

Fundamentals of Quality of Service

C. Pham

Université de Pau et des Pays de l'Adour

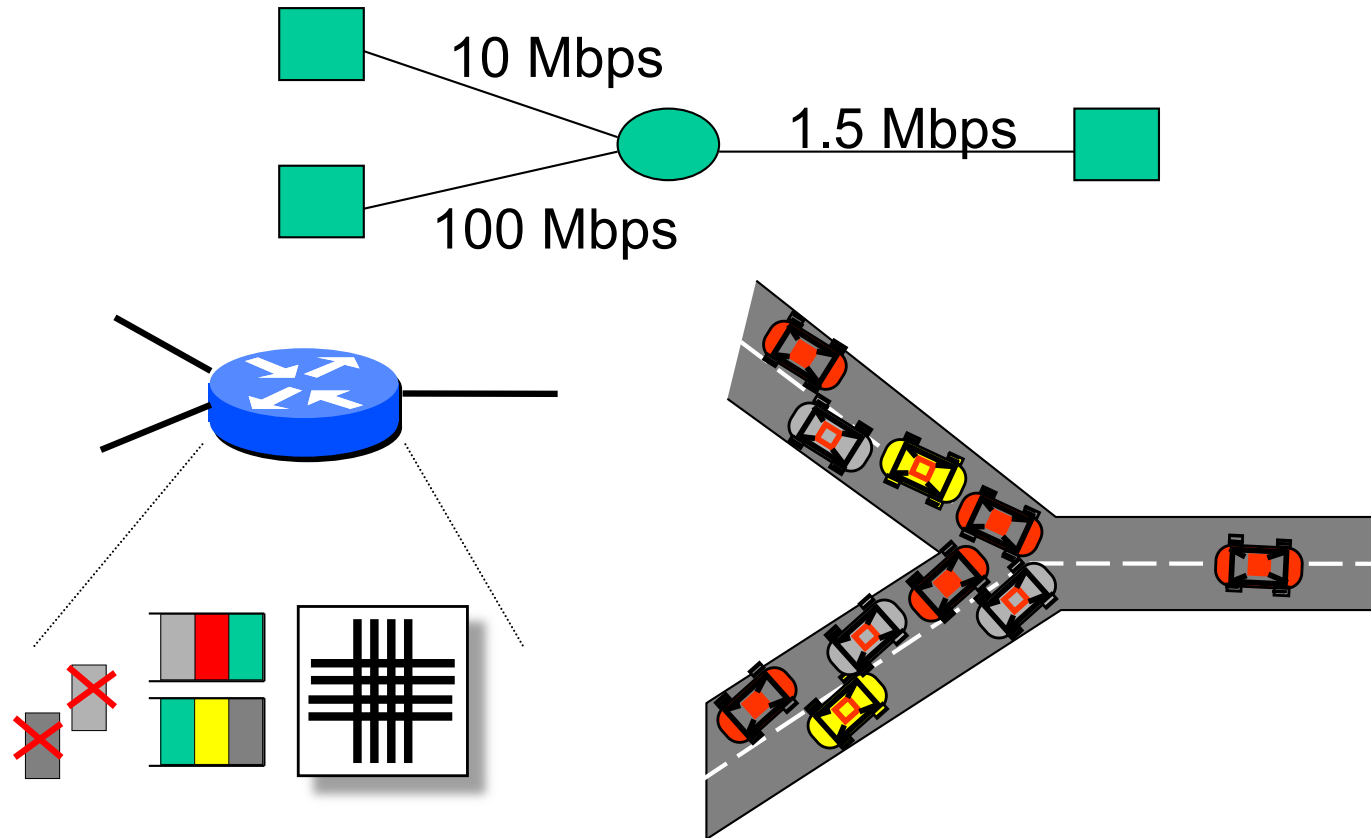
<http://www.univ-pau.fr/~cpham>

Congduc.Pham@univ-pau.fr

These slides borrow material from various sources which are indicated below each slide when necessary

Slides mostly taken from Shivkumar Kalyanaraman which are mostly based on slides of Ion Stoica, Jim Kurose, Srinu Seshan, Srinu Keshav

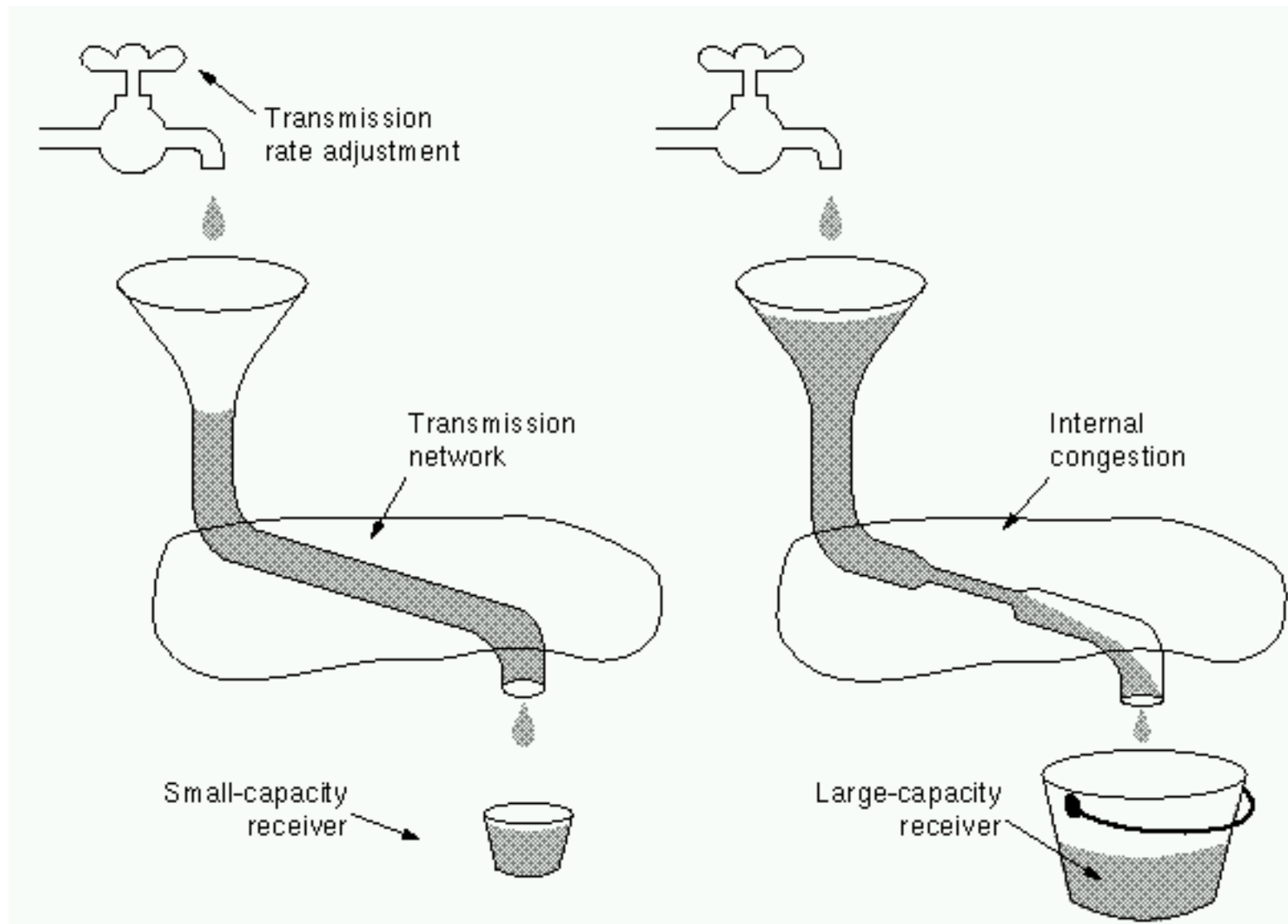
The congestion phenomenon



- ❑ Too many packets sent to the same interface.
- ❑ Difference bandwidth from one network to another

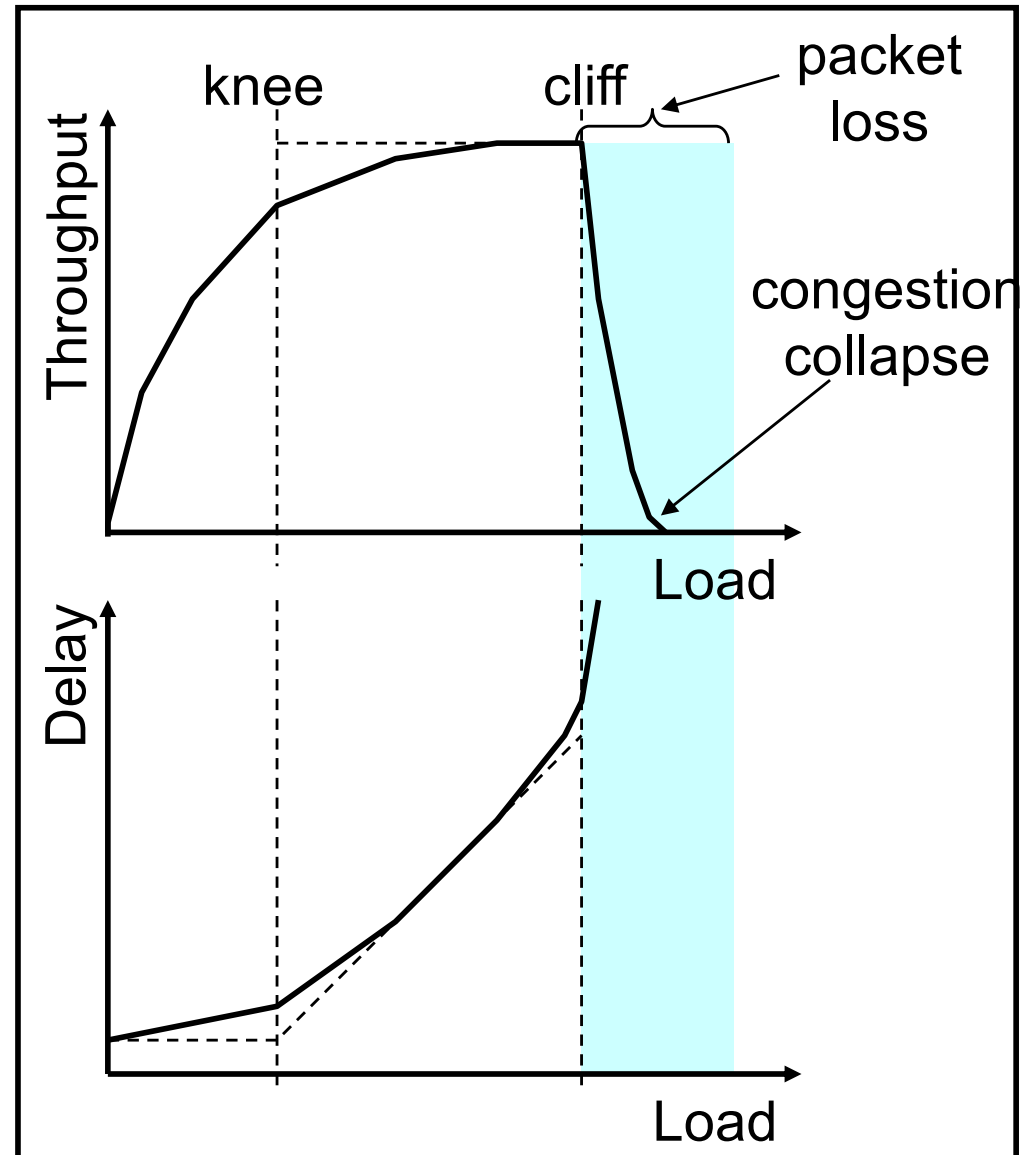
Main consequence: packet losses in routers

