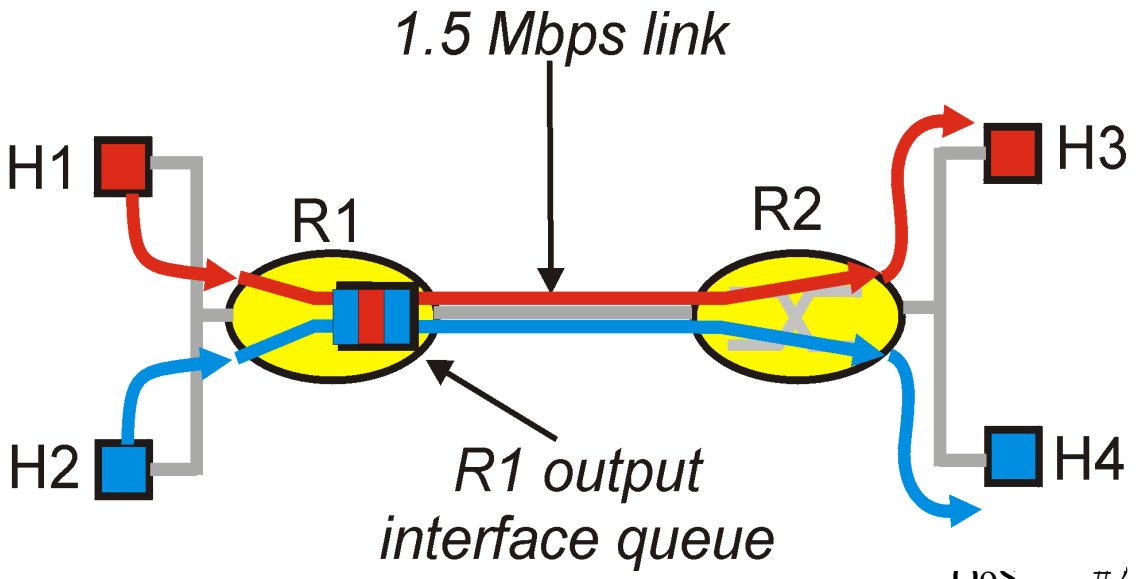
Quality of Service Support

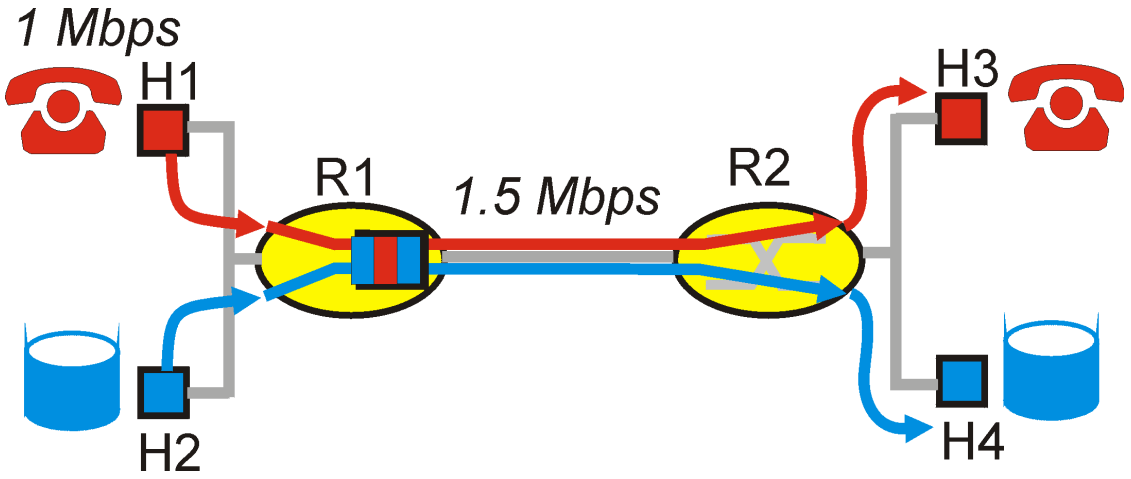
QOS in IP Networks

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



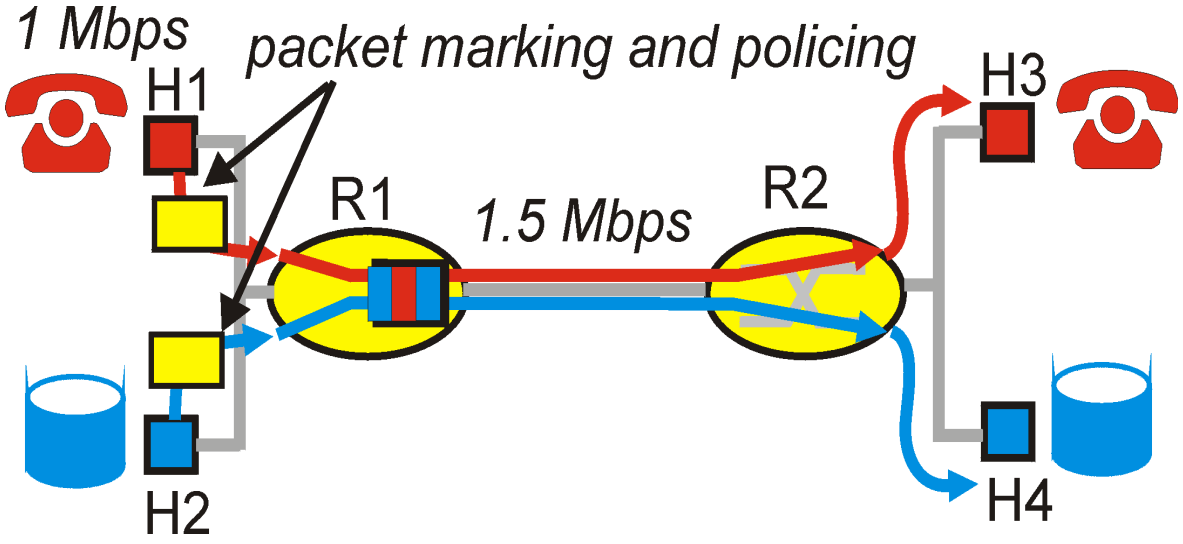
Principles for QOS Guarantees

- 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**



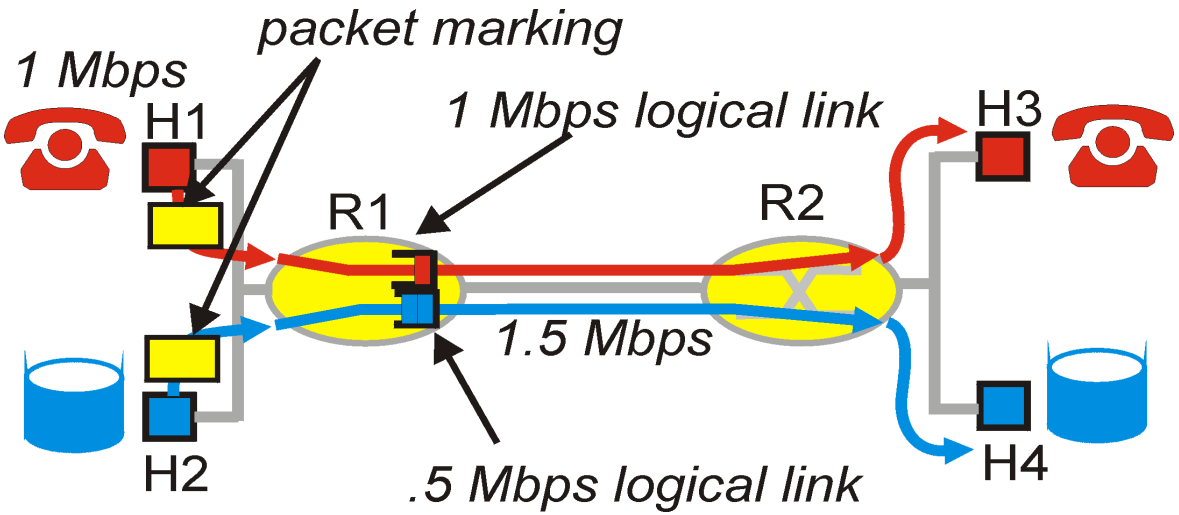
Principles for QOS Guarantees (more)

- 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:



