - Edge router



- Classification, metering, marking

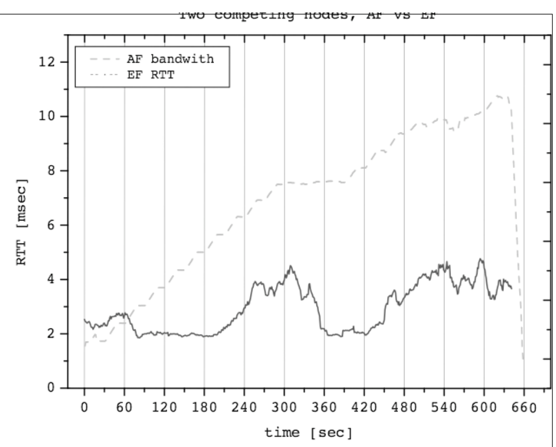
50

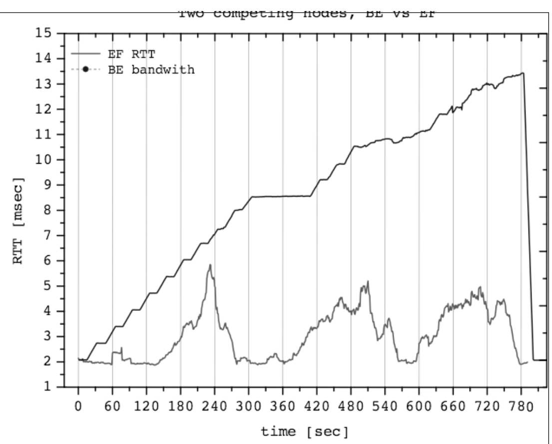
DiffServ - Core router



- Queue management and scheduling
  - EF: high priority
  - AF, BE: WFQ - Weighted Fair Queueing
- Traffic shaping

51





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
- Current approach
  - *DiffServ*

54