Information Technology Engineering

Mohammad Hossein Manshaei

manshaei@gmail.com

1393

Voice and Video over IP

MULTIMEDIA NETWORKING

Slides derived from those available on the Web site of the book "Computer Networking", by Kurose and Ross, PEARSON

Multimedia networking: outline

7.1 multimedia networking applications

7.2 streaming stored video

7.3 voice-over-IP

7.4 protocols for real-time conversational applications: RTP, SIP

7.5 network support for multimedia

Network support for multimedia

Approach	Unit of Allocation	Guarantee	Deployment to date	Complexity	Mechanisms
Making the best of best-effort service	All traffic treated equally	None, or soft	everywhere	minimal	Application-layer support, CDN, over-provisioning
Differential QoS	Classes of Flows	None, or Soft	some	medium	Policing, Scheduling
Guaranteed QoS	Individual Flows	Soft or hard, once a flow is admitted	little	high	Policing, Scheduling, call admission and signaling

Dimensioning best effort networks

- *Approach:* deploy enough link capacity so that congestion doesn't occur, multimedia traffic flows without delay or loss
  - low complexity of network mechanisms (use current "best effort" network)
  - high bandwidth costs
- Challenges:
  - *estimating network traffic demand (bandwidth provisioning):* needed to determine how much bandwidth is "enough" (for that much traffic)
  - *network dimensioning:* how to design a network topology (where to place routers, how to interconnect routers with links, and what capacity to assign to links) to achieve a given level of end-to-end performance

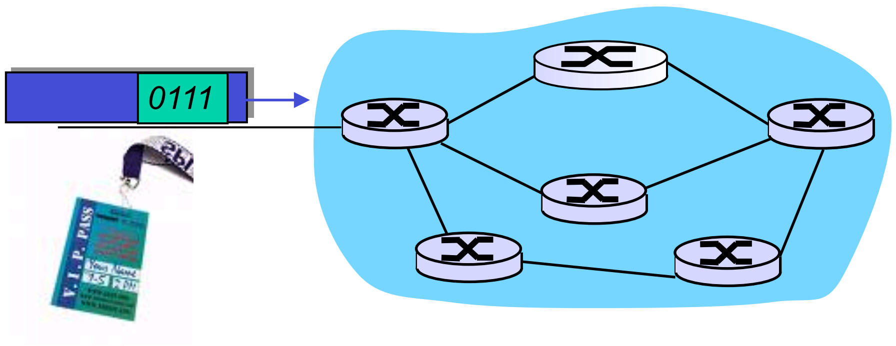
Dimensioning best effort networks

- *Find answer to following issues:*
  - *Models of traffic demand between network end points*
    - *Call level and packet level*
  - *Well-defined performance requirements*
    - *E.g.*, probability that the end-to-end delay of the packet is greater than a maximum tolerable delay be less than some small value
  - Models to predict end-to-end performance for a given workload model, and techniques to find a minimal cost bandwidth allocation that will result in all user requirements being met

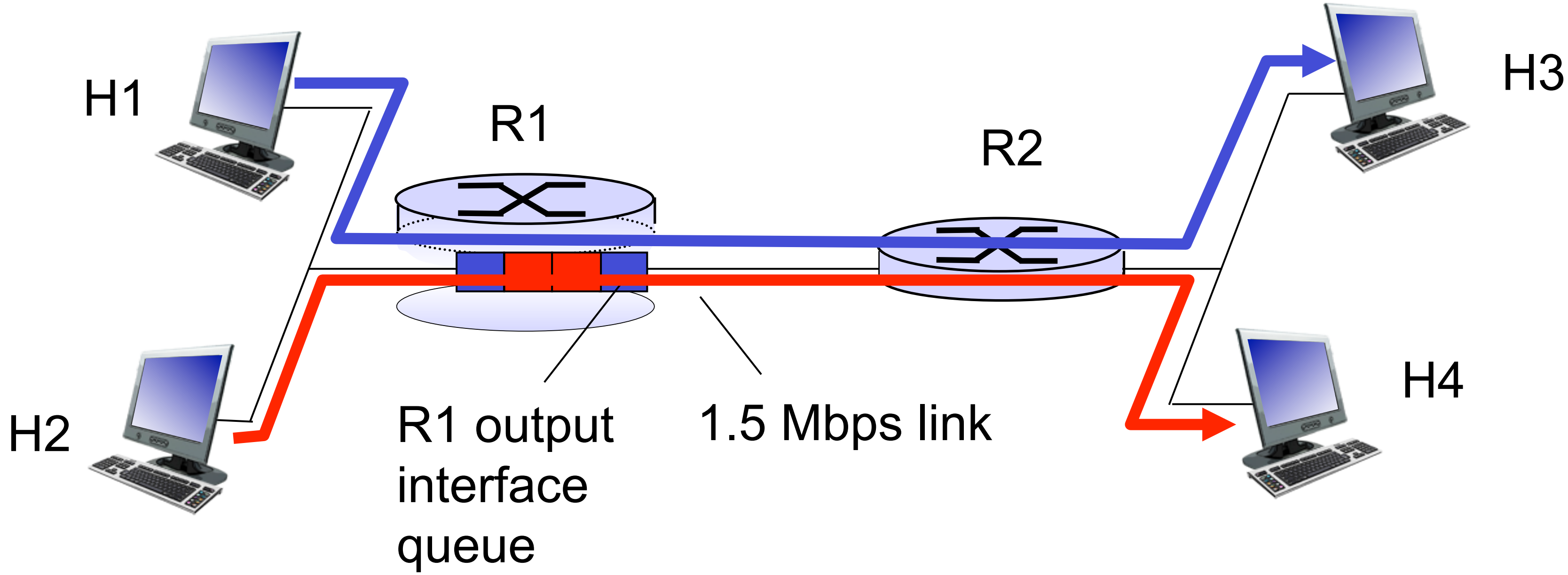
Providing multiple classes of service

- 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 (analogy: VIP service versus regular service)