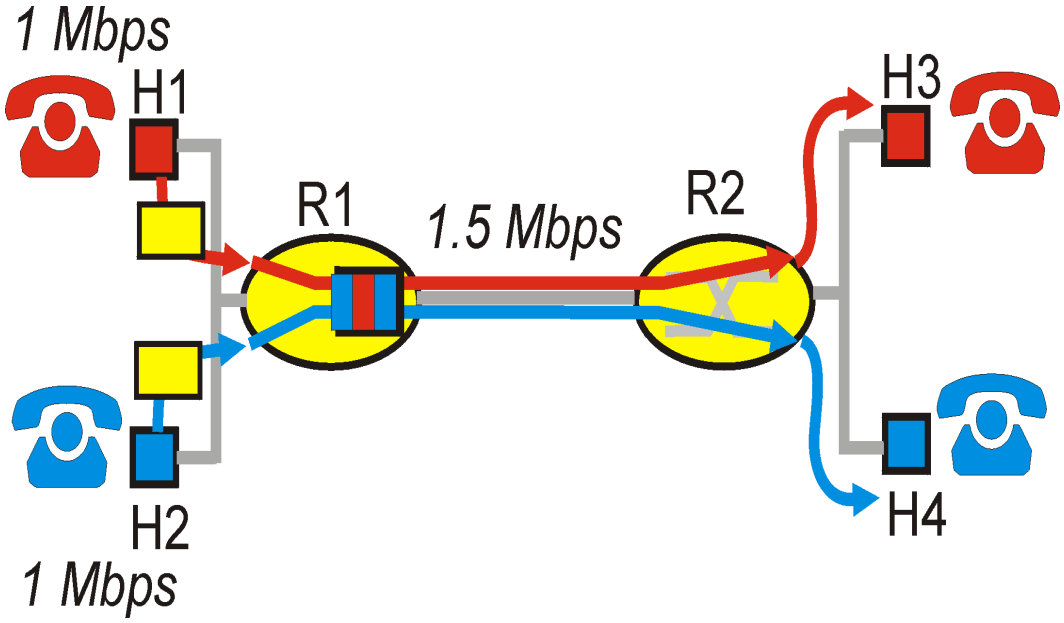
Principles for QOS Guarantees (more)

- 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**

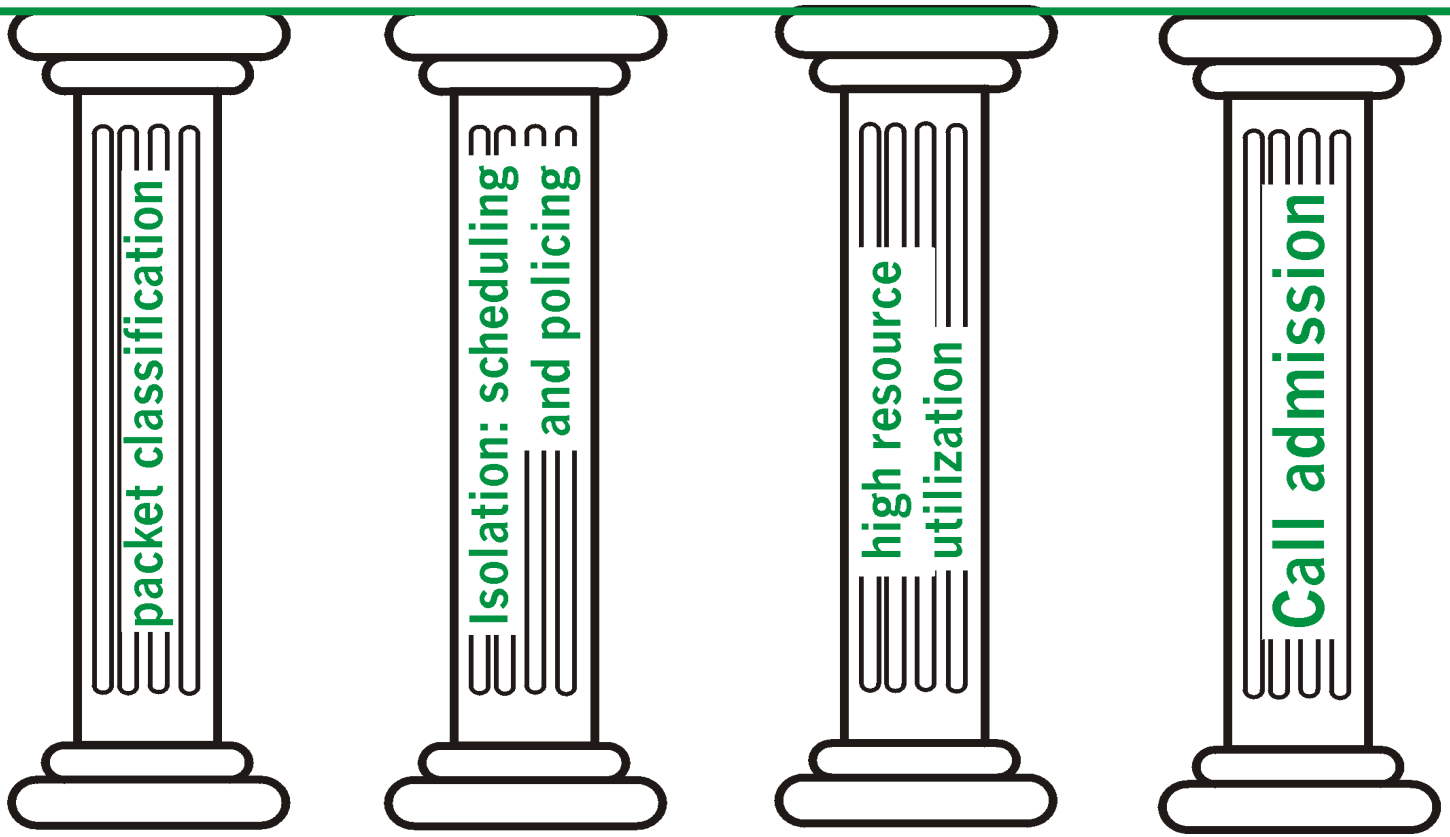


Principles for QOS Guarantees (more)

- Cannot support traffic beyond link capacity
  - Two phone calls each requests 1 Mbps
- **PRINCIPLE 4: Need a Call Admission Process; application flow declares its needs, network may block call if it cannot satisfy the needs**



QoS for networked applications

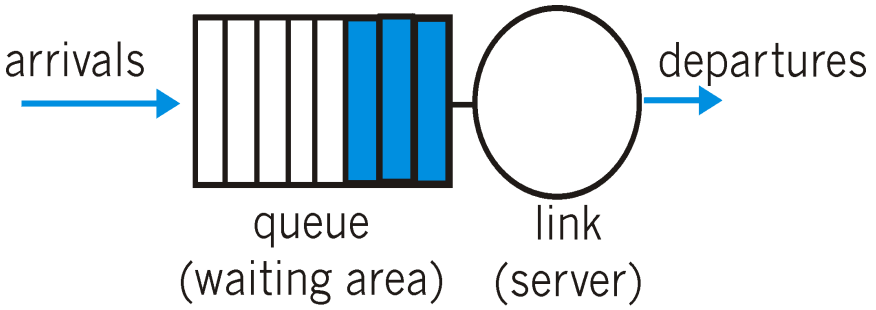


Building blocks

- Scheduling
  - Active Buffer Management
- Traffic Shaping
  - Leaky Bucket
  - Token Bucket
- Modeling
  - The (σ,ρ) Model
  - WFQ and delay guarantee
- Admission Control
  - QoS Routing

Scheduling: How Can Routers Help

- Scheduling: choosing the next packet for transmission
  - FIFO/Priority Queue
  - Round Robin/ DRR
  - Weighted Fair Queuing



- Packet dropping:
  - not drop-tail
  - not only when buffer is full
    - Active Queue Management

- Congestion signaling
  - Explicit Congestion Notification (ECN)

Buffer Size

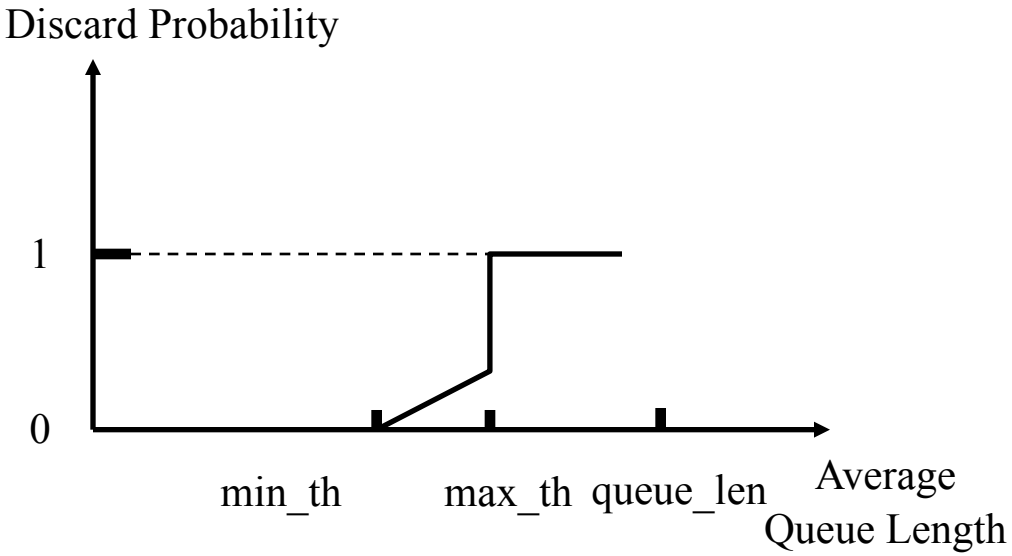
- Why not use infinite buffers?
  - no packet drops!
- Small buffers:
  - often drop packets due to bursts
  - but have small delays
- Large buffers:
  - reduce number of packet drops (due to bursts)
  - but increase delays
- Can we have the best of both worlds?

