Reality of net multimedia support

Network support for multimedia

Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic "class"	None of soft	Packet market, scheduling, policing.	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

Deployment of network QoS

- Why are not Intserv and Diffserv widely deployed?
  - Cost of complexity
  - End-to-end peering problems
    - Difficult to negotiate service classes across ISPs
    - Could standardize service classes, but would complicate ISP peering agreements
    - Still only a best effort service – no guarantee QoS or 'gold' service met!
  - QoS does not work well under severely constrained resources
  - QoS has no impact when capacity plentiful
  - Cheaper to keep core *fast* and *dumb* than to deploy QoS
  - Application layer mechanisms can adjust for network shortcomings

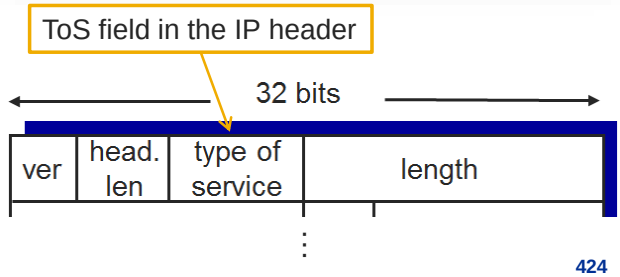
- Main area where diffserv ideas see traction are cellular networks. Why?

Multiple classes of service

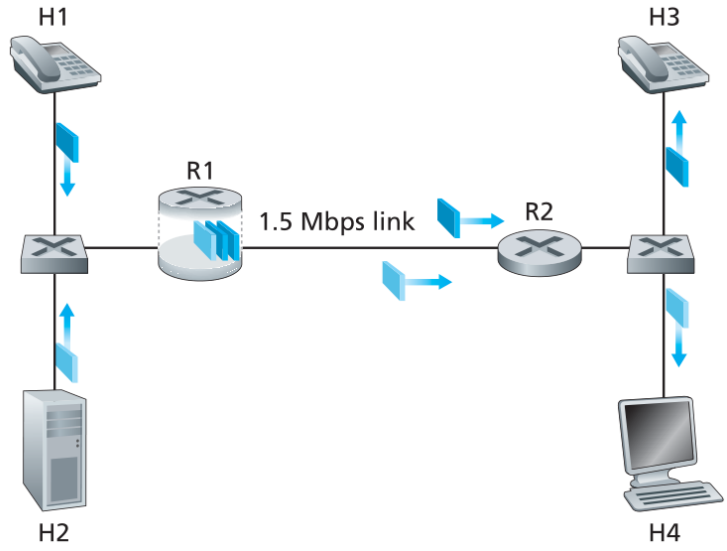
- Thus far: making the best of best effort service
  - One-size fits all service model

- Alternative: multiple classes of service
  - Partition traffic into classes
  - Network treats different classes of traffic differently

- Granularity:
  - Differential service among multiple **classes**, **not** among individual **connections**



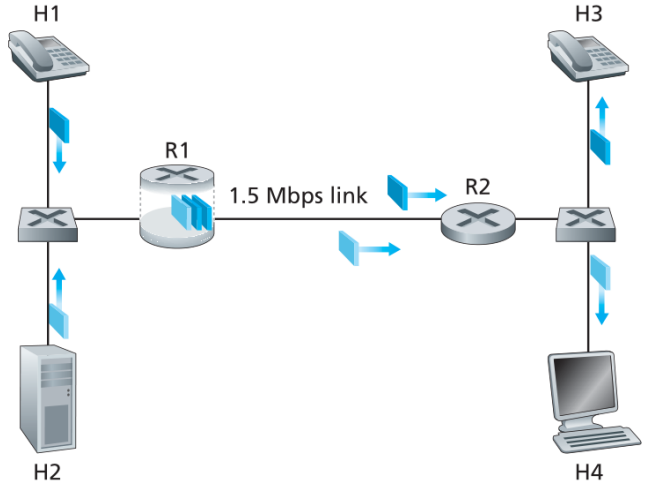
Multiple classes of service



Multiple classes of service

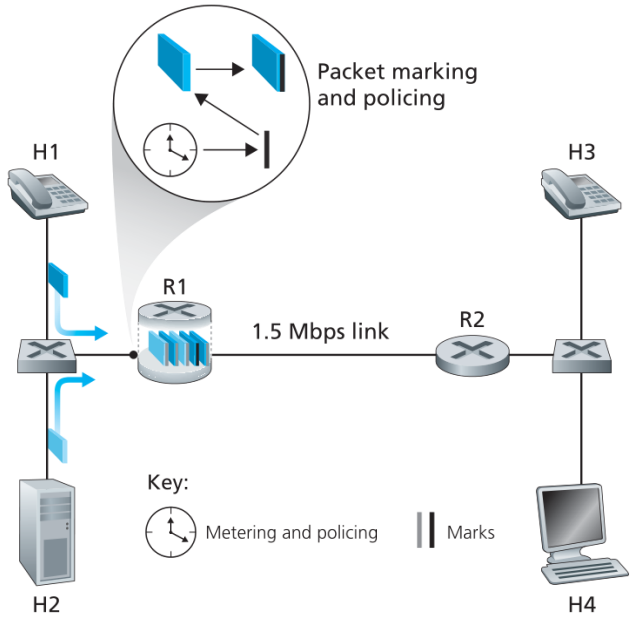
- 1Mbps VoIP and 0.5 Mbps HTTP share 1.5 Mbps link
- What's the problem?
  - HTTP bursts can congest router, cause audio loss
  - Want to give priority to audio over HTTP