- Granularity: differential service among multiple classes, *not among individual connections*
- history: ToS bits

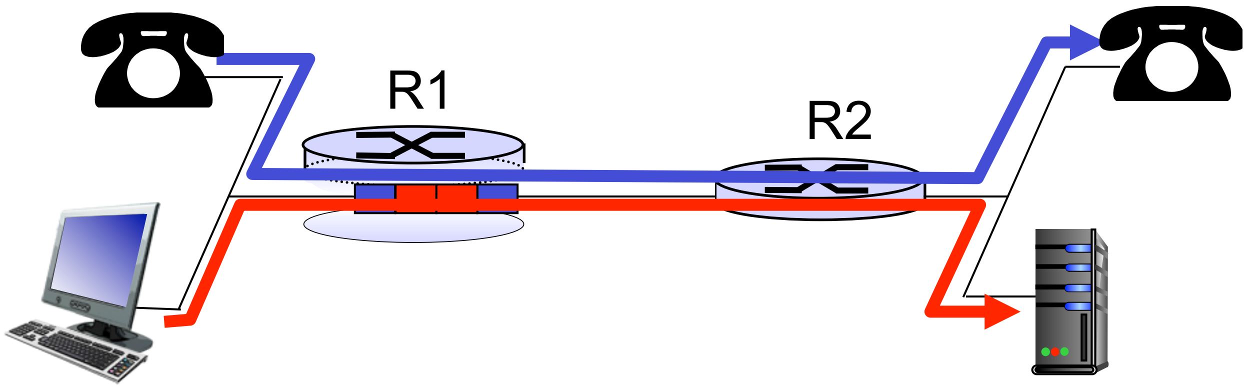


Multiple classes of service: scenario



Scenario 1: mixed HTTP and VoIP

- example: 1Mbps VoIP, HTTP share 1.5 Mbps link.
  - HTTP bursts can congest router, cause audio loss
  - want to give priority to audio over HTTP



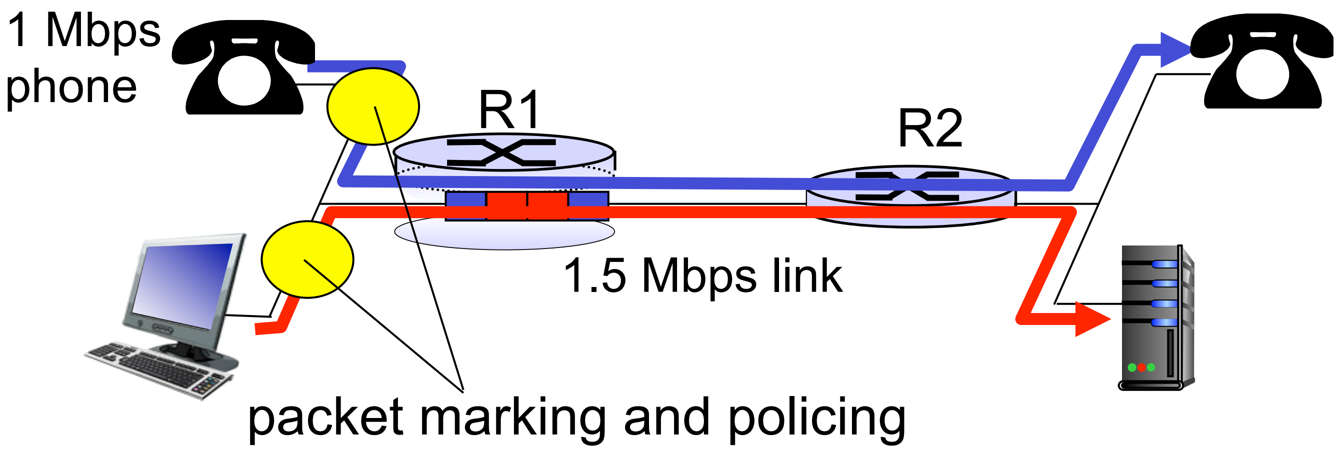
Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

Principles for QOS guarantees (more)

- What if applications misbehave (VoIP sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations

1. marking, policing at network edge

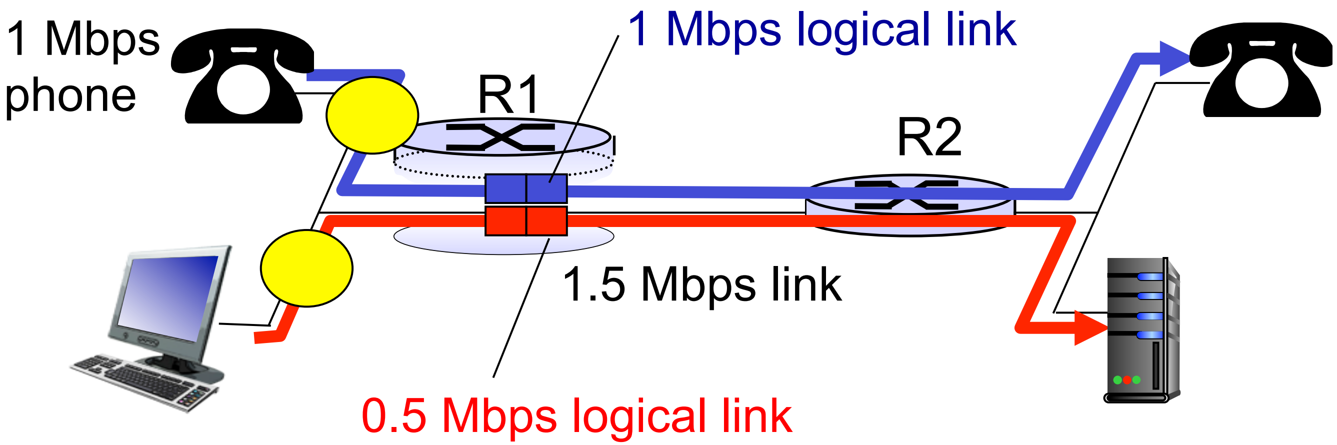


Principle 2

provide protection (isolation) for one class from others

Principles for QOS guarantees (more)

2. Allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flow doesn't use its allocation

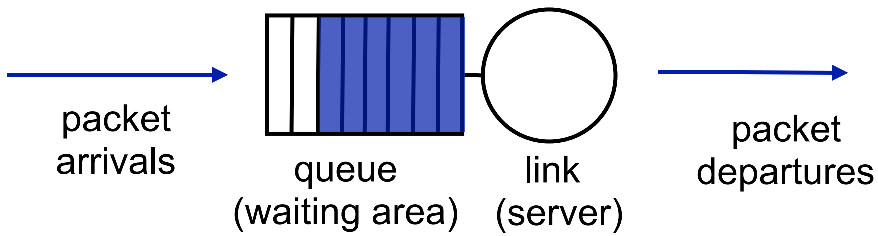


Principle 3

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

Scheduling and policing mechanisms

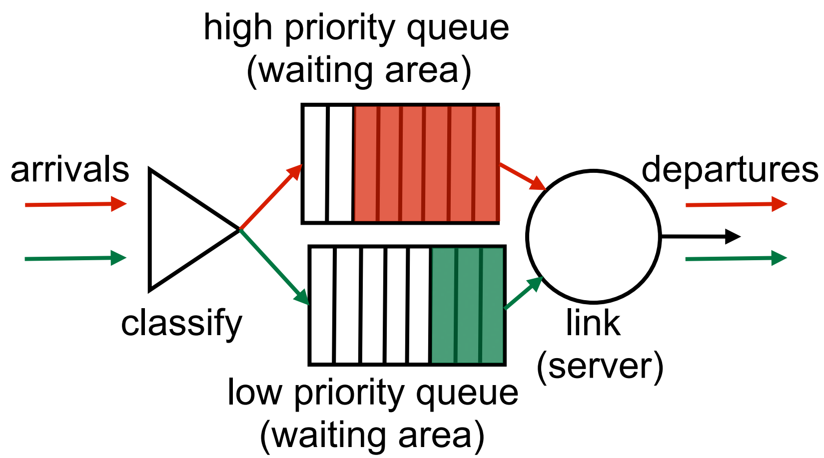
- *Scheduling:* choose next packet to send on link
- *FIFO (first in first out) scheduling:* send in order of arrival to queue
  - real-world example?
  - *Packet-Discarding-Policy:* if packet arrives to full queue: who to discard?
    - *tail drop:* drop arriving packet
    - *priority:* drop/remove on priority basis
    - *random:* drop/remove randomly

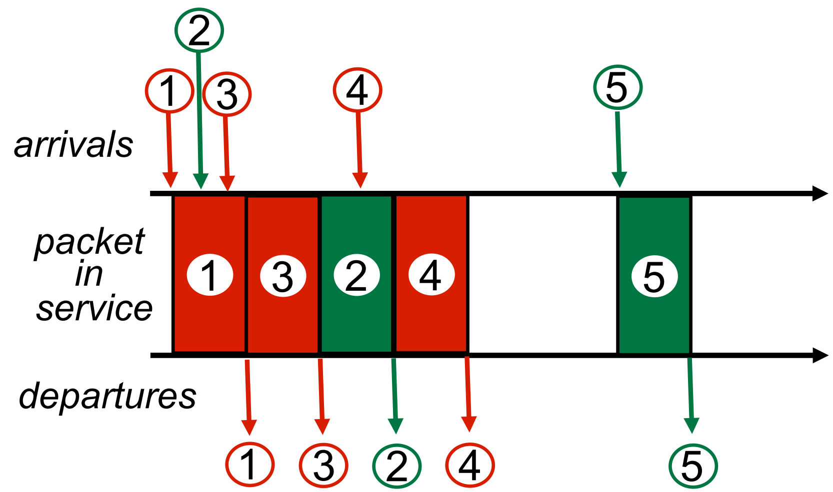


Scheduling policies: priority

Priority scheduling: send highest priority queued packet

- 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?

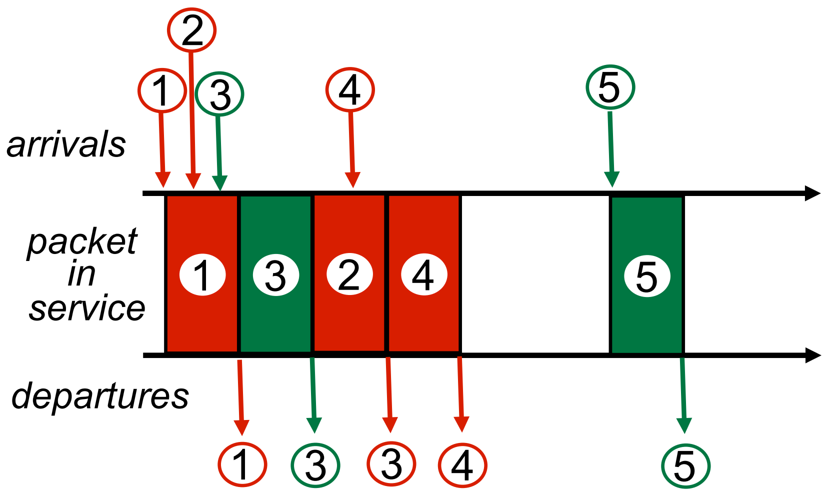




Scheduling policies: still more

Round Robin (RR) scheduling:

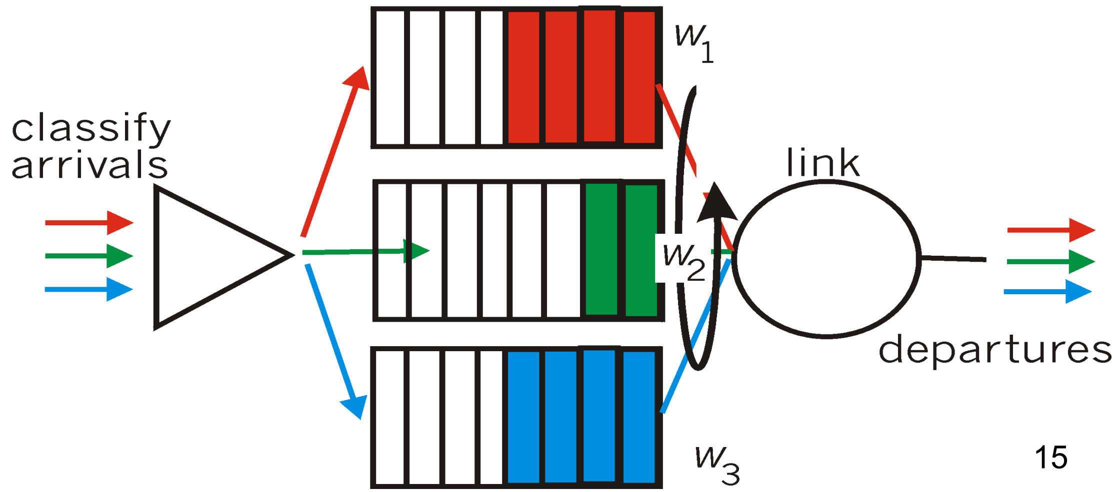
- multiple classes
- cyclically scan class queues, sending one complete packet from each class (if available)
- real world example?



Scheduling policies: still more

Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class gets weighted amount of service in each cycle
- real-world example?
- Class i will then be guaranteed to receive a fraction of service equal to $w_i/(\Sigma w_j)$



Policing mechanisms

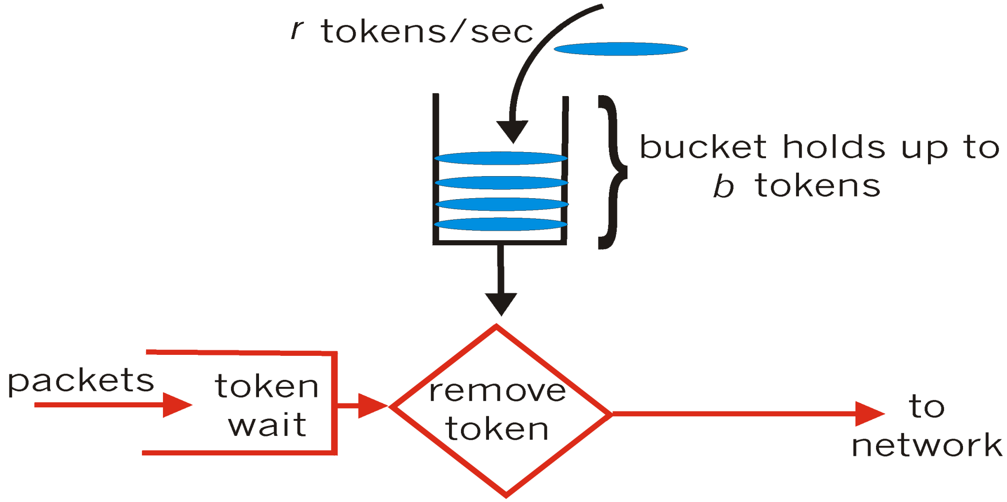
goal: limit traffic to not exceed declared parameters

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., 6,000 packets per minute, while limiting the flow's **peak rate** to 1,500 packets per second
- *(max.) burst size:* max number of pkts sent consecutively (with no intervening idle)

Policing mechanisms: implementation

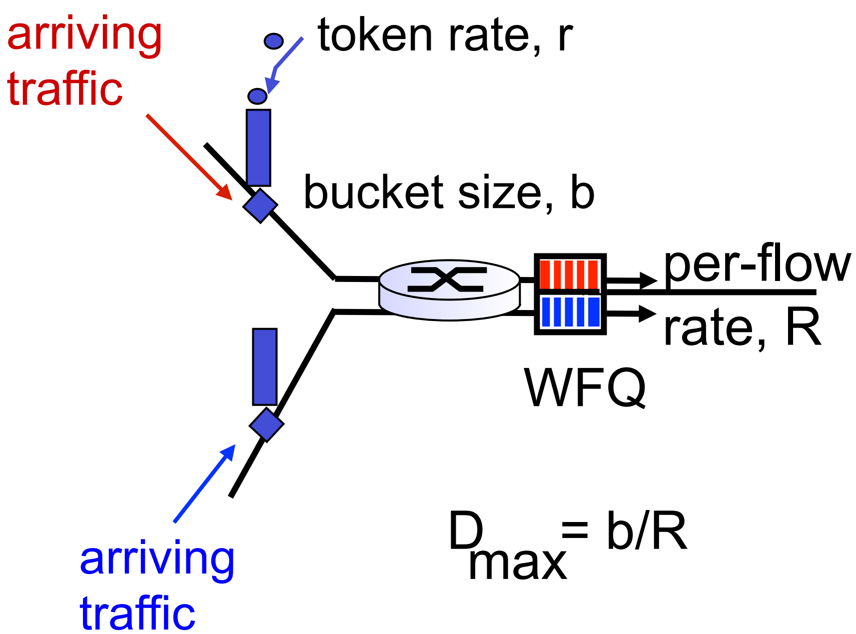
Token bucket: limit input to specified *burst size* and *average rate*