The problem of bottlenecks in networks



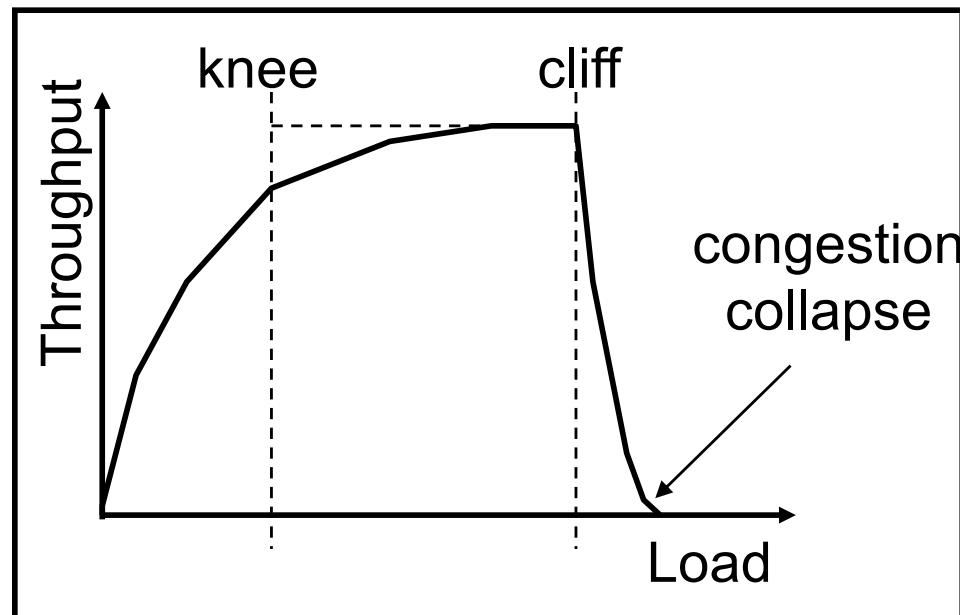
Congestion: A Close-up View

- ❑ knee - point after which
 - ❑ throughput increases very slowly
 - ❑ delay increases fast
- ❑ cliff - point after which
 - ❑ throughput starts to decrease very fast to zero (congestion collapse)
 - ❑ delay approaches infinity
- ❑ Note (in an M/M/1 queue)
 - ❑ $\text{delay} = 1/(1 - \text{utilization})$



Congestion Control vs. Congestion Avoidance

- ❑ Congestion control goal
 - ❑ stay left of cliff
- ❑ Congestion avoidance goal
 - ❑ stay left of knee
- ❑ Right of cliff:
 - ❑ Congestion collapse

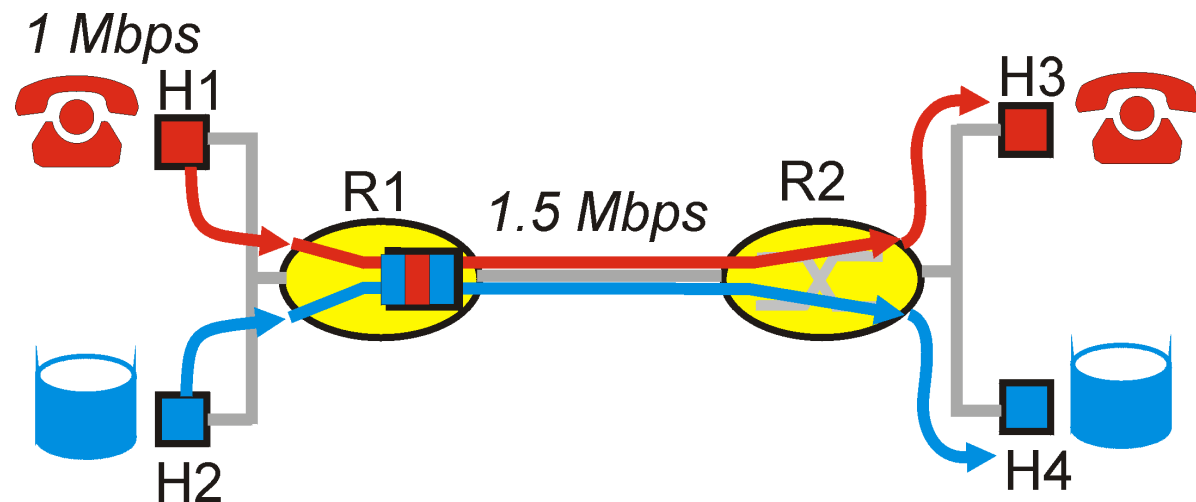


Congestion control principles

- ❑ Reactive
 - ❑ When congestion is detected, inform upstream and downstream nodes,
 - ❑ Then, marks, drops and process packets with priority levels
- ❑ Preventive
 - ❑ Periodical broadcast of node's status (buffer occupancy for instance)
 - ❑ Control of the source, traffic shaping (Leaky Bucket, Token Bucket...),
 - ❑ Flow control, congestion control, admission control.
- ❑ End-to-end
 - ❑ No feedback from the networks
 - ❑ Congestion is detected by end nodes only, using filters (packet losses, RTT variations...)
- ❑ Router-assisted
 - ❑ Congestion indication bit (SNA, DECbit, TCP/ECN, FR, ATM)
 - ❑ More complex router functionalities (XCP)

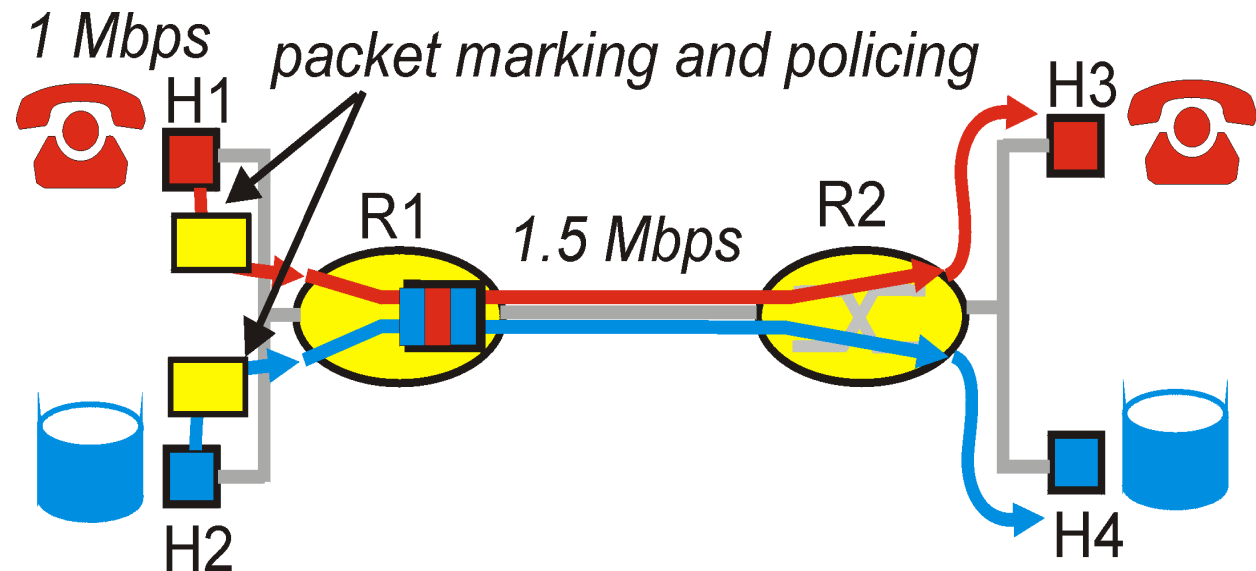
Principles for QOS Guarantees

- 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**



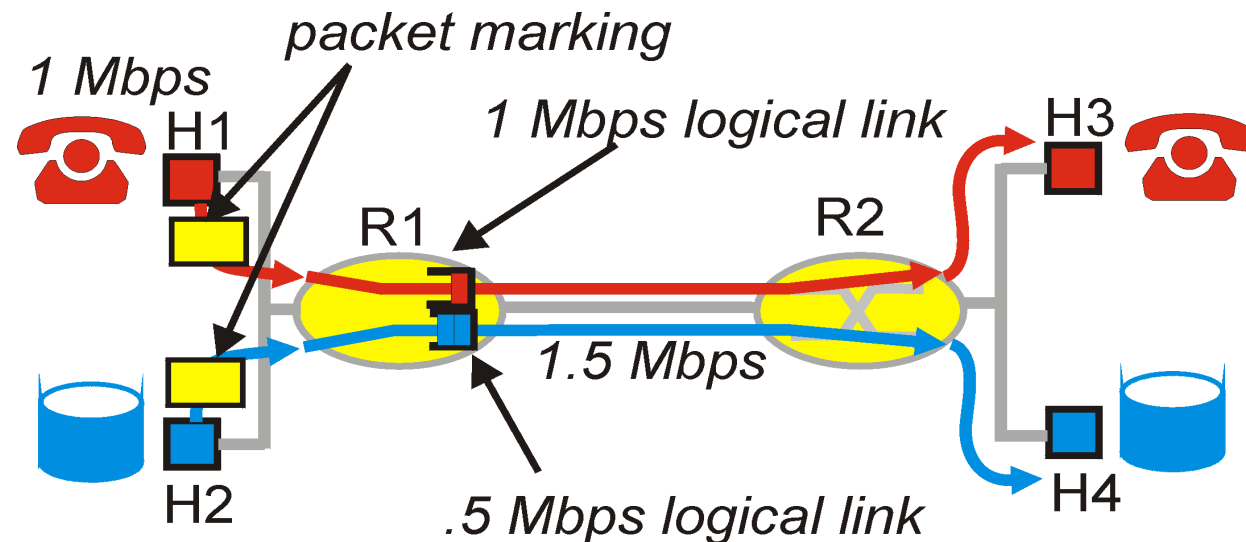
Principles for QOS Guarantees (more)

- ❑ 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:



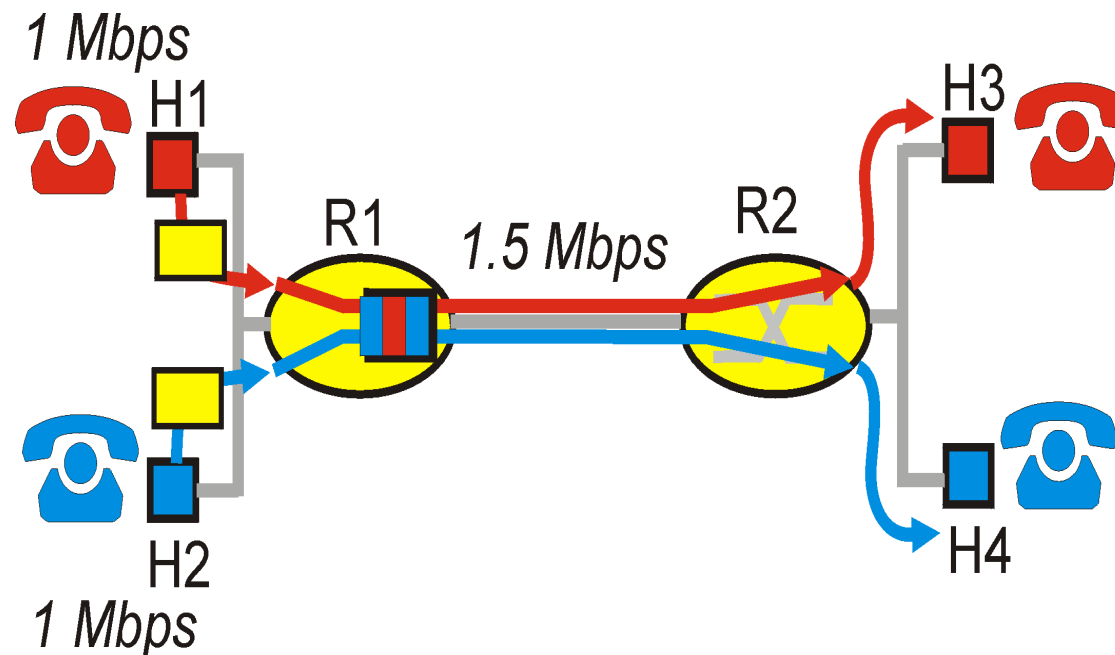
Principles for QOS Guarantees (more)