Random Early Detection (RED)

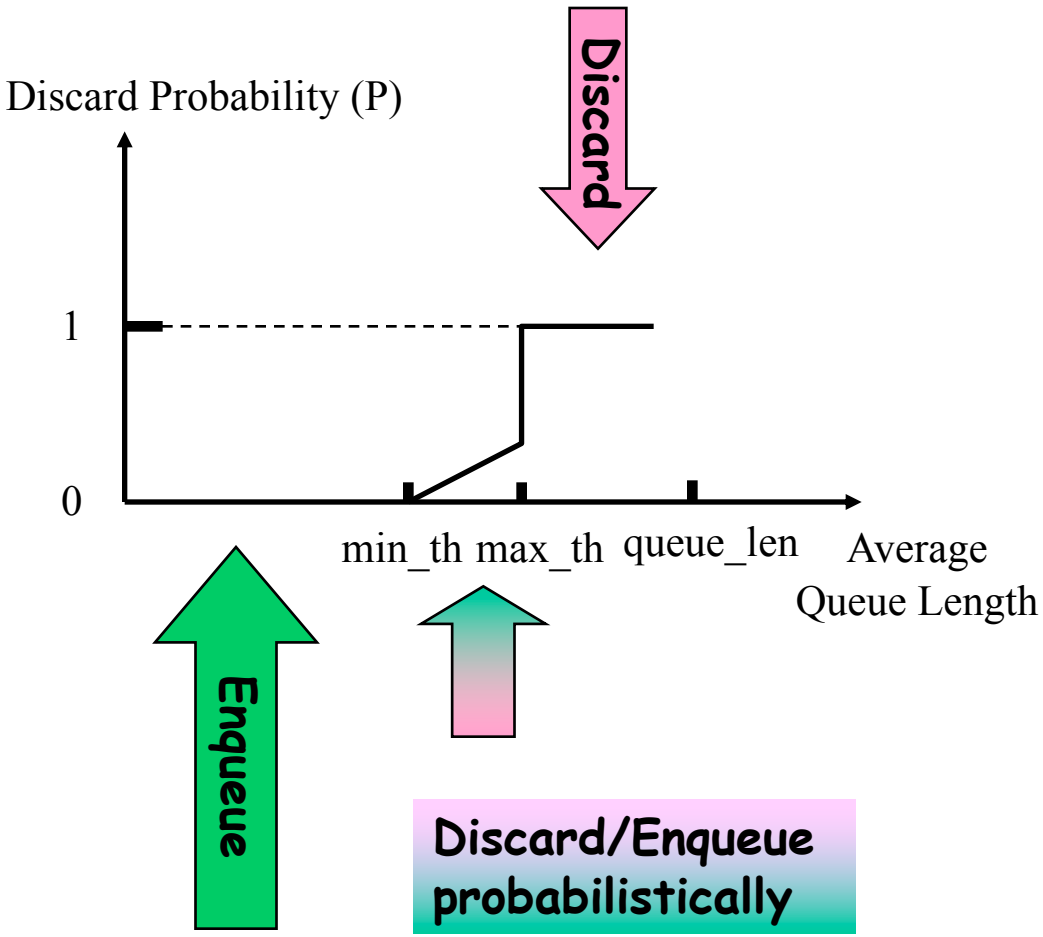
- Basic premise:
  - router should signal congestion when the queue first starts building up (by dropping a packet)
  - but router should give flows time to reduce their sending rates before dropping more packets
  - Note: when RED is coupled with ECN, the router can simply mark a packet instead of dropping it
- Therefore, packet drops should be:
  - early: don't wait for queue to overflow
  - random: don't drop all packets in burst, but space them

RED

- FIFO scheduling
- Buffer management:
  - Probabilistically discard packets
  - Probability is computed as a function of average queue length (why average?)



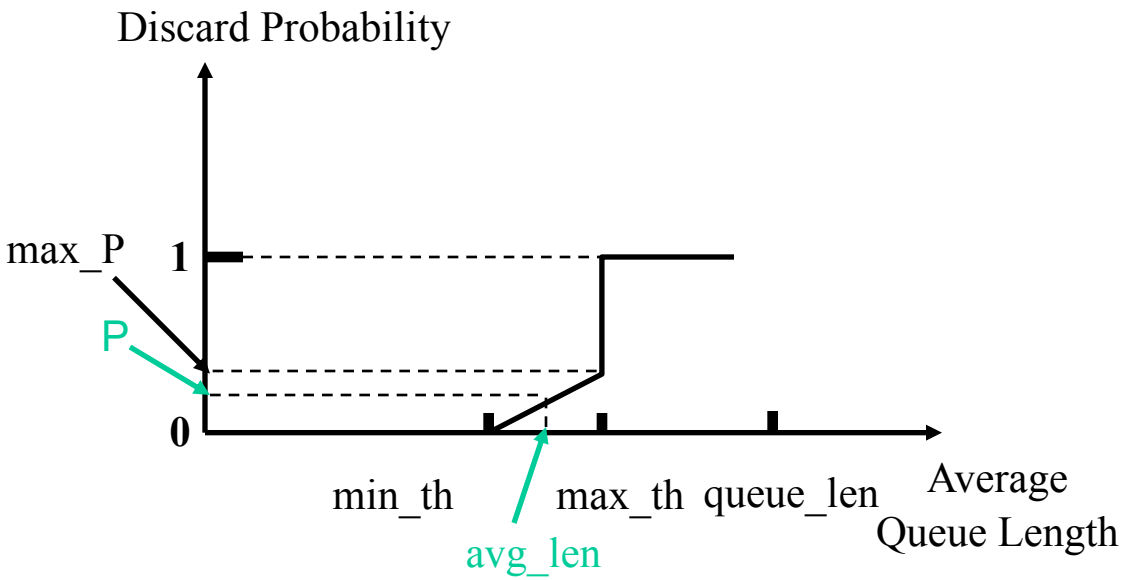
RED (cont'd)


Discard Probability (P)
Discard
1
0
min_th
max_th
queue_len
Average
Queue Length
Enqueue
Discard/Enqueue
probabilistically

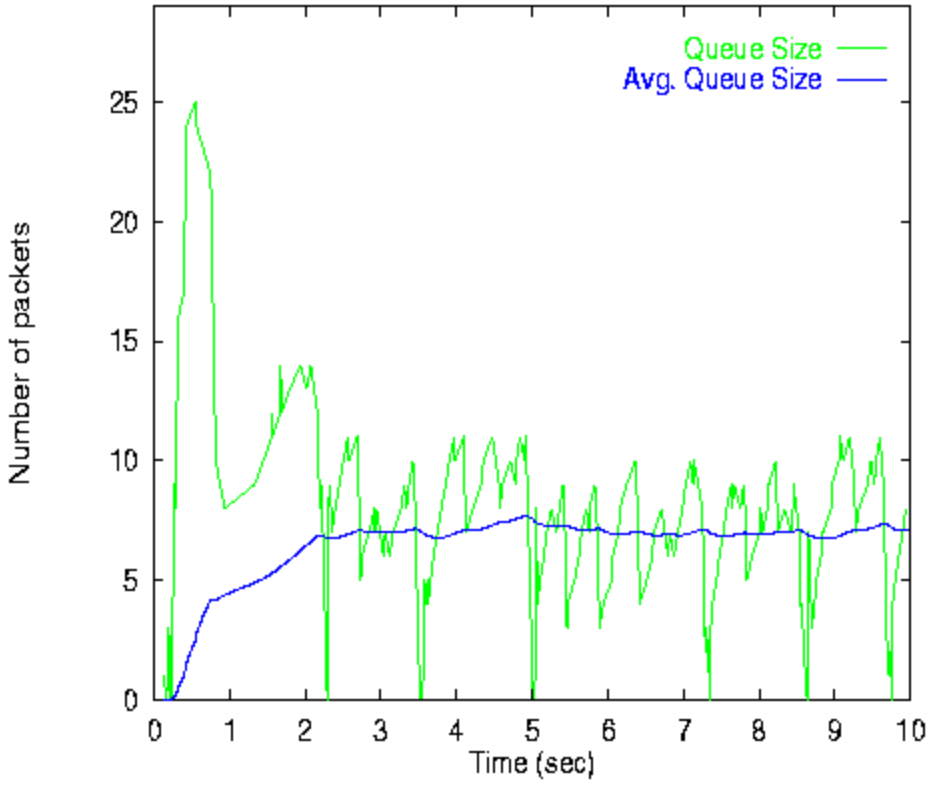
RED (cont'd)