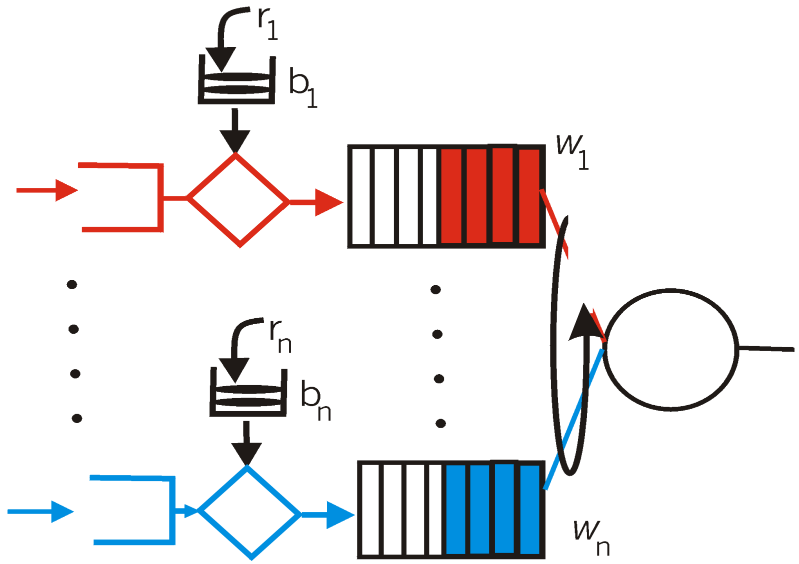


- bucket can hold b tokens
- tokens generated at rate *r token/sec* unless bucket full
- *over interval of length t: number of packets admitted less than or equal to (r t + b)*

Policing and QoS guarantees

- Token bucket, WFQ combine to provide guaranteed upper bound on delay, i.e., *QoS guarantee!*





Differentiated Services [RFC 2475]

- Want "qualitative" service classes
  - "behaves like a wire"
  - relative service distinction: Platinum, Gold, Silver
- *Scalability:* simple functions in network core, relatively complex functions at edge routers (or hosts)
  - signaling, maintaining per-flow router state difficult with large number of flows
- Don't define service classes, provide functional components to build service classes

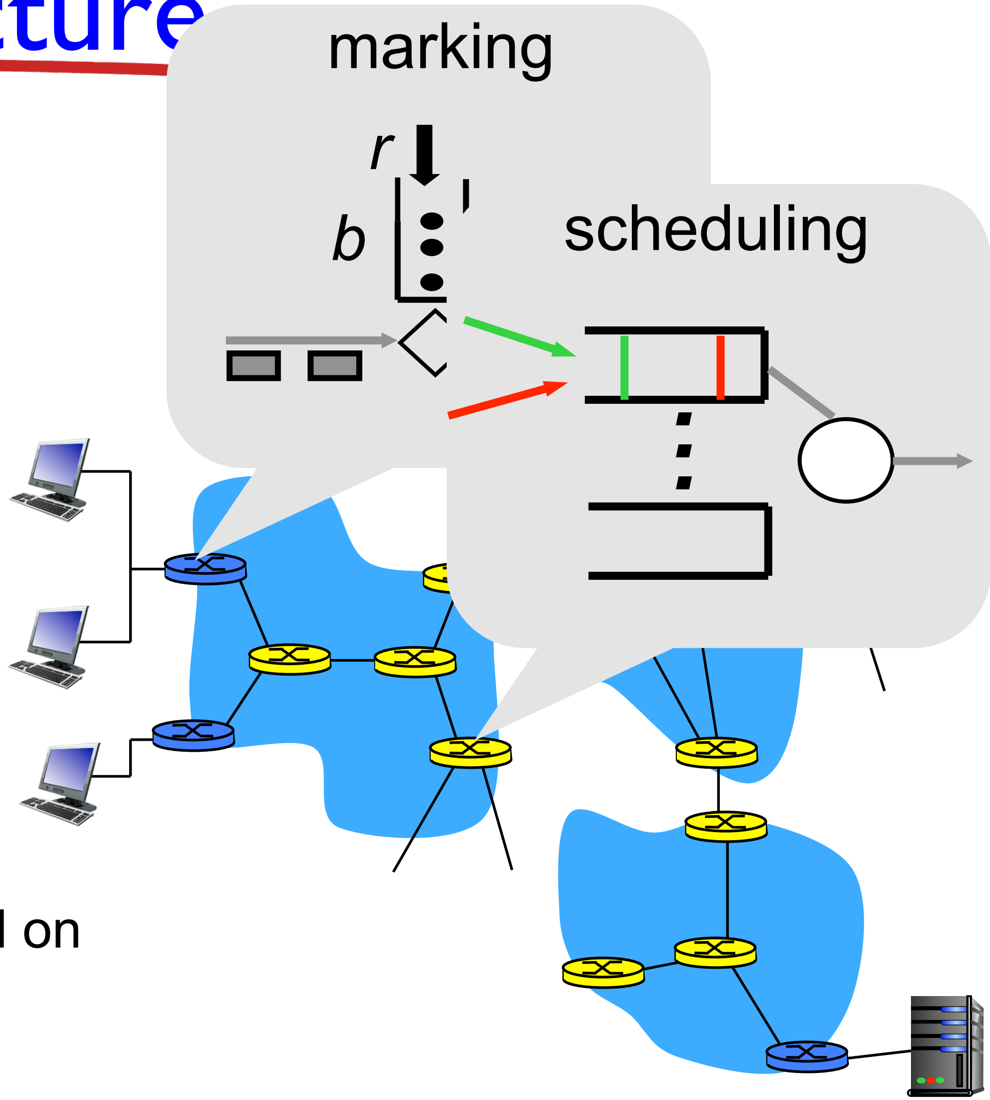
Diffserv architecture

edge router:

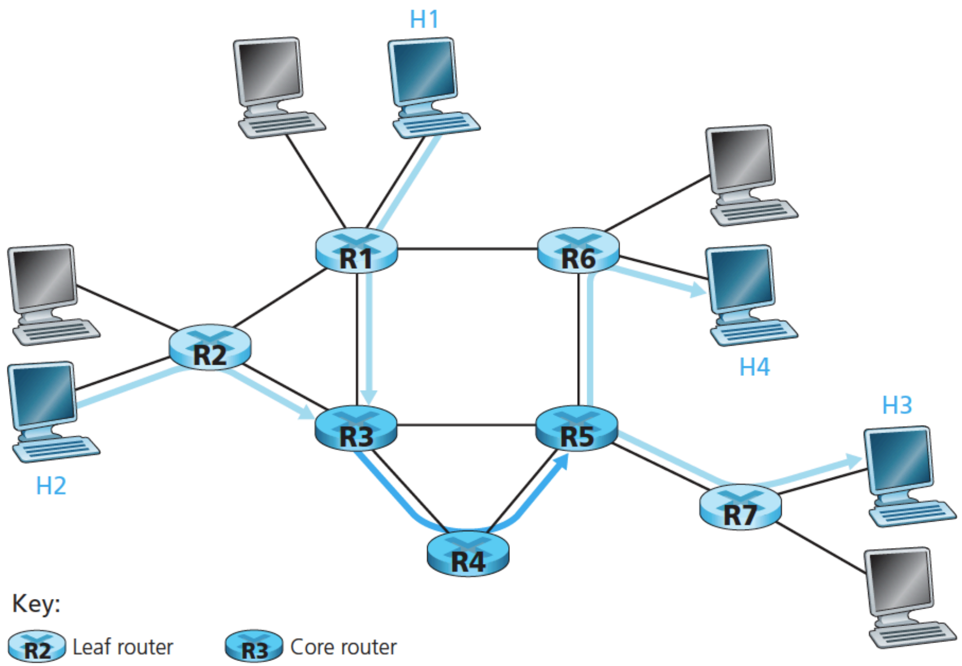
- *per-flow* traffic management
- marks packets as in-profile and out-profile

core router:

- *per class* traffic management
- buffering and scheduling based on *marking* at edge
- preference given to in-profile packets over out-of-profile packets



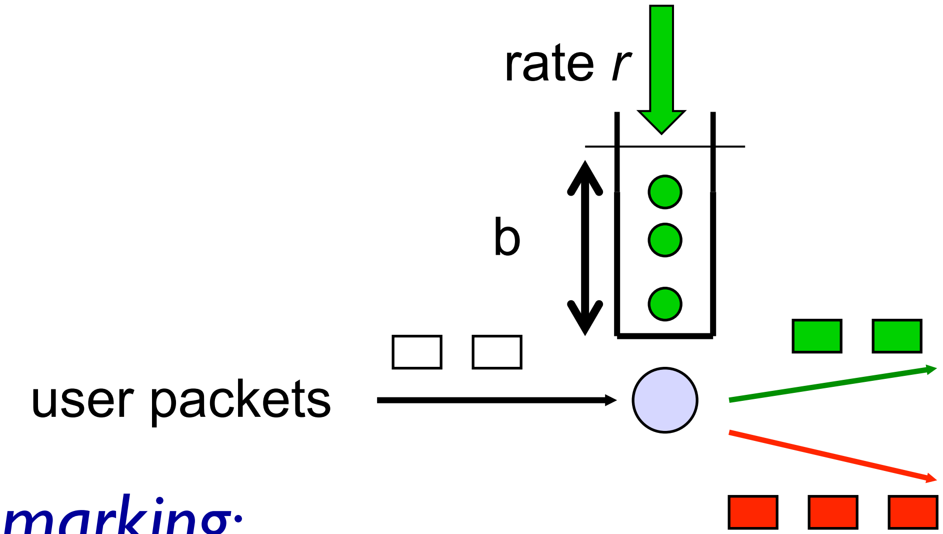
Diffserv: An Example



- Packets being sent from H1 to H3 might be marked at R1
- Packets being sent from H2 to H4 might be marked at R2

Edge-router packet marking

- *profile:* pre-negotiated rate r, bucket size b
- packet marking at edge based on *per-flow* profile

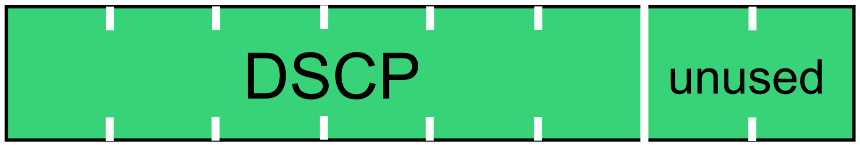


possible use of marking:

- class-based marking: packets of different classes marked differently
- intra-class marking: conforming portion of flow marked differently than non-conforming one

Diffserv packet marking: details

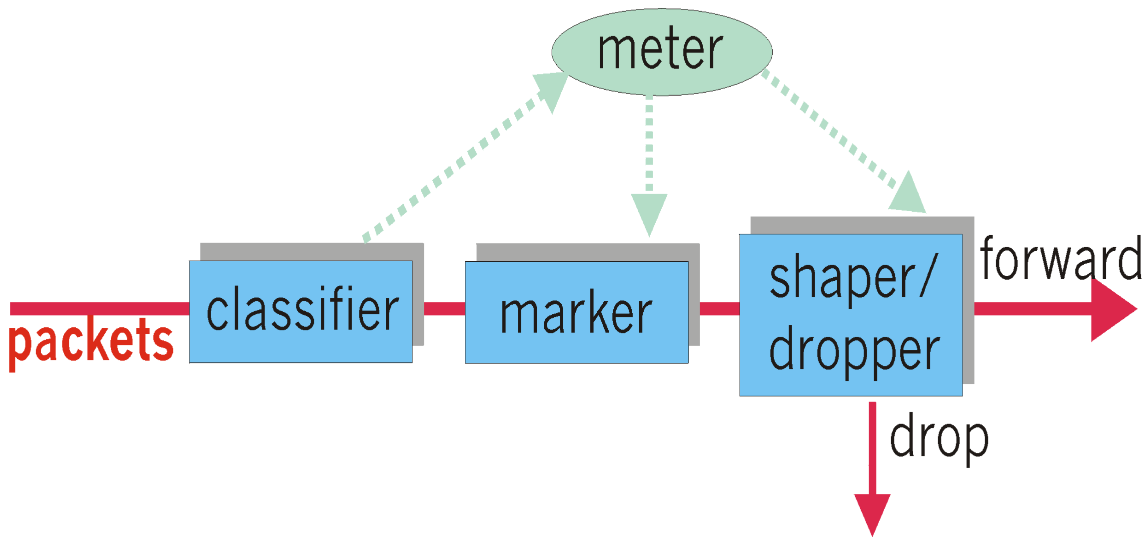
- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP)
  - determine PHB that the packet will receive
  - 2 bits currently unused