- ❑ 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

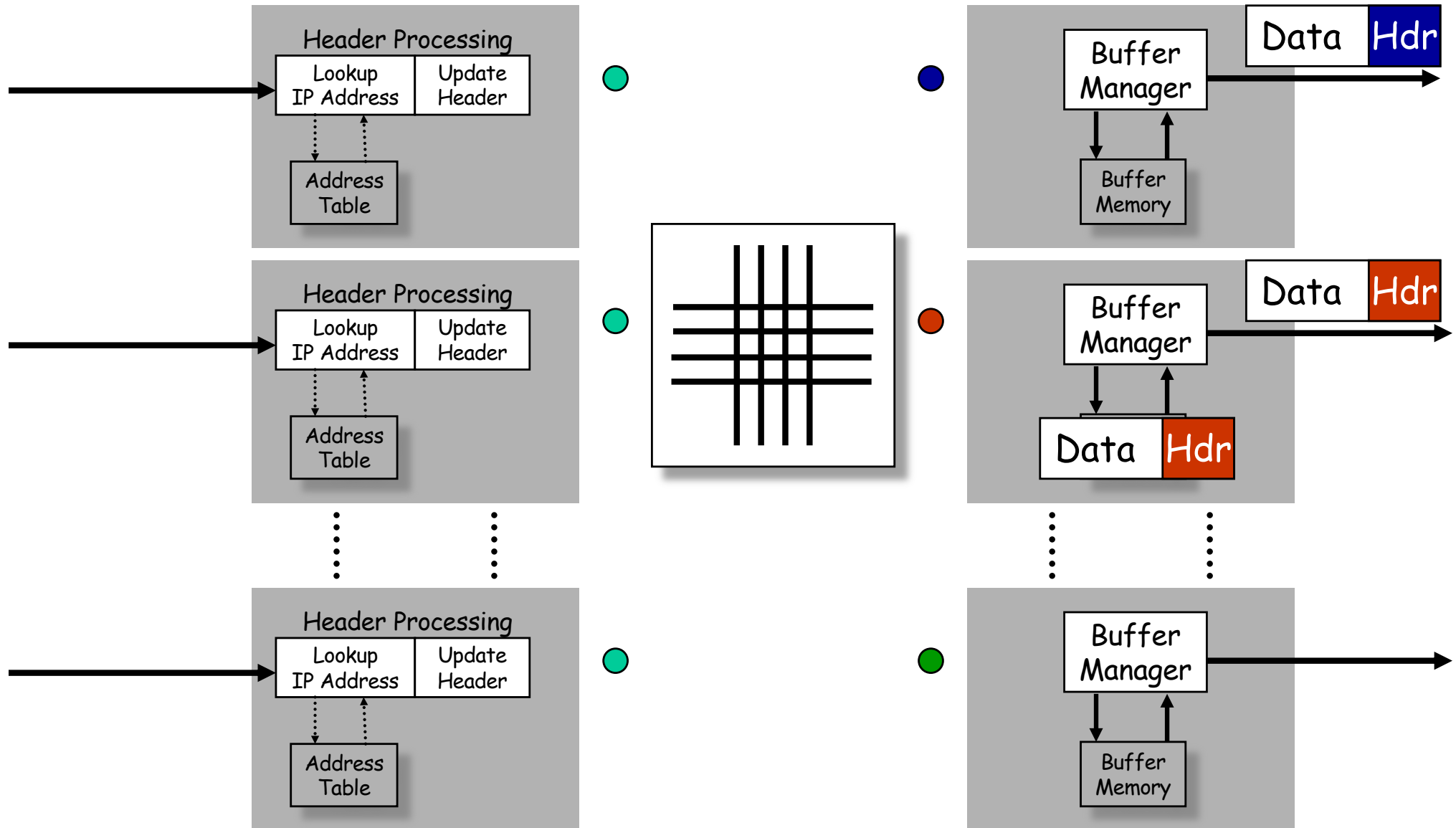


Principles for QOS Guarantees (more)

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



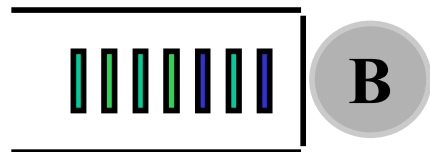
Generic router architecture



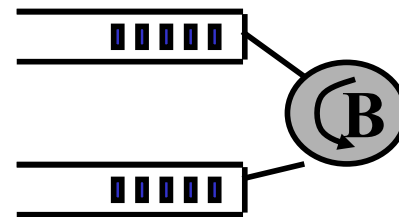
Fundamental Queueing Problems

- ❑ In a FIFO service discipline, the performance assigned to one flow is convoluted with the arrivals of packets from all other flows!
 - ❑ Cant get QoS with a “free-for-all”
 - ❑ Need to use new scheduling disciplines which provide “isolation” of performance from arrival rates of background traffic

FIFO

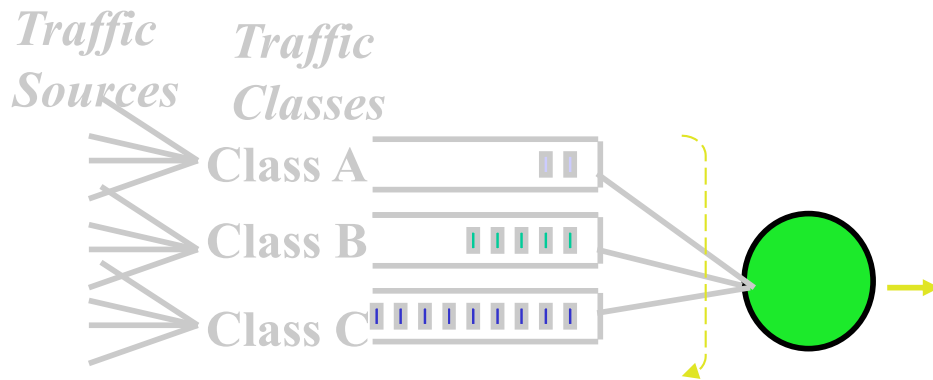


Scheduling Discipline

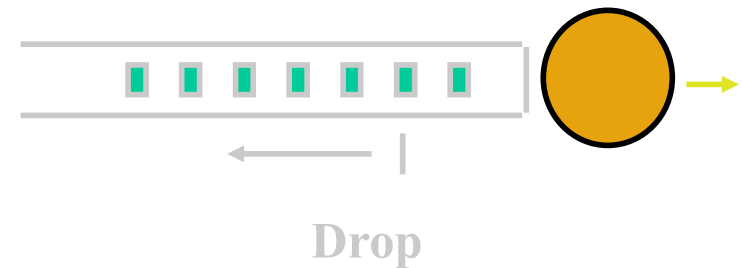


Queuing Disciplines

- ❑ Each router must implement some queuing discipline
- ❑ Queuing allocates bandwidth and buffer space:
 - ❑ Bandwidth: which packet to serve next (scheduling)
 - ❑ Buffer space: which packet to drop next (buff mgmt)
- ❑ Queuing also affects latency



Scheduling



Buffer Management

Typical Internet Queuing

- ❑ FIFO + drop-tail
 - ❑ Simplest choice
 - ❑ Used widely in the Internet
- ❑ FIFO (first-in-first-out)
 - ❑ Implies single class of traffic
- ❑ Drop-tail
 - ❑ Arriving packets get dropped when queue is full regardless of flow or importance
- ❑ Important distinction:
 - ❑ FIFO: scheduling discipline
 - ❑ Drop-tail: drop (buffer management) policy

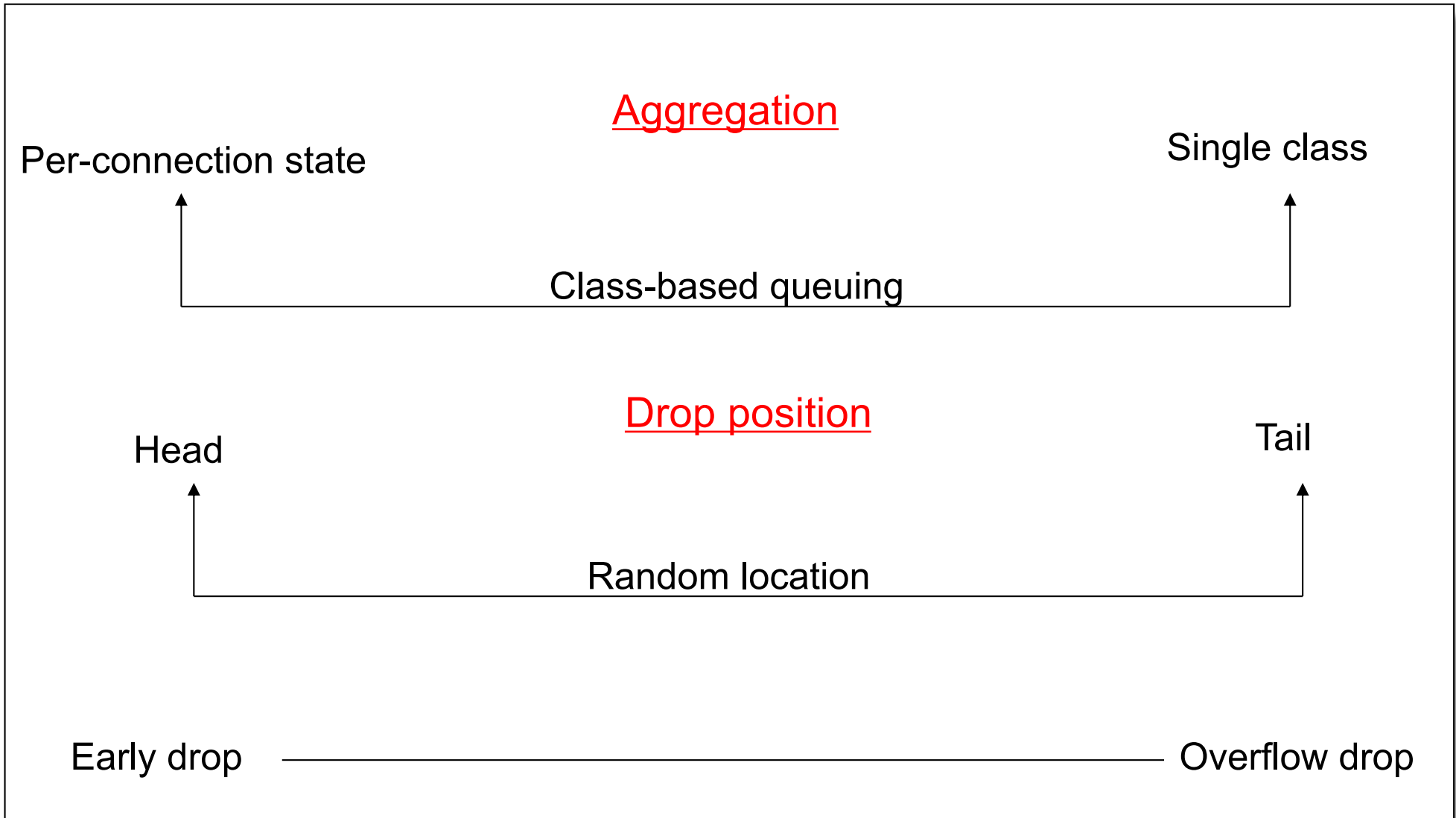
FIFO + Drop-tail Problems

- ❑ FIFO Issues: In a FIFO discipline, the service seen by a flow is *convoluted* with the arrivals of packets from all other flows!
 - ❑ No isolation between flows: full burden on e2e control
 - ❑ No policing: send more packets → get more service
- ❑ Drop-tail issues:
 - ❑ Routers are forced to have large queues to maintain high utilizations
 - ❑ Larger buffers => larger steady state queues/delays
 - ❑ Synchronization: end hosts react to same events because packets tend to be lost in bursts
 - ❑ Lock-out: a side effect of burstiness and synchronization is that a few flows can monopolize queue space