- Setting the discard probability P:

$$P = \max\_P \frac{\text{avg\_len - min\_th}}{\text{max\_th - min\_th}}$$



Average vs Instantaneous Queue



RED and TCP

- Sequence of actions (Early drop)
  - Duplicate Acks
  - Fast retransmit
    - Session recovers
  - Lower source rate
- Fairness in drops
  - Bursty versus non-Bursy
  - Probability of drop depends on rate.
- Disadvantages
  - Many additional parameters
  - Increasing the loss

RED Summary

- Basic idea is sound, but does not always work well
  - Basically, dropping packets, early or late is a bad thing
- High network utilization with low delays when flows are long lived
- Average queue length small, but capable of absorbing large bursts
- Many refinements to basic algorithm make it more adaptive
  - requires less tuning
- Does not work well for short lived flows (like Web traffic)
  - Dropping packets in an already short lived flow is devastating
- Better to mark ECN instead of dropping packets
  - ECN not widely supported

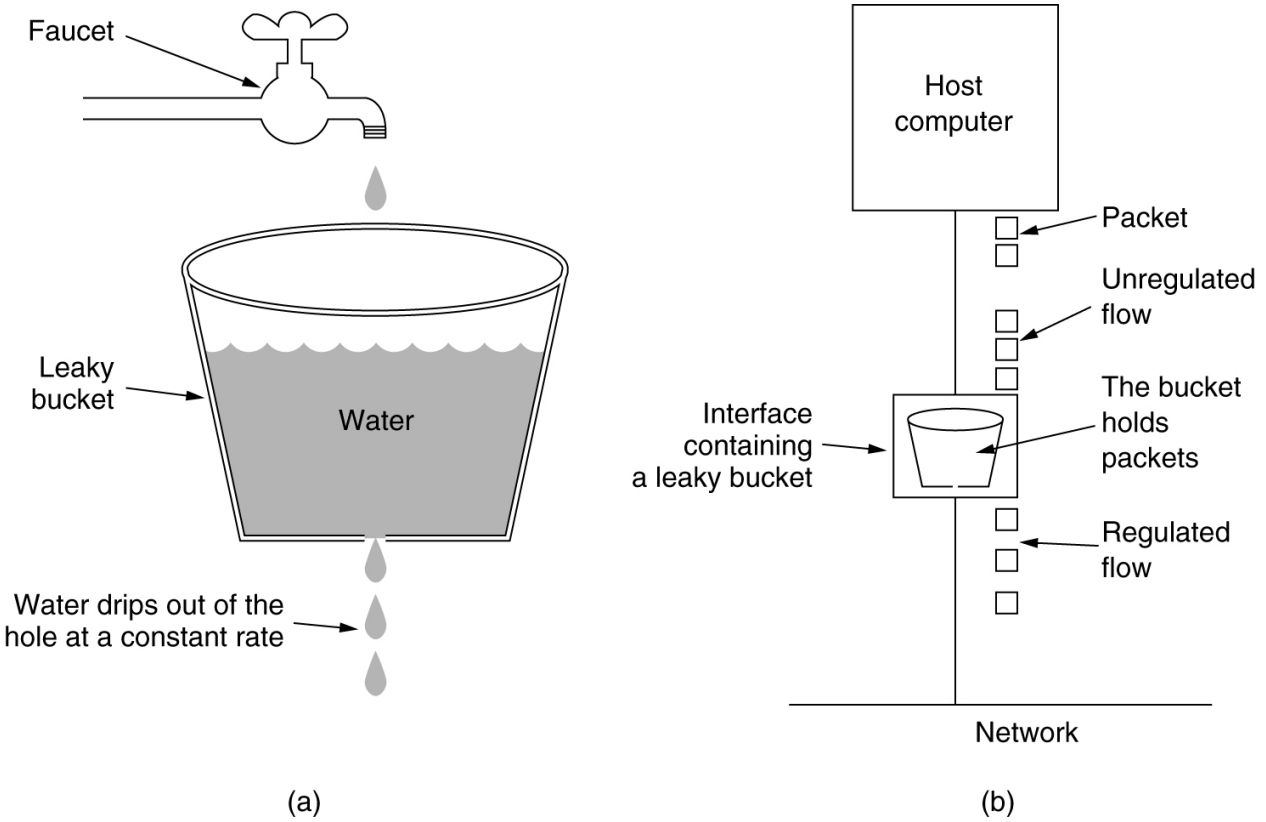
Traffic Shaping

- Traffic shaping controls the *rate* at which packets are sent (not just how many).
  - Used in ATM and Integrated Services networks.
- At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).
- Two traffic shaping algorithms are:
  - Leaky Bucket
  - Token Bucket

The Leaky Bucket Algorithm

- The **Leaky Bucket Algorithm**
  - used to control rate in a network.
  - It is implemented as a single-server queue
    - with constant service time.
  - If the bucket (buffer) overflows then packets are discarded.
- Leaky Bucket (parameters r and B):
  - Every r time units: send a packet.
  - For an arriving packet
    - If queue not full (less than B) then enqueue
- Note that the output is a "perfect" constant rate.

The Leaky Bucket Algorithm



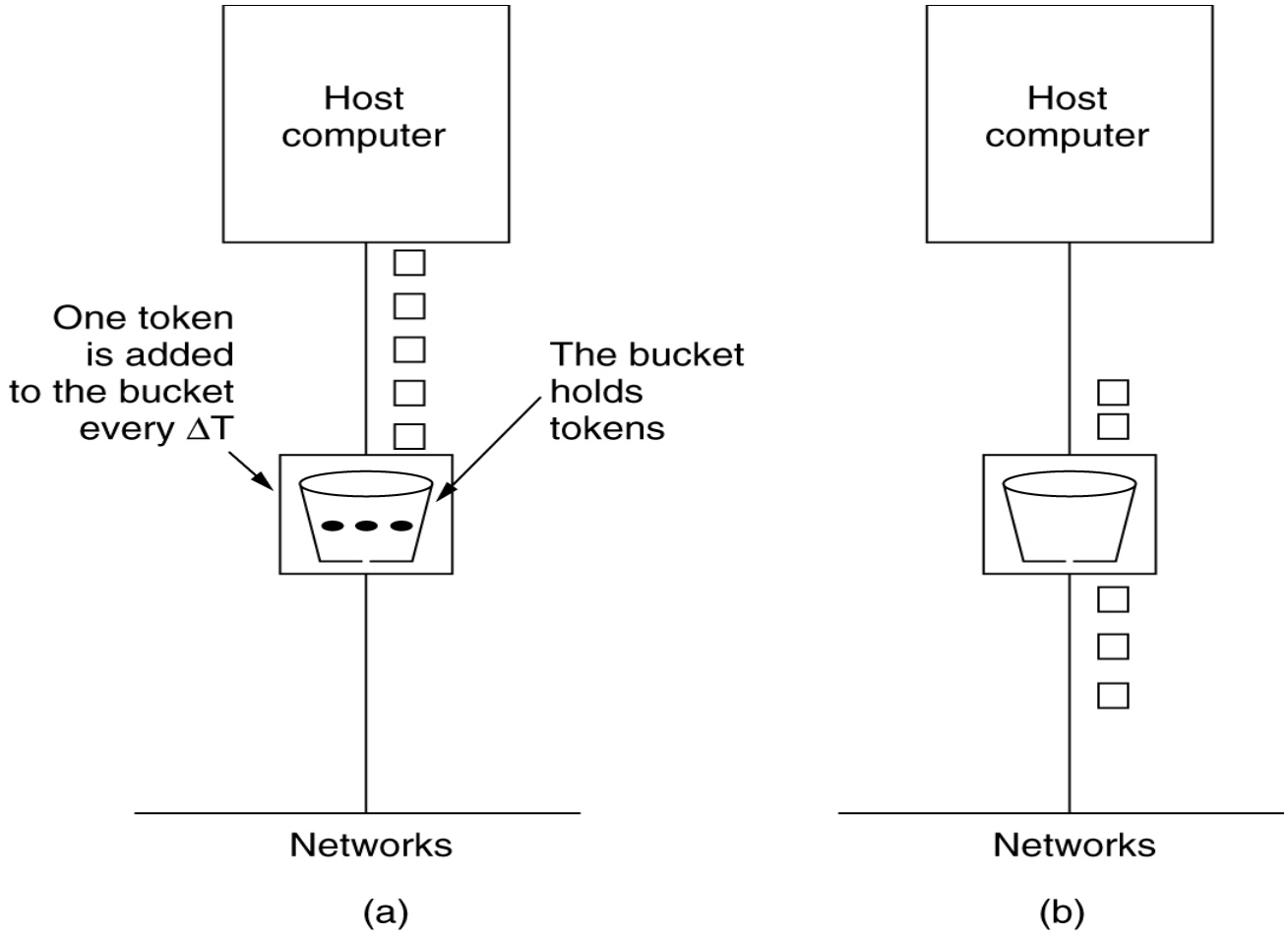
(a) A leaky bucket with water. (b) a leaky bucket with packets.