Classification, Conditioning

may be desirable to limit traffic injection rate of some class:

- user declares traffic profile (e.g., rate, burst size)
- traffic metered, shaped if non-conforming



Forwarding Per-hop Behavior (PHB)

- PHB result in a different *observable (measurable)* forwarding performance behavior
- PHB does *not* specify what mechanisms to use to ensure required PHB performance behavior
- examples:
  - class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - class A packets leave first before packets from class B

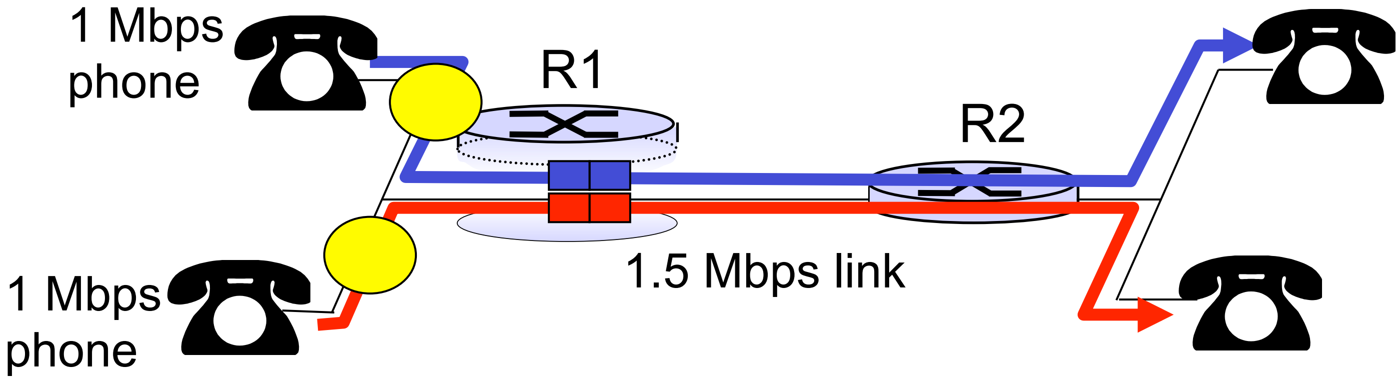
Forwarding PHB

PHBs proposed:

- *expedited forwarding:* pkt departure rate of a class equals or exceeds specified rate
  - logical link with a minimum guaranteed rate
- *assured forwarding:* 4 classes of traffic
  - each guaranteed minimum amount of bandwidth
  - each with three drop preference partitions