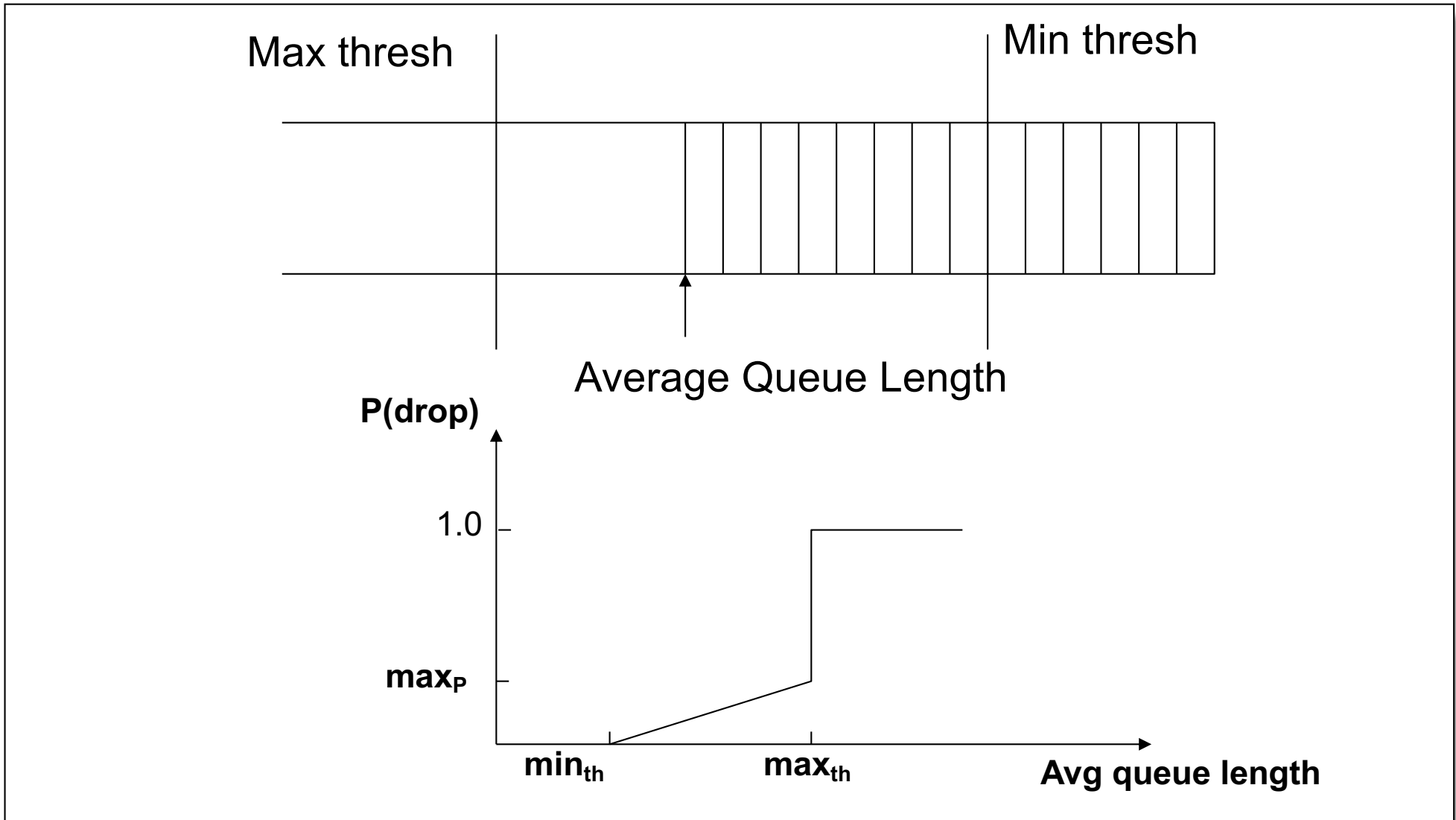
Queue Management Ideas

- ❑ Synchronization, lock-out:
 - ❑ Random drop: drop a randomly chosen packet
 - ❑ Drop front: drop packet from head of queue
- ❑ High steady-state queuing vs burstiness:
 - ❑ Early drop: Drop packets before queue full
 - ❑ Do not drop packets “too early” because queue may reflect only burstiness and not true overload
- ❑ Misbehaving vs Fragile flows:
 - ❑ Drop packets proportional to queue occupancy of flow
 - ❑ Try to protect fragile flows from packet loss (eg: color them or classify them on the fly)
- ❑ Drop packets vs Mark packets:
 - ❑ Dropping packets interacts w/ reliability mechanisms
 - ❑ Mark packets: need to trust end-systems to respond!

Packet Drop Dimensions



Random Early Detection (RED)



Random Early Detection (RED)

- ❑ Maintain running average of queue length
 - ❑ Low pass filtering
- ❑ If $\text{avg } Q < \text{min}_{th}$ do nothing
 - ❑ Low queuing, send packets through
- ❑ If $\text{avg } Q > \text{max}_{th}$, drop packet
 - ❑ Protection from misbehaving sources
- ❑ Else **mark (or drop)** packet in a manner **proportional to queue length** & **bias** to protect against **synchronization**
 - ❑ $P_b = \max_p(\text{avg} - \text{min}_{th}) / (\text{max}_{th} - \text{min}_{th})$
 - ❑ Further, bias P_b by history of unmarked packets
 - ❑ $P_a = P_b / (1 - \text{count} * P_b)$

RED Issues

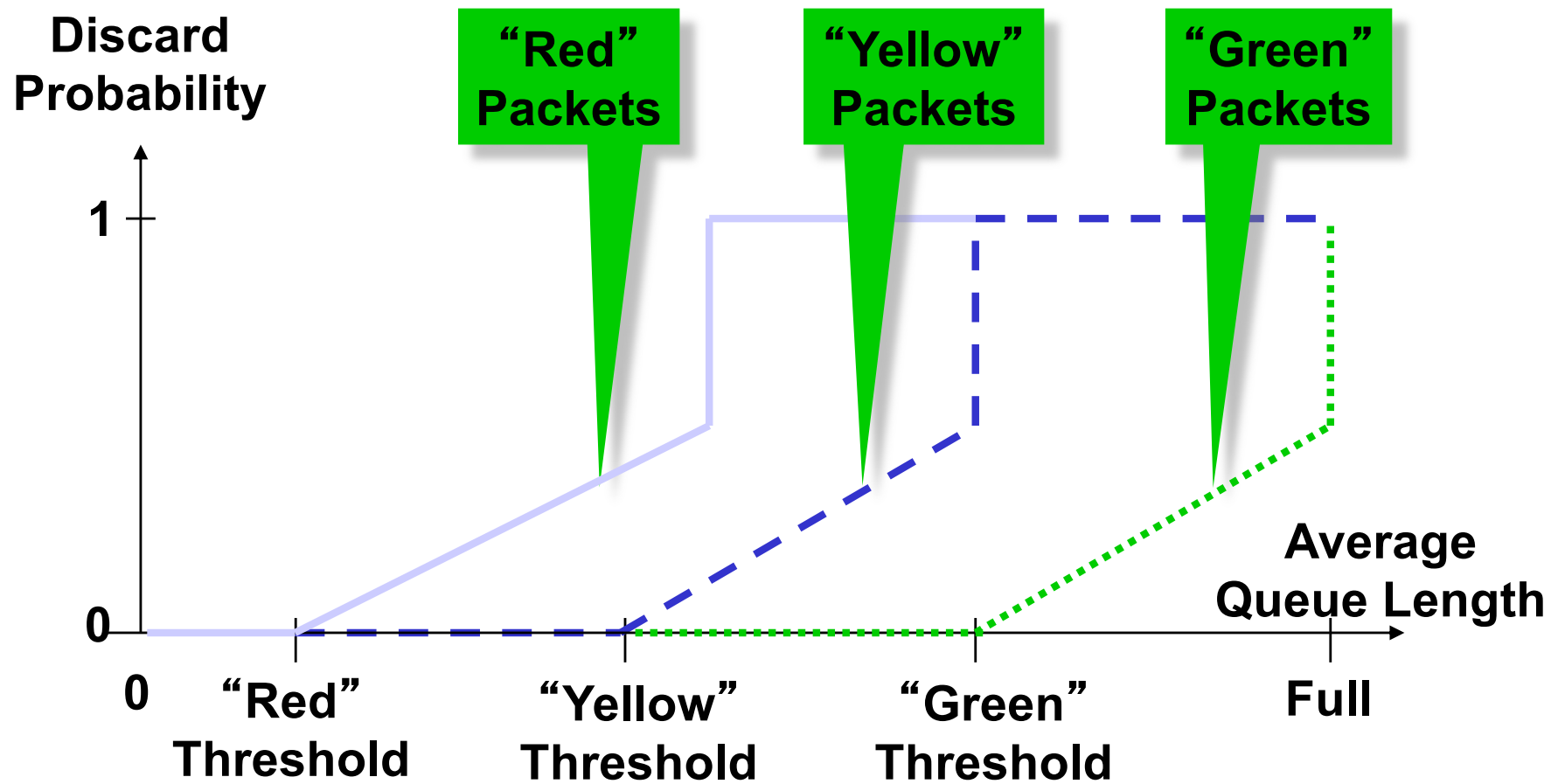
❑ Issues:

- ❑ Breaks synchronization well
- ❑ Extremely sensitive to parameter settings
- ❑ Wild queue oscillations upon load changes
- ❑ Fail to prevent buffer overflow as #sources increases
- ❑ Does not help fragile flows (eg: small window flows or retransmitted packets)
- ❑ Does not adequately isolate cooperative flows from non-cooperative flows

❑ Isolation:

- ❑ Fair queuing achieves isolation using per-flow state
- ❑ RED penalty box: Monitor history for packet drops, identify flows that use disproportionate bandwidth