Token Bucket Algorithm

- Highlights:
  - The bucket holds tokens.
  - To transmit a packet, we "use" one token.
- Allows the output rate to vary,
  - depending on the size of the burst.
  - In contrast to the Leaky Bucket
- Granularity
  - Packets (or bits)

- Token Bucket (r, MaxTokens):
  - Generate a token every r time units
    - If number of tokens more than MaxToken, reset to MaxTokens.
  - For an arriving packet: enqueue
  - While buffer not empty and there are tokens:
    - send a packet and discard a token

The Token Bucket Algorithm

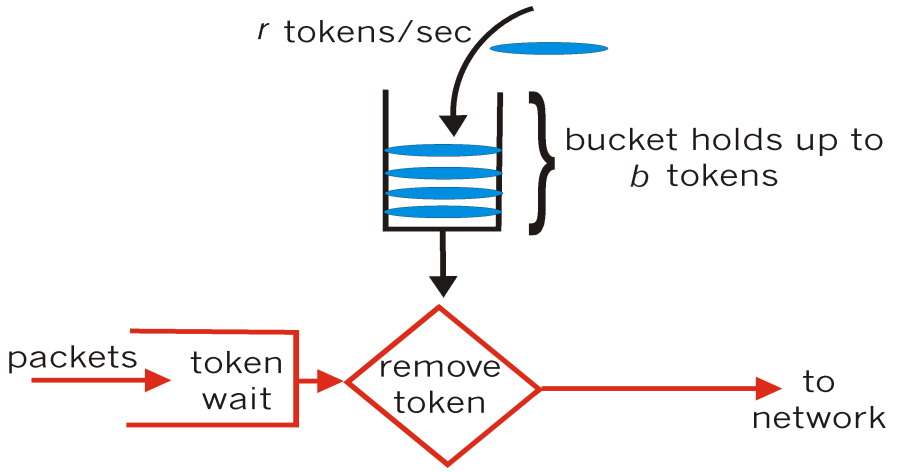


(a) Before. (b) After.

Token bucket example

arrival	queue	Token bucket	sent
p1 (5)	-	0	-
p2 (2)	p1	3	-
p3 (1)	p2	6-5=1	p1
		4-2-1=1	p3,p2
		4	
		6	

parameters:
MaxTokens=6
1/r=3 (=3 token/time)



Leaky Bucket vs Token Bucket

Leaky Bucket

- Discard:
  - Packets
- Rate:
  - fixed rate (perfect)
- Arriving Burst:
  - Waits in bucket

Token Bucket

- Discard:
  - Tokens
  - Packet management separate
- Rate:
  - Average rate
  - Bursts allowed
- Arriving Burst:
  - Can be sent immediately

The (σ,ρ) Model

- Parameters:
  - The average rate is ρ.
  - The maximum burst is σ.
- (σ,ρ) Model:
  - Over an interval of length *t*,
  - the number of packets/bits that are admitted
  - is less than or equal to *(σ+ρt)*.
  - Composing flows (σ_1,ρ_1) & (σ_2,ρ_2)
    - Resulting flow $(\sigma_1+\sigma_2,\rho_1+\rho_2)$
- Token Bucket Algorithm:
  - σ = MaxTokens & ρ=1/r per time unit
- Leaky Bucket Algorithm
  - σ = 0 & ρ= =1/r per time unit

Using (σ,ρ) Model for admission Control

- What does a router need to support streams: (σ_1,ρ_1) … (σ_k,ρ_k)
  - Buffer size $B > \Sigma\ \sigma_i$
  - Rate $R > \Sigma\ \rho_i$
- Admission Control (at the router)
  - Can support (σ_k,ρ_k) if
  - Enough buffers and bandwidth
    - $R > \Sigma\ \rho_i$ and $B > \Sigma\ \sigma_i$

Delay Bounds: WFQ

- Recall: $work_S(i, a,b)$
  - # bits transmitted for flow i in time [a,b] by policy S.

- Theorem (Parekh-Gallager: Single link):
  - Assume maximum packet size L_{max}
  - Then for any time t:

  $work_{GPS}(i,1,t) - work_{WFQ}(i, 1,t) \leq L_{max}$

- Corollary:
  - For any packet p and link rate R
    - Let Time(p,S) be its completion time in policy S
    - Then Time(p,WFQ)-Time(p,GPS) $\leq L_{max}/R$

Parekh-Gallagher theorem

Suppose a given connection is (σ,ρ) constrained, has maximal packet size L, and passes through K WFQ schedulers, such that in the ith scheduler

- there is total rate $r(i)$
- from which the connection gets $g(i)$.

Let g be the minimum over all $g(i)$, and suppose all packets are at most $L_{\max}$ bits long. Then

$$\text{end - to - end delay} \le \frac{\sigma}{g} + \sum_{i=1}^{k} \frac{L}{g(i)} + \sum_{i=1}^{k} \frac{L}{r(i)}$$

P-G theorem: Interpretation

$$\text{end-to-end delay} \le \frac{\sigma}{g} + \sum_{i=1}^{k} \frac{L}{g(i)} + \sum_{i=1}^{k} \frac{L}{r(i)}$$

Delay of last packet of a burst. Only in bottleneck node

GPS term

store&forward penalty

WFQ lag behind GPS: each node

GPS to WFQ correction

Significance

- WFQ can provide end-to-end delay bounds
- So WFQ provides both fairness and performance guarantees
- Bound holds regardless of cross traffic behavior
- Can be generalized for networks where schedulers are variants of WFQ, and the link service rate changes over time

Fine Points

- To get a delay bound, need to pick g
  - the lower the delay bound, the larger g needs to be
  - large g means exclusion of more competitors from link
- Sources must be leaky-bucket regulated
  - but choosing leaky-bucket parameters is problematic
- WFQ couples delay and bandwidth allocations
  - low delay requires allocating more bandwidth
  - wastes bandwidth for low-bandwidth low-delay sources

Approaches to QoS

Integrated Services

- Network wide control
- Admission Control
- Absolute guarantees
- Traffic Shaping
- Reservations
  - RSVP

Differentiated Services

- Router based control
  - Per hop behavior
- Resolves contentions
  - Hot spots
- Relative guarantees
- Traffic policing
  - At entry to network