Per-connection QOS guarantees

- *basic fact of life:* can not support traffic demands beyond link capacity

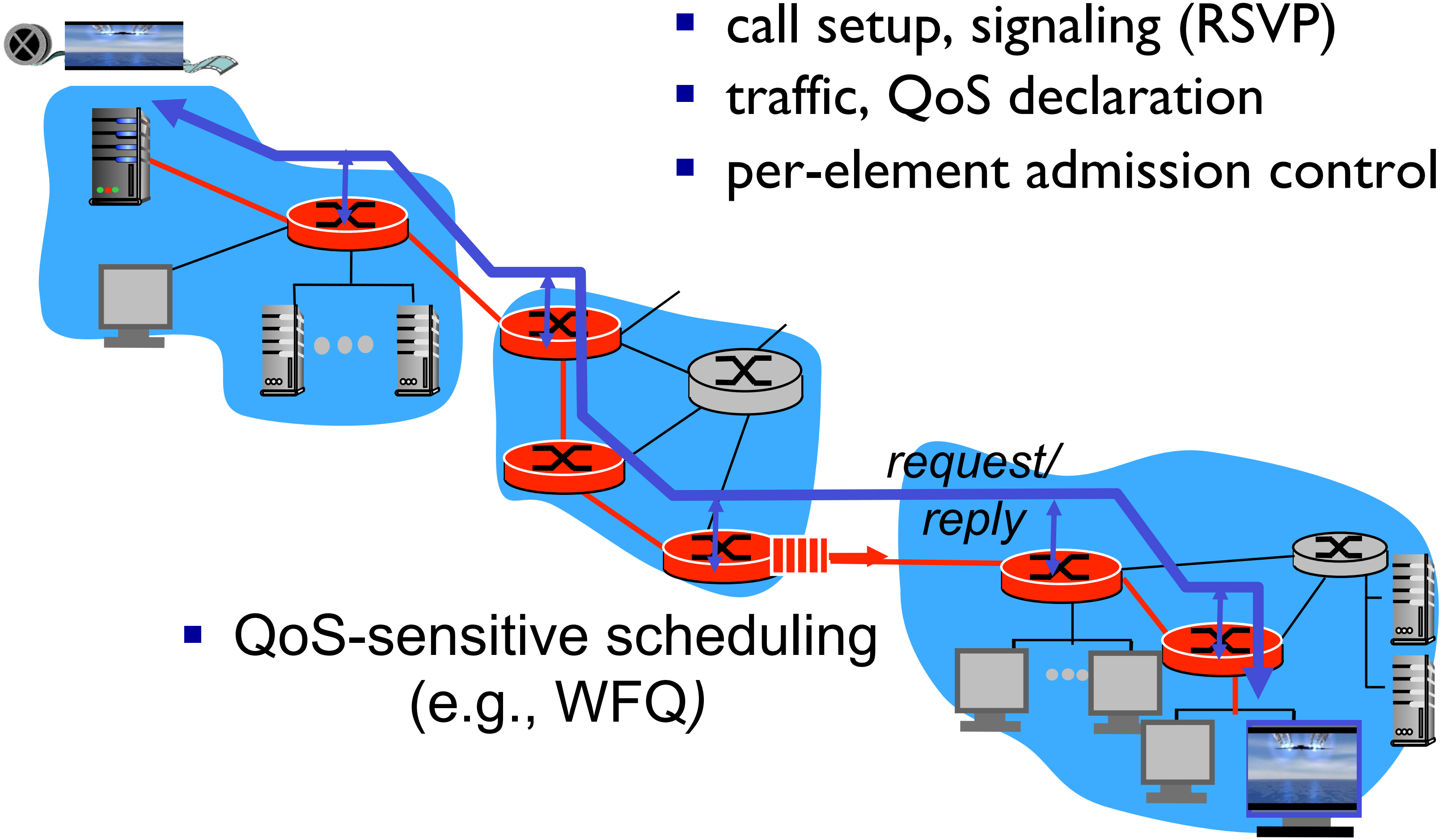


Principle 4

call admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

QoS Guarantee Scenario

- *resource reservation*
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control



- QoS-sensitive scheduling (e.g., WFQ)

IETF Integrated Services

- architecture for providing QOS guarantees in IP networks for individual application sessions
- resource reservation: routers maintain state info (a la VC) of allocated resources, QoS req's
- admit/deny new call setup requests:

Question: can newly arriving flow be admitted with performance guarantees while not violated QoS guarantees made to already admitted flows?

Call Admission

Arriving session must :

- declare its QOS requirement
  - R-spec: defines the QOS being requested
- characterize traffic it will send into network
  - T-spec: defines traffic characteristics
- signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - RSVP

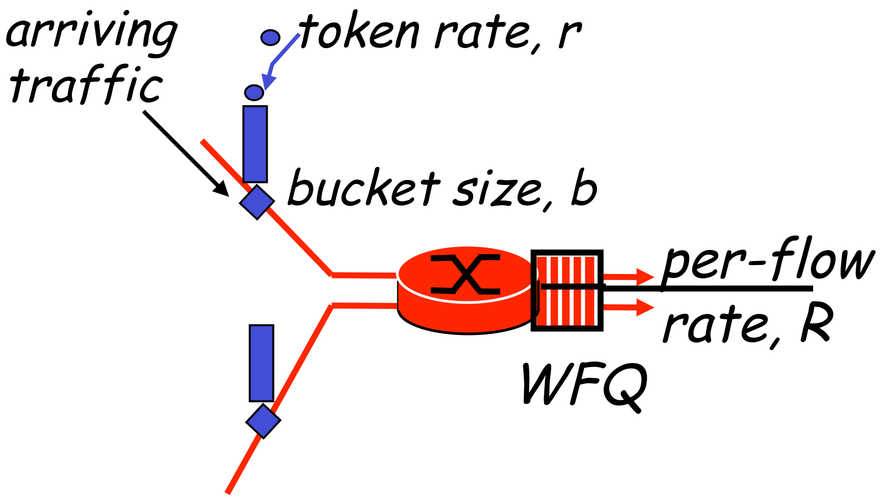
Intserv QoS: Service models [rfc2211, rfc 2212]

Guaranteed service:

- worst case traffic arrival: leaky-bucket-policed source
- simple (mathematically provable) *bound* on delay [Parekh 1992, Cruz 1988]

Controlled load service:

- *"a quality of service closely approximating the QoS that same flow would receive from an unloaded network element."*



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

Signaling in the Internet

connectionless (stateless) forwarding by IP routers + *best effort service* = *no network signaling protocols in initial IP design*

- New requirement: reserve resources along end-to-end path (end system, routers) for QoS for multimedia applications
- RSVP: Resource Reservation Protocol [RFC 2205]
  - " … allow users to communicate requirements to network in robust and efficient way." i.e., signaling !
- earlier Internet Signaling protocol: ST-II [RFC 1819]

RSVP Design Goals

1. Accommodate heterogeneous receivers (different bandwidth along paths)
2. Accommodate different applications with different resource requirements
3. Make multicast a first class service, with adaptation to multicast group membership
4. Leverage existing multicast/unicast routing, with adaptation to changes in underlying unicast, multicast routes
5. Control protocol overhead to grow (at worst) linear in # receivers
6. Modular design for heterogeneous underlying technologies

RSVP: does not…

- specify how resources are to be reserved
  - rather: a mechanism for communicating needs
- determine routes packets will take
  - that's the job of routing protocols
  - signaling decoupled from routing
- interact with forwarding of packets
  - separation of control (signaling) and data (forwarding) planes

Chapter 7: Summary

Principles

- classify multimedia applications
- identify network services applications need
- making the best of best effort service

Protocols and Architectures

- specific protocols for best-effort
- mechanisms for providing QoS
- architectures for QoS
  - multiple classes of service
  - QoS guarantees, admission control

RSVP: overview of operation

- Senders, receiver join a multicast group
  - done outside of RSVP
  - senders need not join group
- Sender-to-network signaling
  - *path message:* make sender presence known to routers
  - path teardown: delete sender's path state from routers
- Receiver-to-network signaling
  - *reservation message:* reserve resources from sender(s) to receiver
  - reservation teardown: remove receiver reservations
- Network-to-end-system signaling
  - path error
  - reservation error