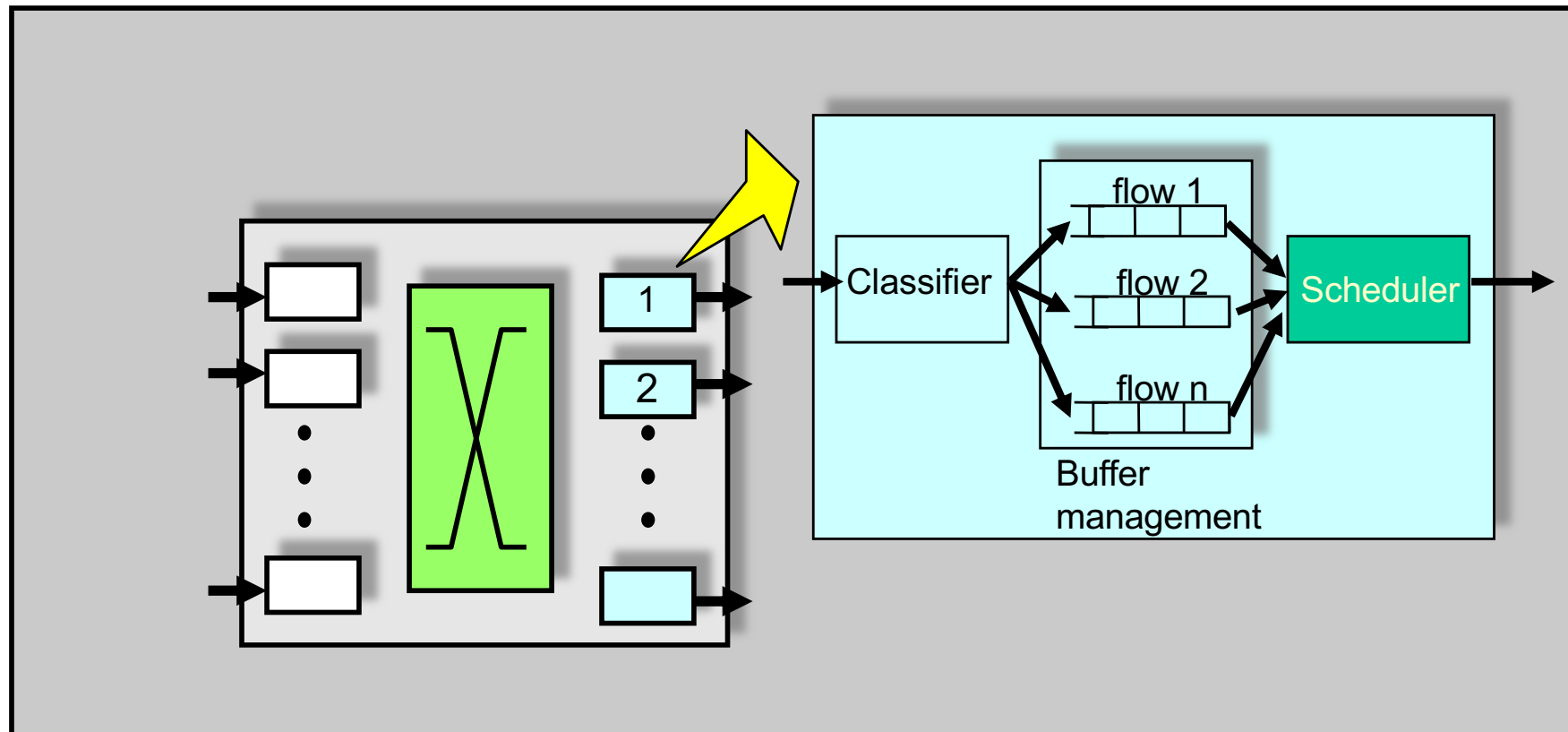
RED with Multiple Thresholds



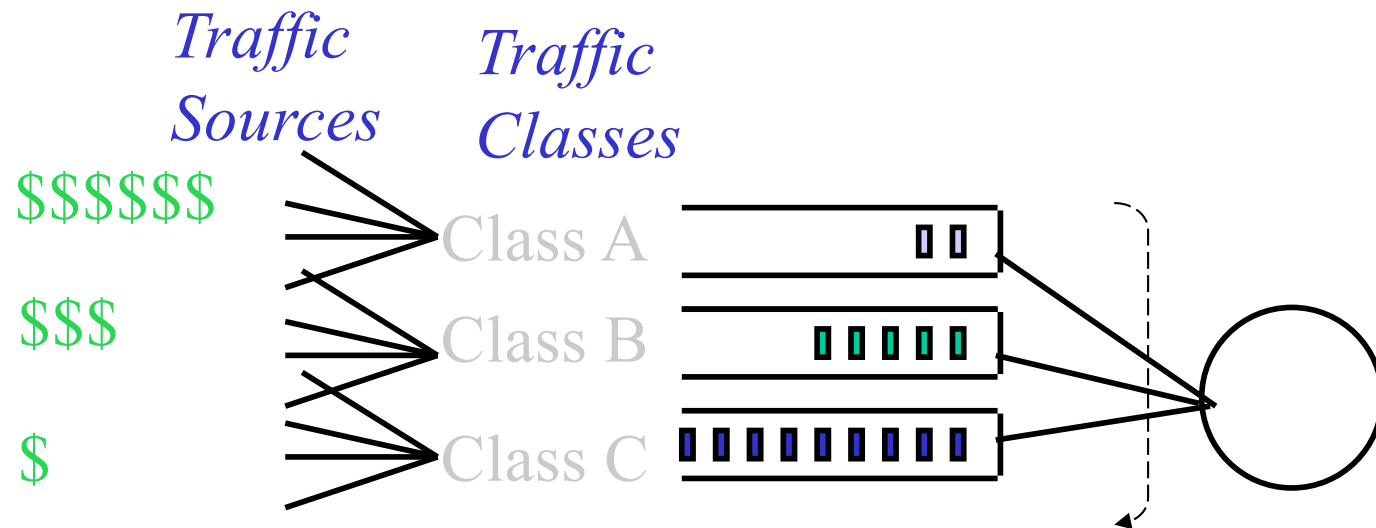
SCHEDULING

Packet Scheduling

- ❑ Decide when and what packet to send on output link
 - ❑ Usually implemented at output interface



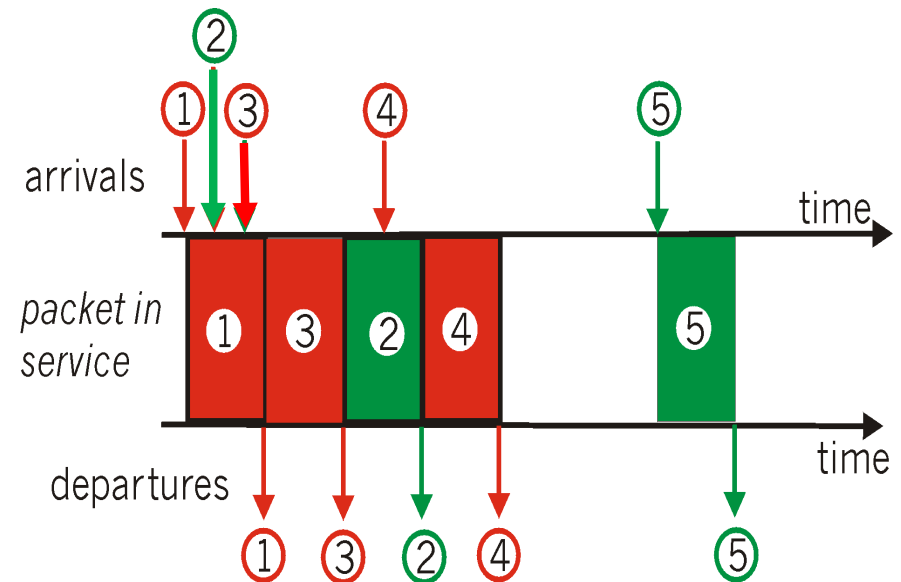
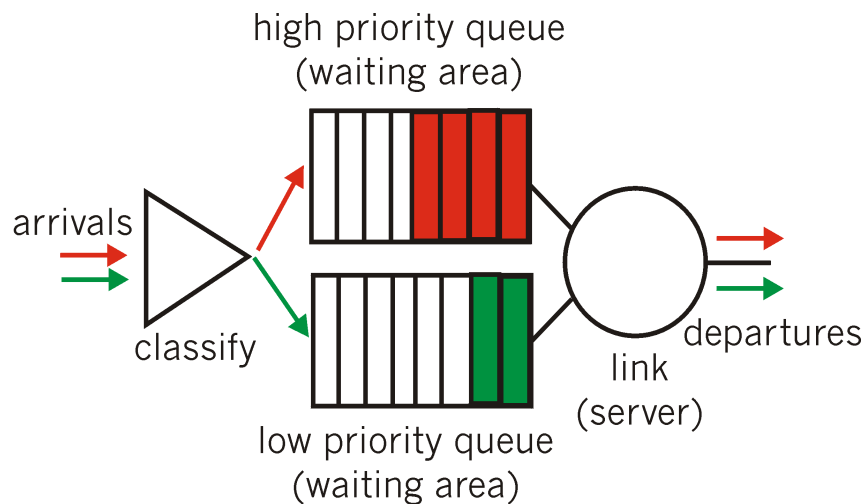
Mechanisms: Queuing/Scheduling



- ❑ Use a few bits in header to indicate which queue (class) a packet goes into (also branded as CoS)
- ❑ High \$\$ users classified into high priority queues, which also may be less populated
 - ❑ => lower delay and low likelihood of packet drop
- ❑ Ideas: priority, round-robin, classification, aggregation, ...

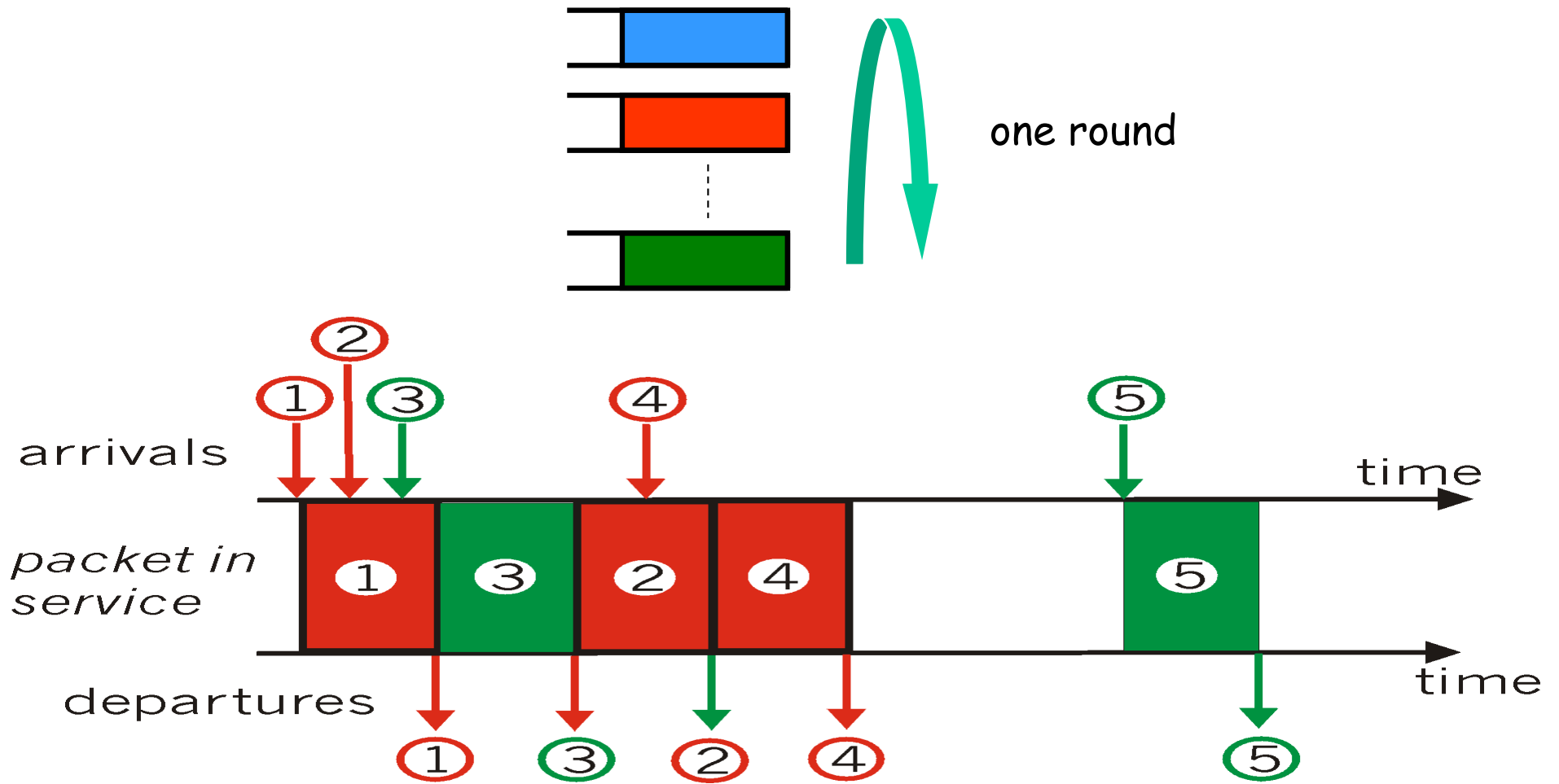
Priority Queueing

- Priority Queueing: classes have different priorities; class may depend on explicit marking or other header info, eg IP source or destination, TCP Port numbers, etc.
- Transmit a packet from the highest priority class with a non-empty queue
- Preemptive and non-preemptive versions