Insight 1: Packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly



Packet marking

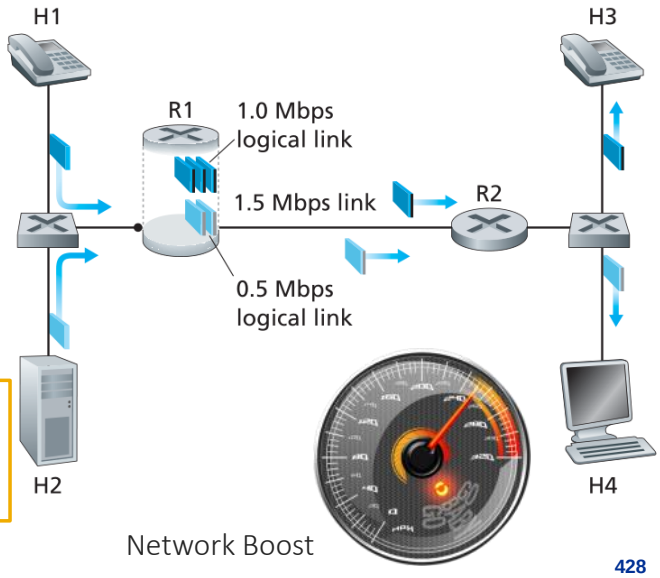
- What happens if VoIP sends higher than declared rate?
- Policing: force source adherence to bandwidth allocations
- Where should the packets be marked? (routers or end hosts)



Insight 2: Provide protection (isolation) for one class from others

Strict isolation

- Allocate fixed (non-sharable) bandwidth to each flow
- What's the problem?
  - Inefficient use of bandwidth if flows does not use its allocation



Insight 3: While providing isolation, it is desirable to use resources as efficiently as possible

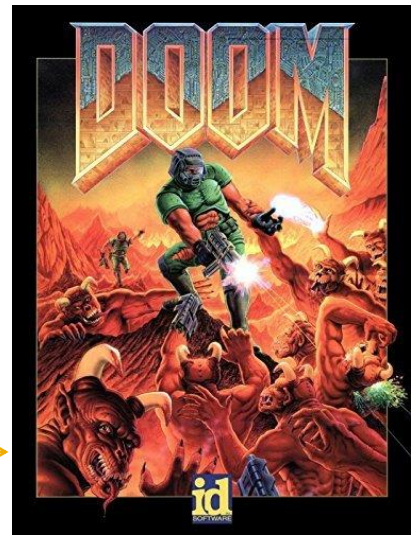
Scheduling and Policing

Scheduling

Which queued packet to send next?

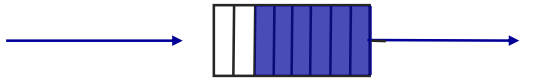
How to constrain flow over-sending?

Policing



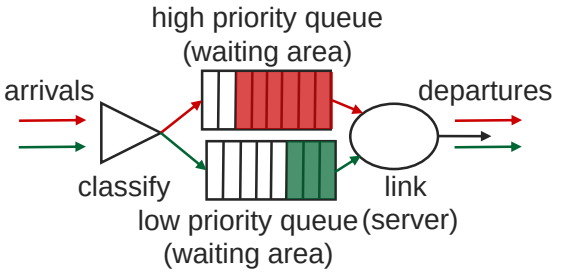
Scheduling disciplines

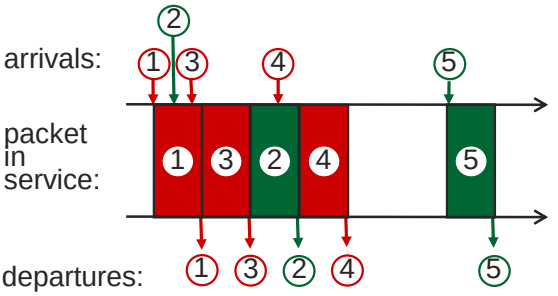
- FIFO (first in first out) scheduling: send in order of arrival to queue
  - Real-world example?
  - Discard policy: if packet arrives to full queue: who to discard?
  - Tail drop: drop arriving packet
  - Priority: drop/remove on priority basis
  - Random: drop/remove randomly



Scheduling disciplines

- Priority scheduling:
  - Send highest priority queued packet first
- 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?