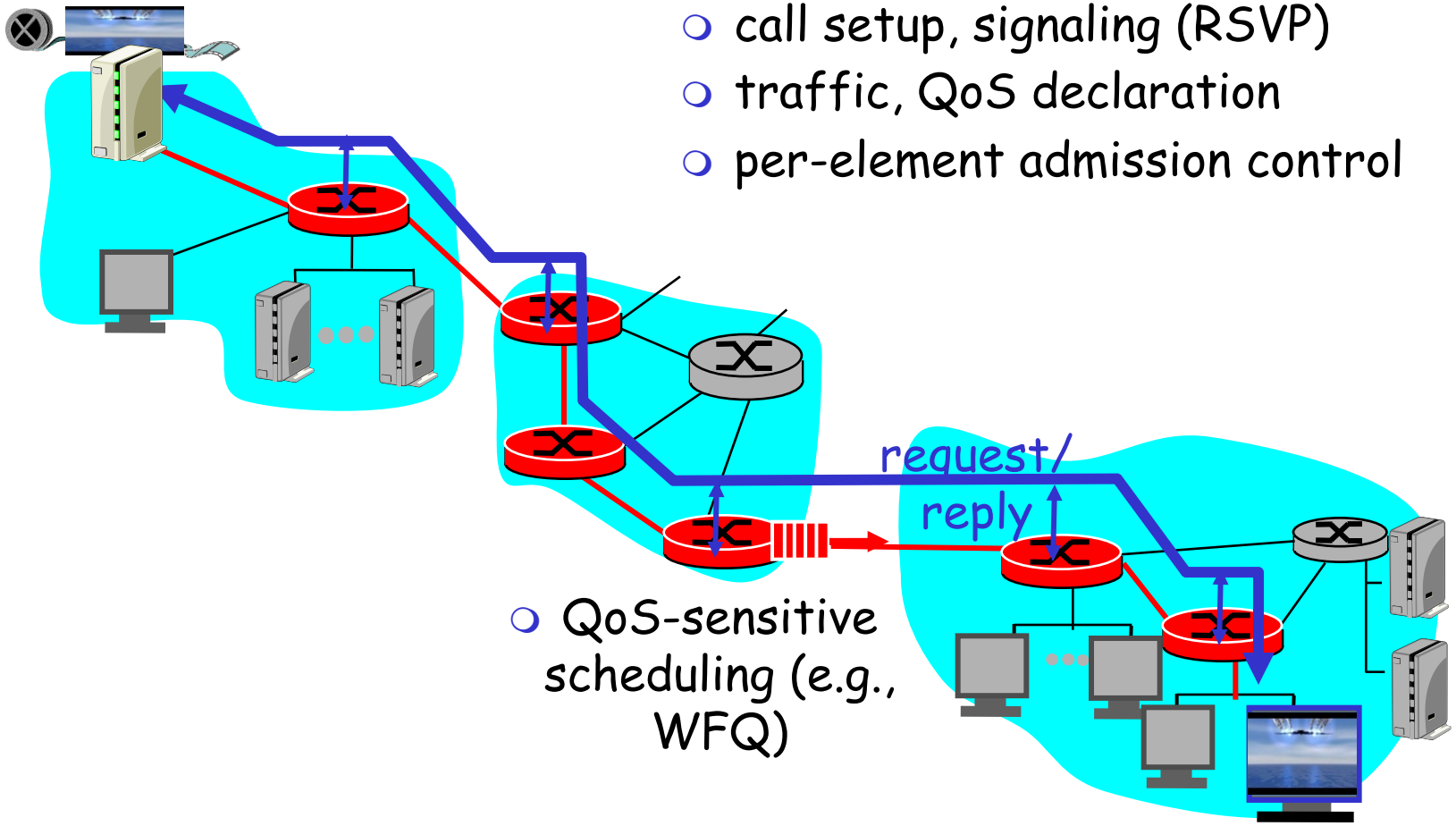
IETF Integrated Services

- 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?

Intserv: QoS guarantee scenario

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



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

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

RSVP request (T-Spec)

- A token bucket specification
  - bucket size, b
  - token rate, r
  - the packet is transmitted onward only if the number of tokens in the bucket is at least as large as the packet
- peak rate, p
  - p > r
- maximum packet size, M
- minimum policed unit, m
  - All packets less than m bytes are considered to be m bytes
  - Reduces the overhead to process each packet
  - Bound the bandwidth overhead of link-level headers

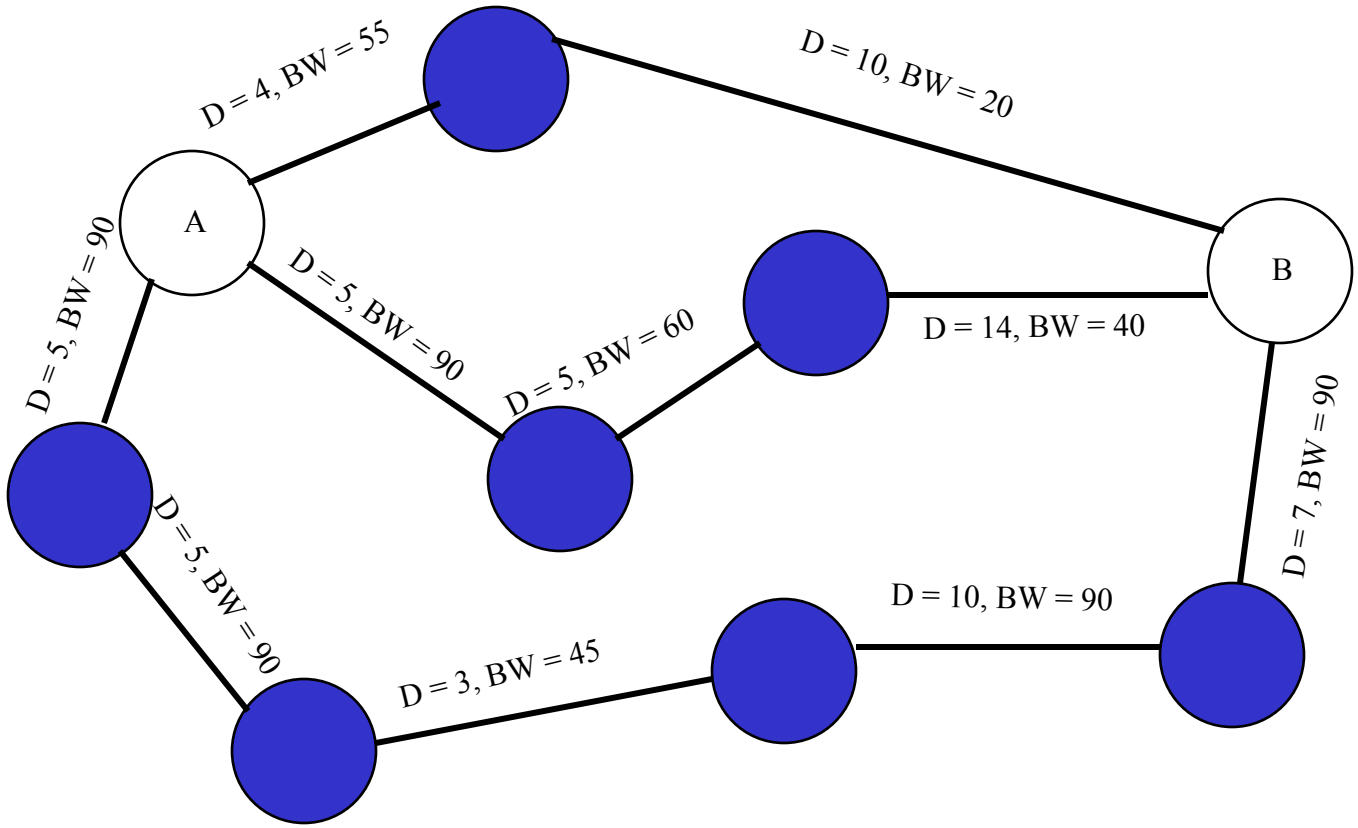
RSVP request (R-spec)

- An indication of the QoS control service requested
  - Controlled-load service and Guaranteed service
- For Controlled-load service
  - Simply a Tspec
- For Guaranteed service
  - A Rate (R) term, the bandwidth required
    - $R \geq r$, extra bandwidth will reduce queuing delays
  - A Slack (S) term
    - The difference between the desired delay and the delay that would be achieved if rate R were used
    - With a zero slack term, each router along the path must reserve R bandwidth
    - A nonzero slack term offers the individual routers greater flexibility in making their local reservation
    - Number decreased by routers on the path.

QoS Routing: Multiple constraints

- A request specifies the desired **QoS requirements**
  - e.g., BW, Delay, Jitter, packet loss, path reliability etc
- Three (main) type of constraints:
  - Additive: e.g., delay
  - Multiplicative: e.g., loss rate
  - Maximum (or Minimum): e.g., Bandwidth
- Task
  - Find a (min cost) path which satisfies the constraints
  - if no feasible path found, **reject** the connection
  - Generally, multiple constraints is HARD computationally.
- Simple case:
  - BW and delay

Example of QoS Routing



Constraints: Delay (D) < 25, Available Bandwidth (BW) > 30

IETF Differentiated Services

Concerns with Intserv:

- Scalability: signaling, maintaining per-flow router state difficult with large number of flows
- Flexible Service Models: Intserv has only two classes. Also want "qualitative" service classes
  - "behaves like a wire"
  - relative service distinction: Platinum, Gold, Silver

Diffserv approach:

- simple functions in network core, relatively complex functions at edge routers (or hosts)
- Don't define service classes, provide functional components to build service classes