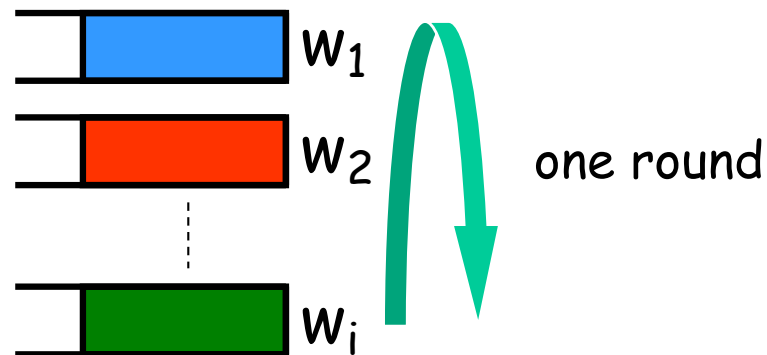
Round Robin (RR)

- Round Robin: scan class queues serving one from each class that has a non-empty queue



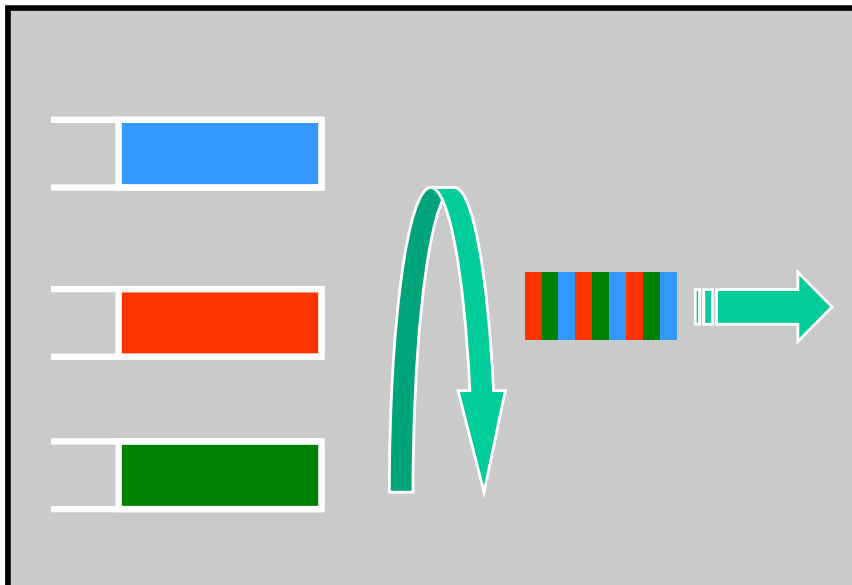
Weighted Round Robin (WRR)

- Assign a weight to each connection and serve a connection in proportion to its weight
- Ex:
 - Connection A, B and C with same packet size and weight 0.5, 0.75 and 1. How many packets from each connection should a round-robin server serve in each round?
 - Answer: Normalize each weight so that they are all integers: we get 2, 3 and 4. Then in each round of service, the server serves 2 packets from A, 3 from B and 4 from C.



Generalized Processor Sharing (GPS)

- Assume a fluid model of traffic
 - Visit each non-empty queue in turn (like RR)
 - Serve infinitesimal from each
 - Leads to “max-min” fairness
- GPS is un-implementable!
 - We cannot serve infinitesimals, only packets



max-min fairness

Consider n sources $1, \dots, n$ requesting resources x_1, \dots, x_n with $x_1 < x_2 < \dots < x_n$ for instance. Link or server capacity is C .

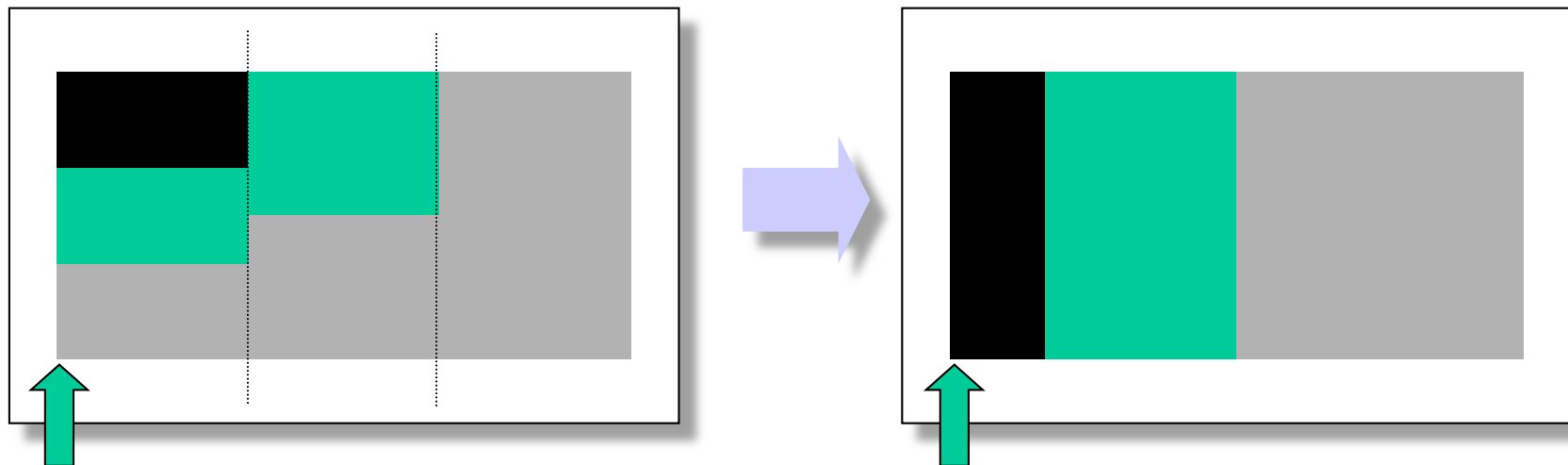
We assign C/n to source 1. If $C/n > x_1$, we give $C/n + (C/n - x_1)/(n-1)$ to the remaining $(n-1)$ sources. If this amount is greater than x_2 , we iterate.

Packet Approximation of Fluid System

- ❑ GPS un-implementable
- ❑ Standard techniques of approximating fluid GPS
 - ❑ Select packet that finishes first in GPS assuming that there are no future arrivals (emulate GPS on the side)
- ❑ Important properties of GPS
 - ❑ Finishing order of packets currently in system independent of future arrivals
- ❑ Implementation based on virtual time
 - ❑ Assign virtual finish time to each packet upon arrival
 - ❑ Packets served in increasing order of virtual times

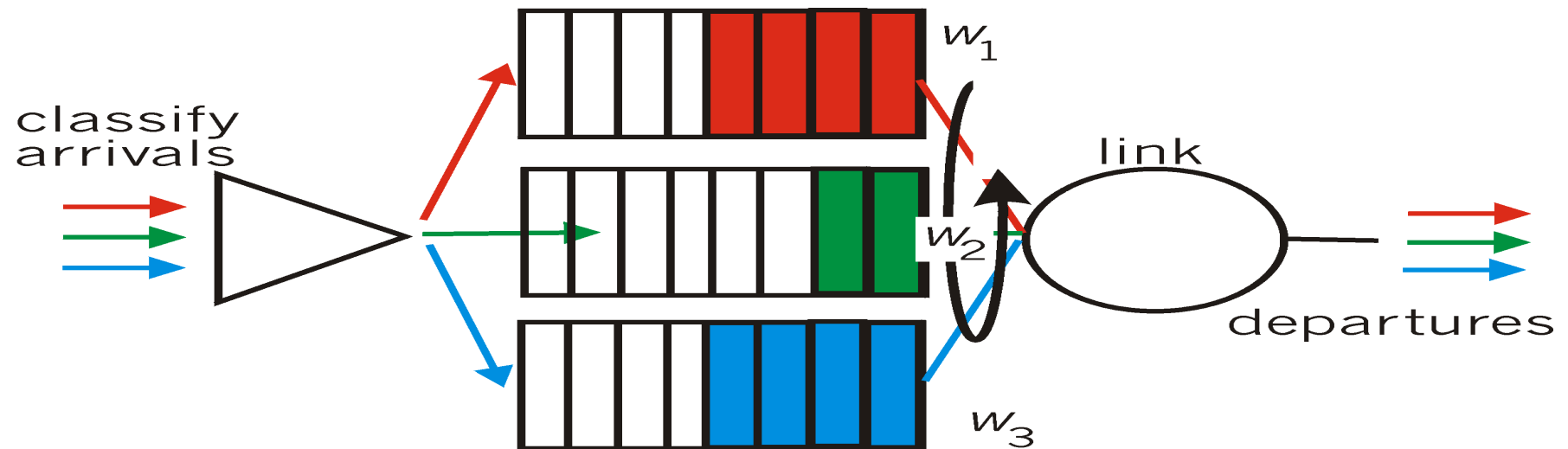
Fair Queuing (FQ)

- ❑ Idea: serve packets in the order in which they would have finished transmission in the fluid flow system
- ❑ Mapping bit-by-bit schedule onto packet transmission schedule
- ❑ Transmit packet with the lowest finish time at any given time



Weighted Fair Queueing

- Variation of FQ: Weighted Fair Queueing (WFQ)
- Weighted Fair Queueing: is a generalized Round Robin in which an attempt is made to provide a class with a differentiated amount of service over a given period of time



Implementing WFQ

- ❑ WFQ needs per-connection (or per-aggregate) scheduler state → implementation complexity.
 - ❑ complex iterated deletion algorithm
 - ❑ complex sorting at the output queue on the service tag
- ❑ WFQ needs to know the weight assigned for each queue → manual configuration, signalling.
- ❑ WFQ is not perfect...
- ❑ Router manufacturers have implemented as early as 1996 WFQ in their products
 - ❑ from CISCO 1600 series
 - ❑ Fore System ATM switches

QOS SPECIFICATION,
TRAFFIC, SERVICE
CHARACTERIZATION,
BASIC MECHANISMS

Service Specification

- ❑ **Loss:** probability that a flow's packet is lost
- ❑ **Delay:** time it takes a packet's flow to get from source to destination
- ❑ **Delay jitter:** maximum difference between the delays experienced by two packets of the flow
- ❑ **Bandwidth:** maximum rate at which the source can send traffic
- ❑ **QoS spectrum:**