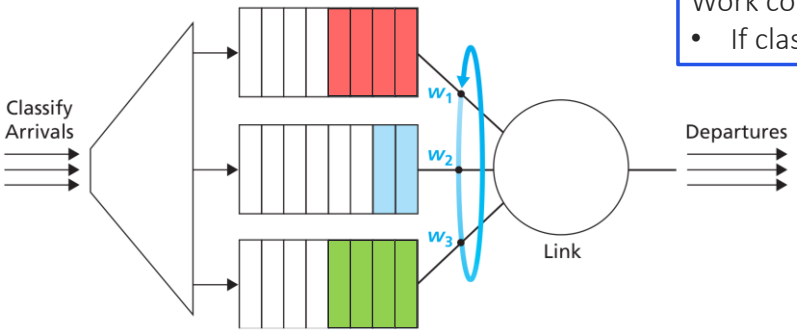




Scheduling policies

- Round Robin (RR) scheduling:
  - Multiple classes
  - Cyclically scan class queues, sending one complete packet from each class (if available)
- Weighted Fair Queuing (WFQ):
  - Generalized Round Robin
  - Each class gets weighted amount of service in each cycle

Work conserving disciplines:
- If class queue empty go to next one



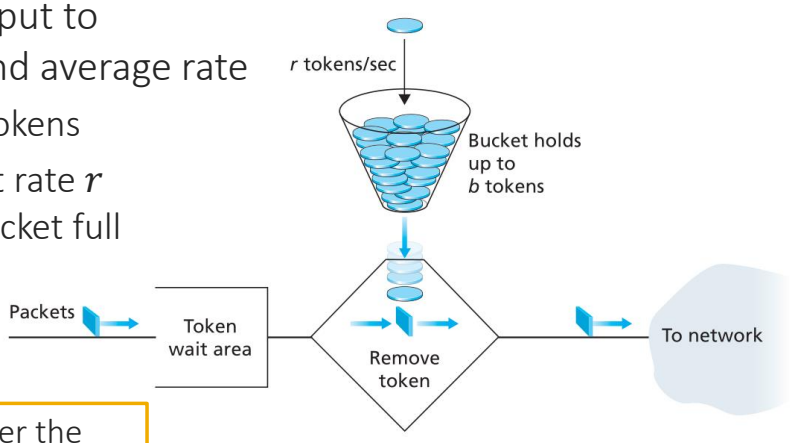
In WFQ each queue gets a share of the link rate proportional to its weight $w_i/\sum w_j$

Policing criteria

- 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 pps 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

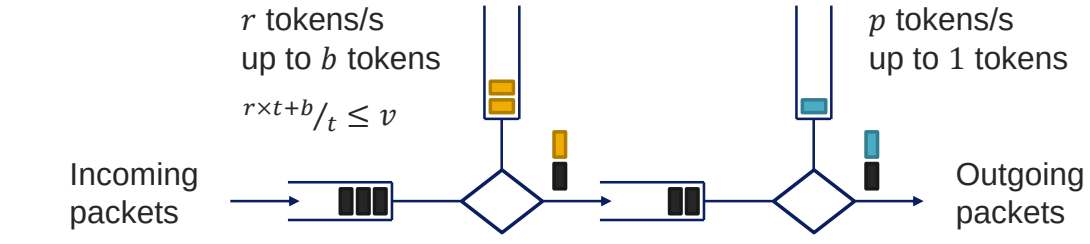


How many packets can enter the network during time interval t?

$r \times t + b$

Policing question

Design a token bucket policing mechanism that limits *average rate* to v packets and *peak rate* to p packets in an interval t.

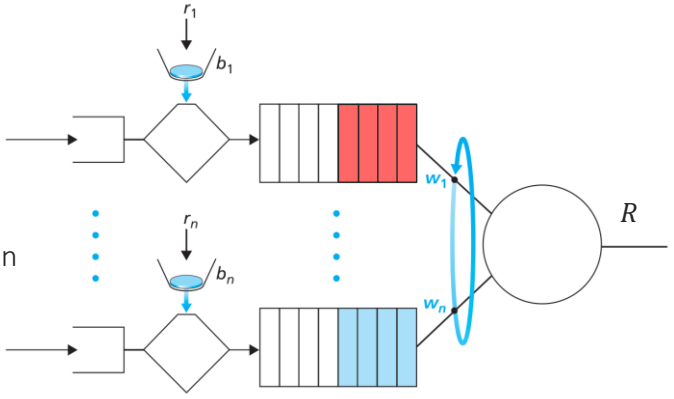


Why not limit b to 1?

Why not make b large?

Guarantee of delay

- Token bucket and WFQ mechanisms can be combined to guarantee an upper bound on delay



What is the minimum forwarding rate for traffic class (queue) 1?

Answer: $R \times w_1 / \sum w_j$

What is the maximum queuing delay of packets in queue 1 following a token bucket shaper with token generation rate r_1 and capacity b_1?

Answer: $d_{\max} = \frac{b_1}{R \times w_1 / \sum w_j}$

What happens to $d_{\max}$ when $r_1 > R \times w_1 / \sum w_j$?