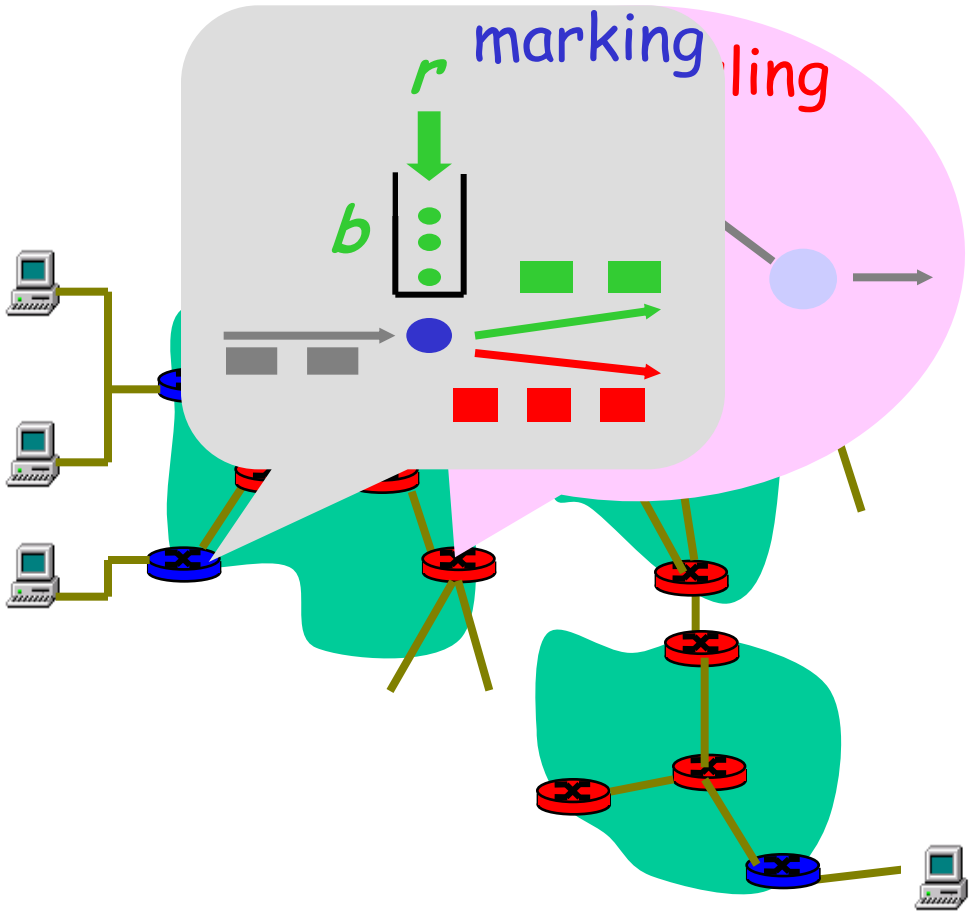
Diffserv Architecture

Edge router:

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

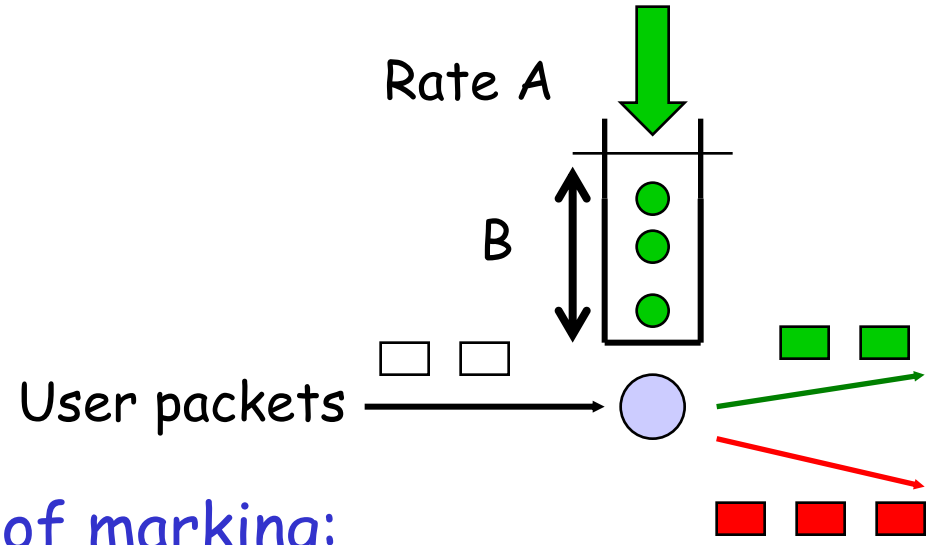
Core router:

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



Edge-router Packet Marking

- profile: pre-negotiated rate A, and token bucket size B
- packet marking at edge based on per-flow profile

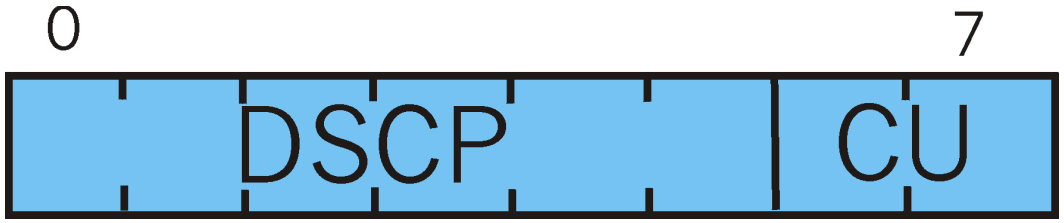


Possible usage of marking:

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

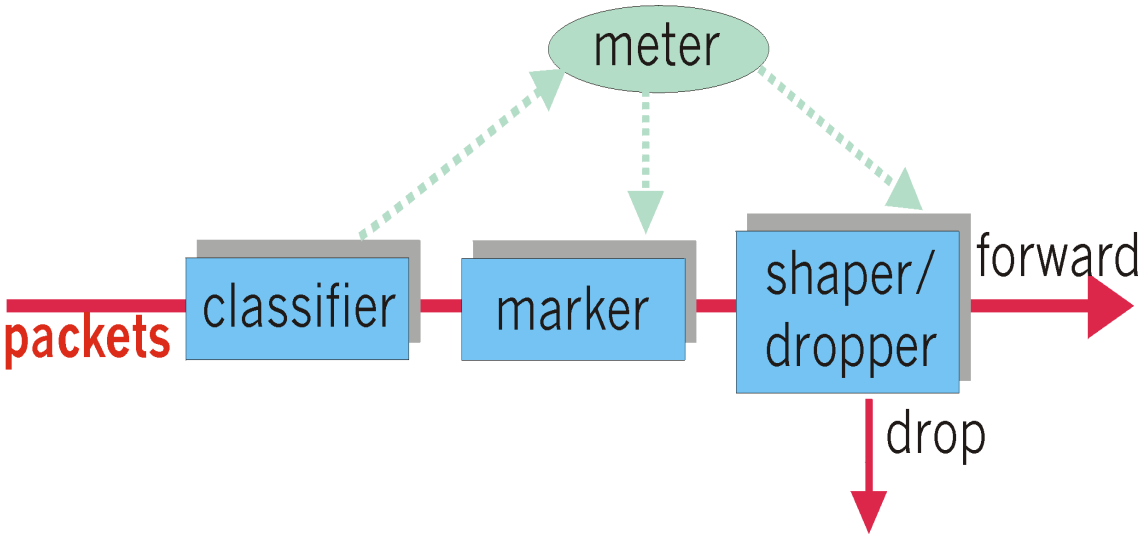
Classification and Conditioning

- 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
- 2 bits are currently unused



Classification and Conditioning

- It may be desirable to limit traffic injection rate of some class; user declares traffic profile (eg, rate and burst size); traffic is metered and shaped if non-conforming



Forwarding (PHB)