Best Effort

Leased Line

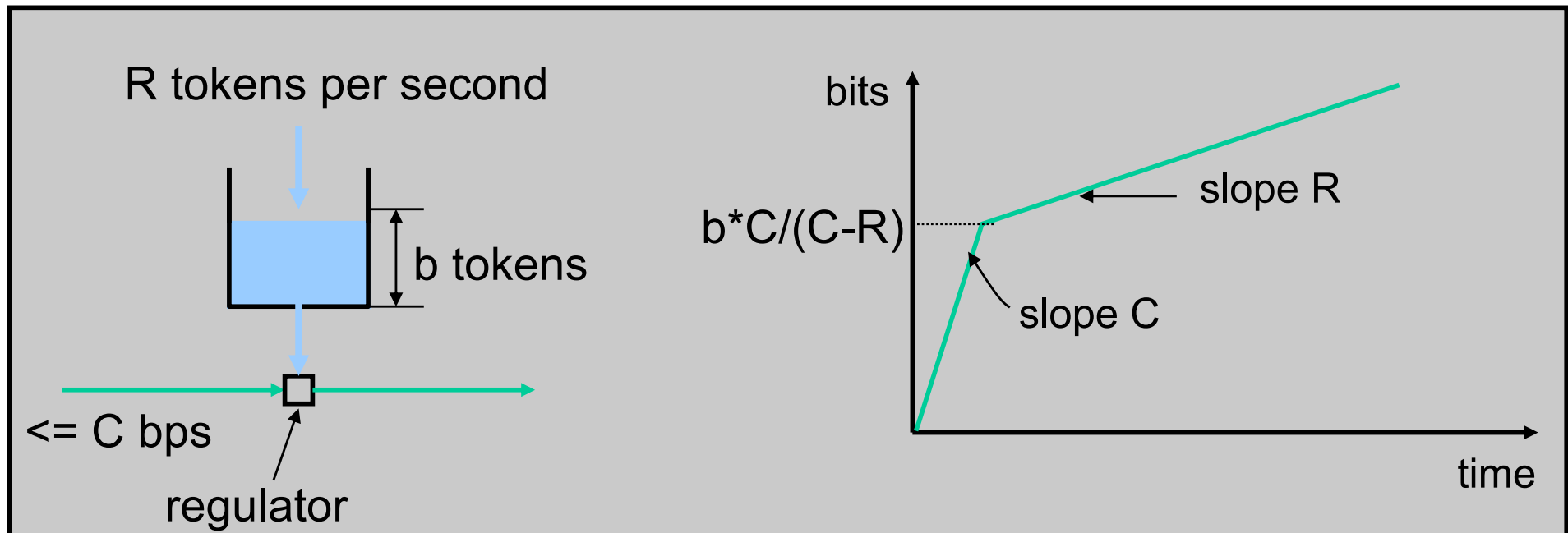


Traffic and Service Characterization

- ❑ To quantify a service one has to know
 - ❑ Flow's traffic arrival
 - ❑ Service provided by the router, i.e., resources reserved at each router
- ❑ Examples:
 - ❑ Traffic characterization: token bucket
 - ❑ Service provided by router: fix rate and fix buffer space
 - ❑ **Characterized by a service model** (service curve framework)

Ex: Token Bucket

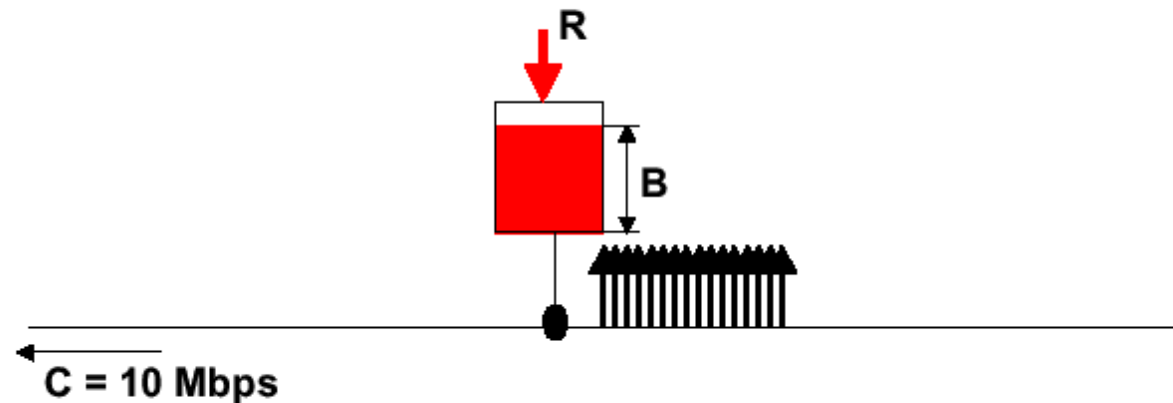
- ❑ Characterized by three parameters (b , R , C)
 - ❑ b - token depth
 - ❑ R - average arrival rate
 - ❑ C - maximum arrival rate (e.g., link capacity)
- ❑ A bit is transmitted only when there is an available token
 - ❑ When a bit is transmitted exactly one token is consumed



Token Bucket

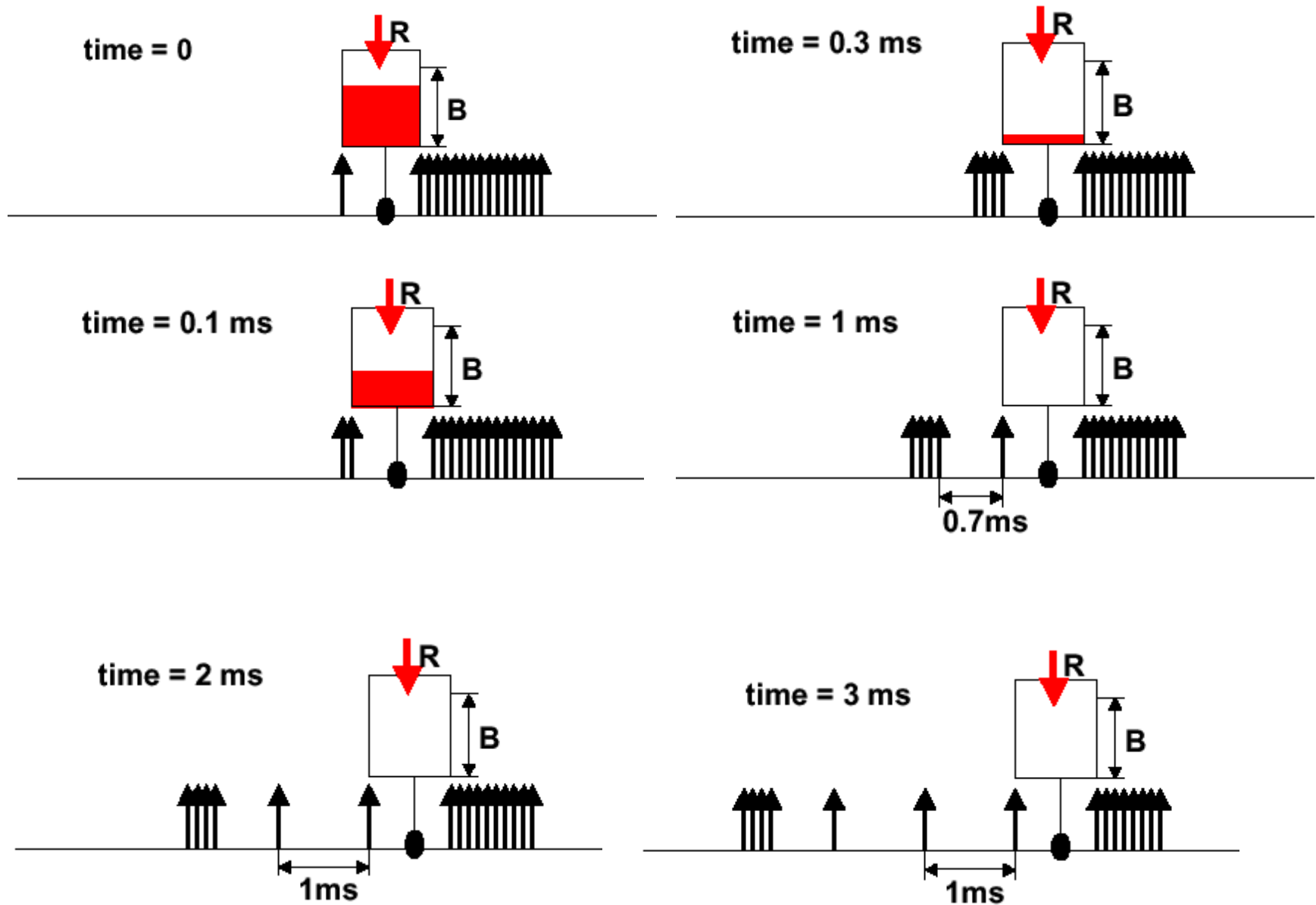
Example

- $B = 4000$ bits, $R = 1$ Mbps, $C = 10$ Mbps
- Packet length = 1000 bits
- Assume the bucket is initially full and a “large” burst of packets arrives