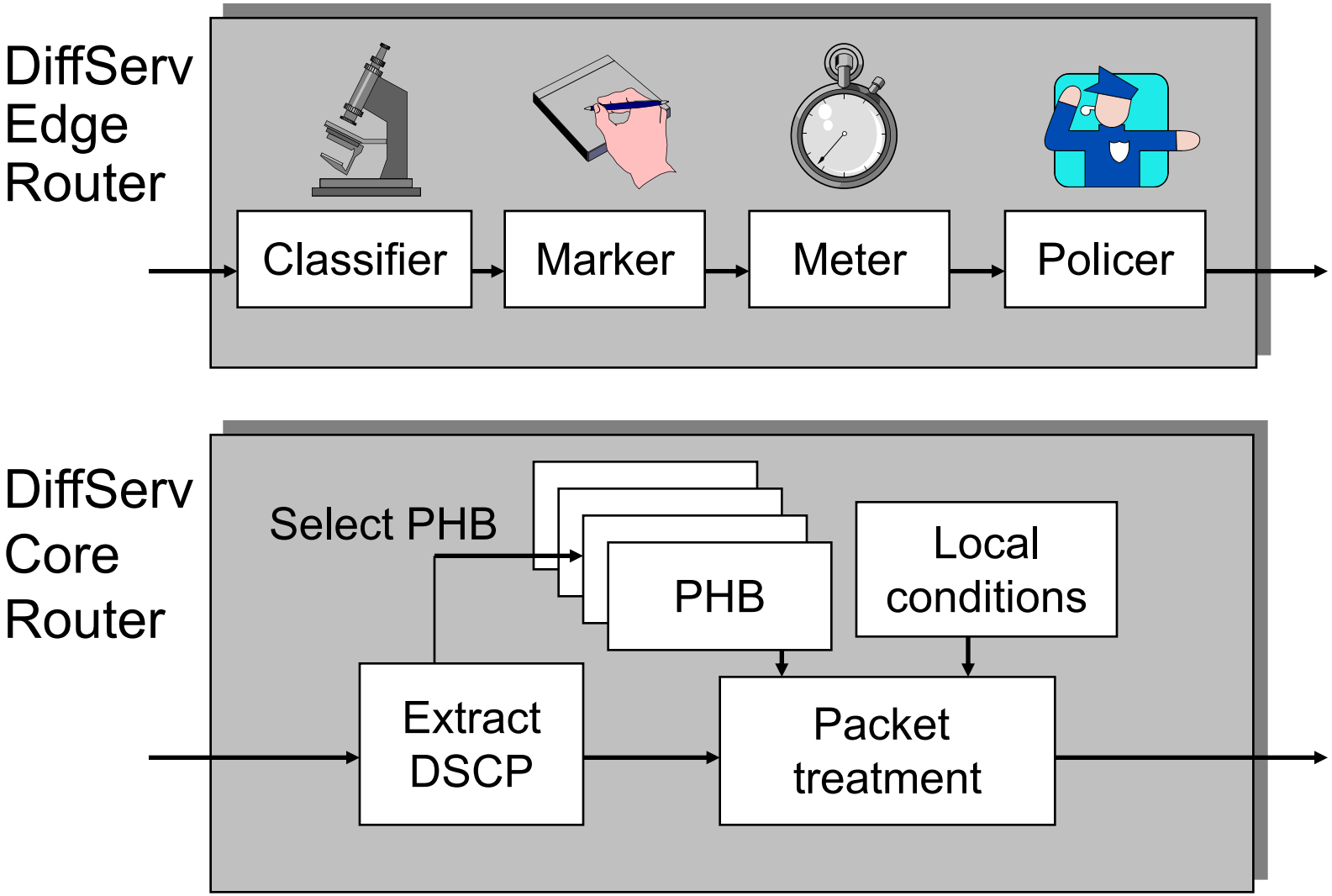
- Per Hop Behavior (PHB) result in a different observable (measurable) forwarding performance behavior
- PHB does not specify what mechanisms to use to ensure required PHB performance behavior
- Examples:
  - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - Class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

Forwarding (PHB)

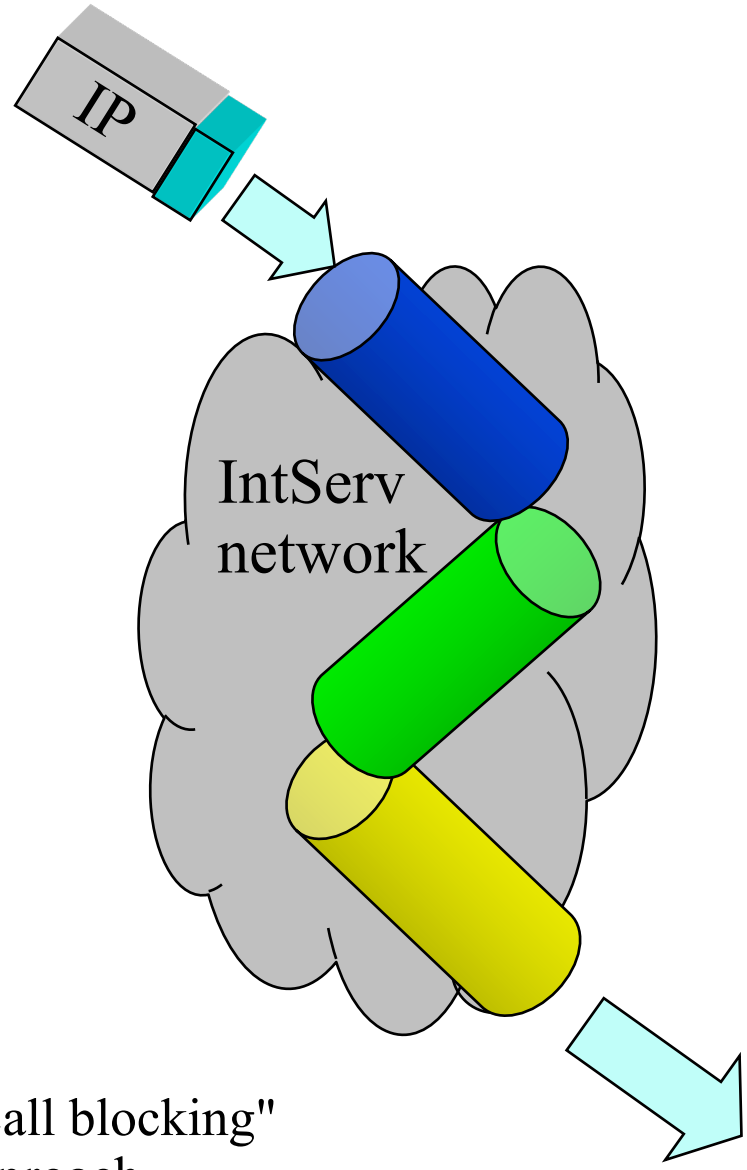
PHBs being developed:

- Expedited Forwarding: pkt departure rate of a class equals or exceeds specified rate
  - logical link with a minimum guaranteed rate
    - Premium service
  - DSCP = 101110 (46)
- Assured Forwarding: 4 classes of traffic
  - each guaranteed minimum amount of bandwidth
  - each with three drop preference partitions
    - Gold, silver, bronze

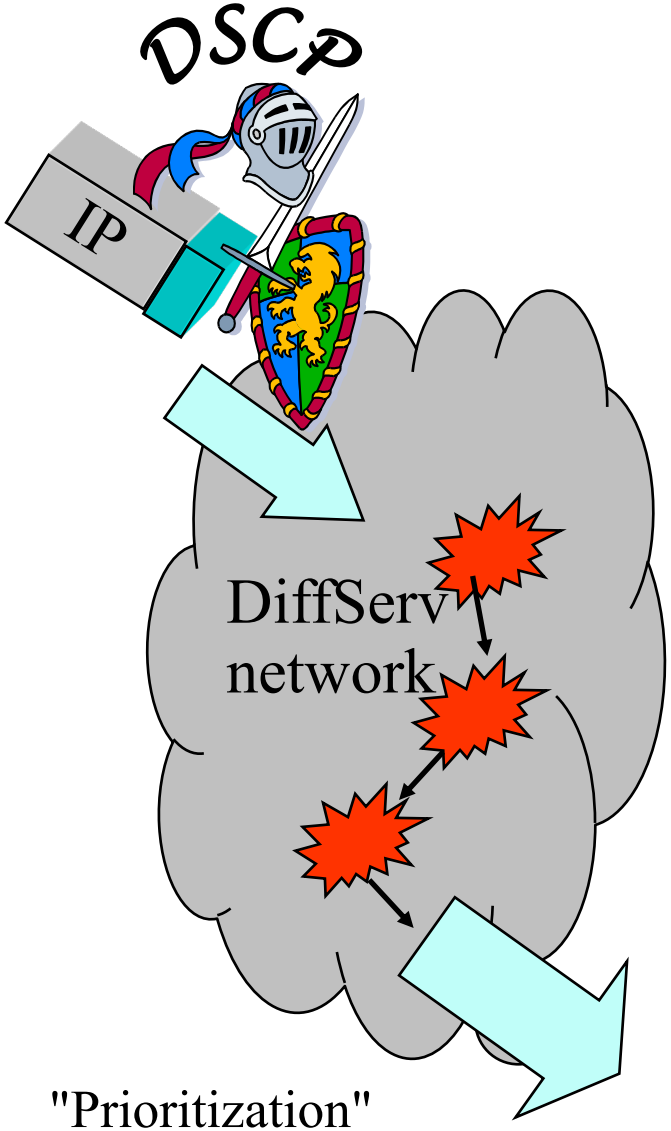
DiffServ Routers



IntServ vs. DiffServ



"Call blocking"
approach



"Prioritization"
approach

Comparison of Intserv & Diffserv Architectures

	Intserv	Diffserv
Granularity of service differentiation	Individual Flow	Aggregate of flows
State in routers(e.g. scheduling, buffer management)	Per Flow	Per Aggregate
Traffic Classification Basis	Several header fields	DS Field
Type of service differentiation	Deterministic or statistical guarantees	Absolute or relative assurance
Admission Control	Required	Required for absolute differentiation
Signaling Protocol	Required(RSVP)	Not required for relative schemes

Comparison of Intserv & Diffserv Architectures

	Intserv	Diffserv
Coordination for service differentiation	End-to-End	Local (Per-Hop)
Scope of Service Differentiation	A Unicast or Multicast path	Anywhere in a Network or in specific paths
Scalabilty	Limited by the number of flows	Limited by the number of classes of service
Network Accounting	Based on flow characteristics and QoS requirement	Based on class usage
Network Management	Similar to Circuit Switching networks	Similar to existing IP networks
Interdomain deployment	Multilateral Agreements	Bilateral Agreements