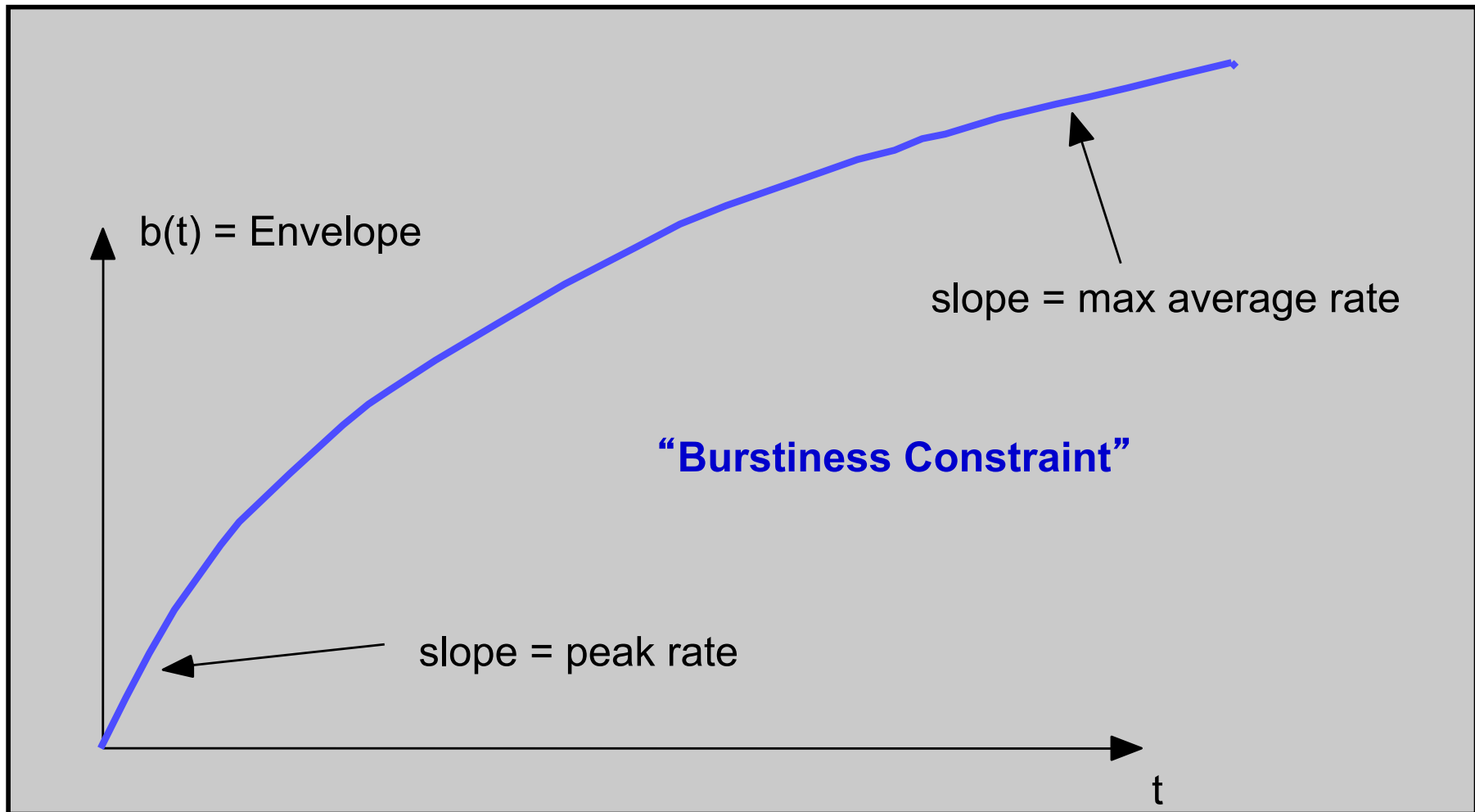
istoica@cs.cmu.edu

Token Bucket



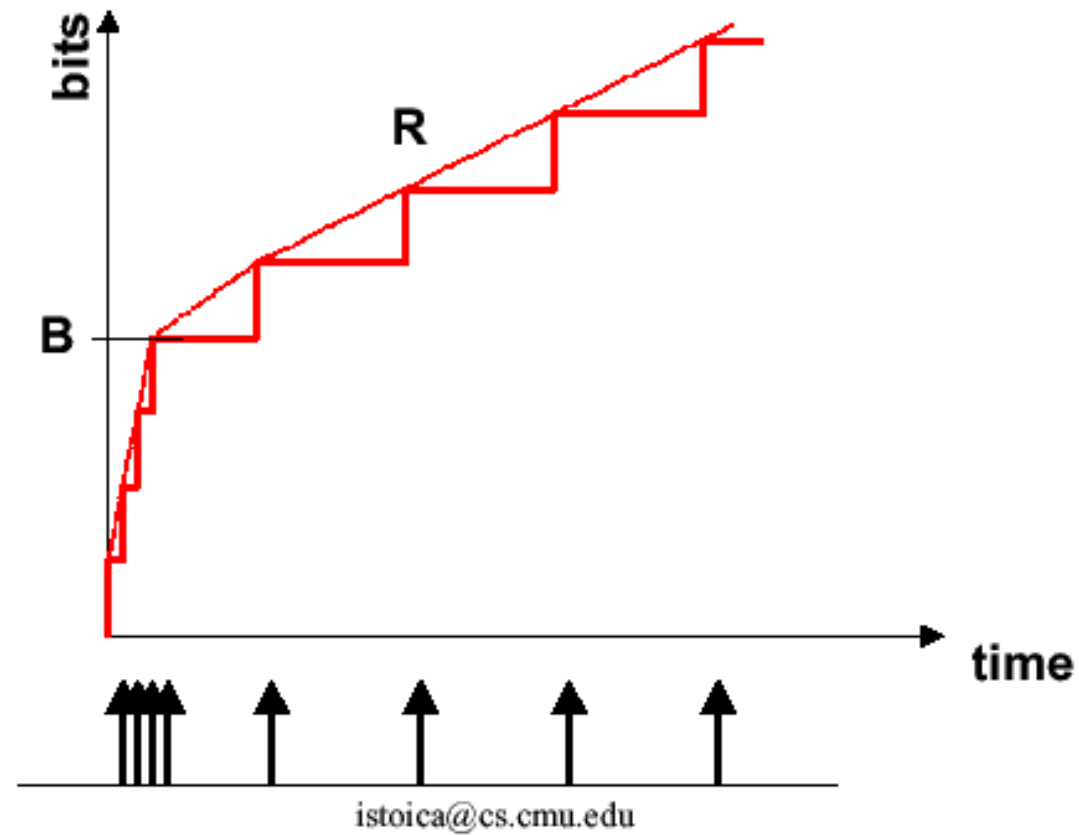
Traffic Envelope (Arrival Curve)

- Maximum amount of service that a flow can send during an interval of time t



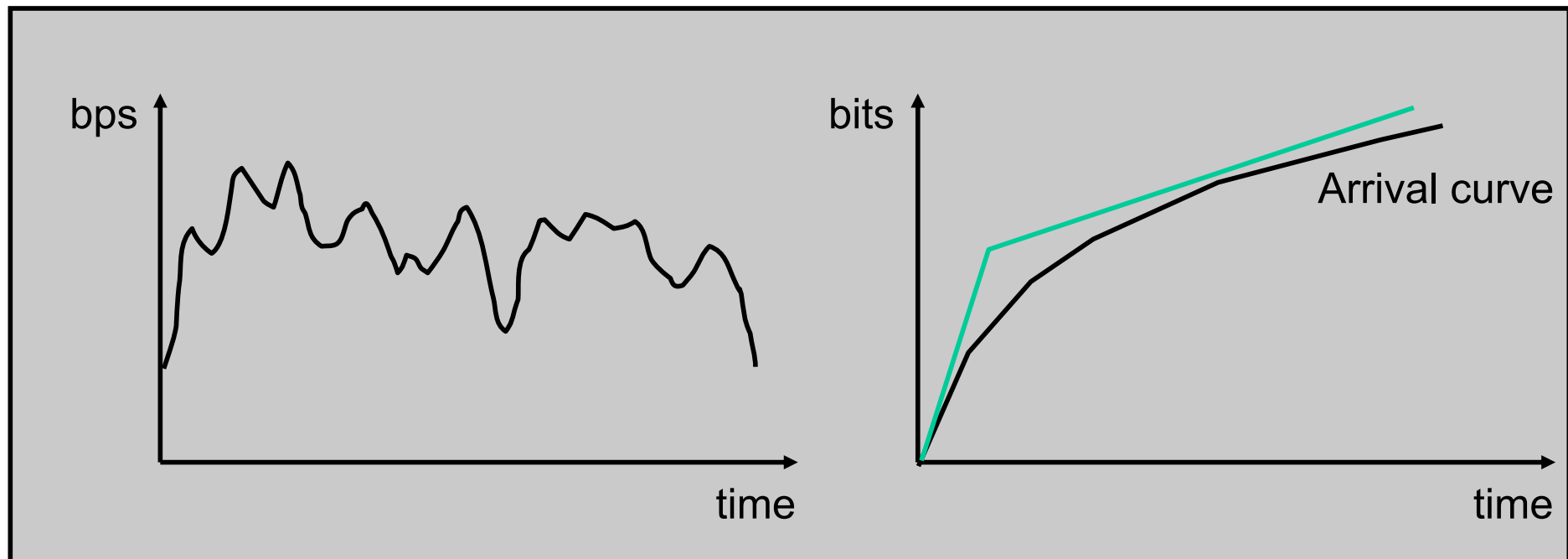
Arrival curve

$A(t)$ – number of bits received up to time t



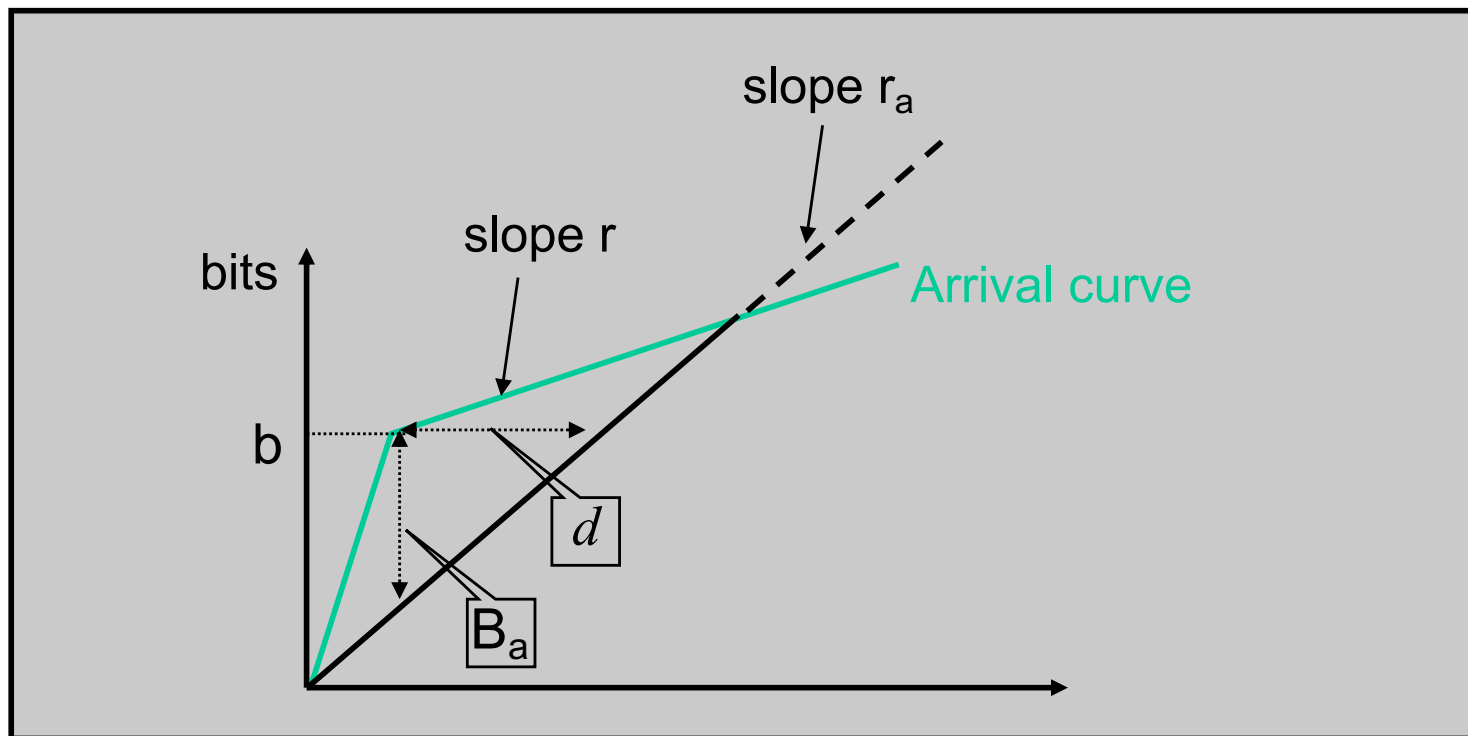
Characterizing a Source by Token Bucket

- ❑ Arrival curve - maximum amount of bits transmitted by time t
- ❑ Use token bucket to bound the arrival curve

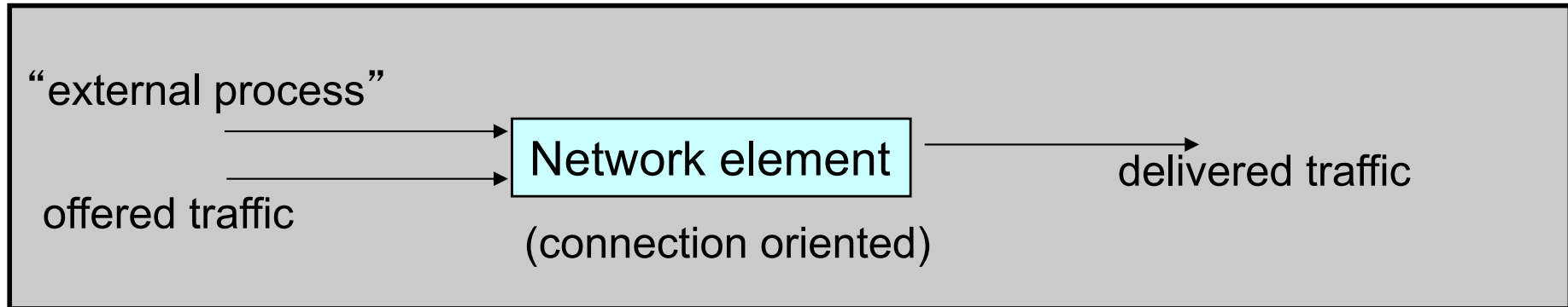


Per-hop Reservation with Token Bucket

- Given b, r, R and per-hop delay d
- Allocate bandwidth r_a and buffer space B_a such that to guarantee d

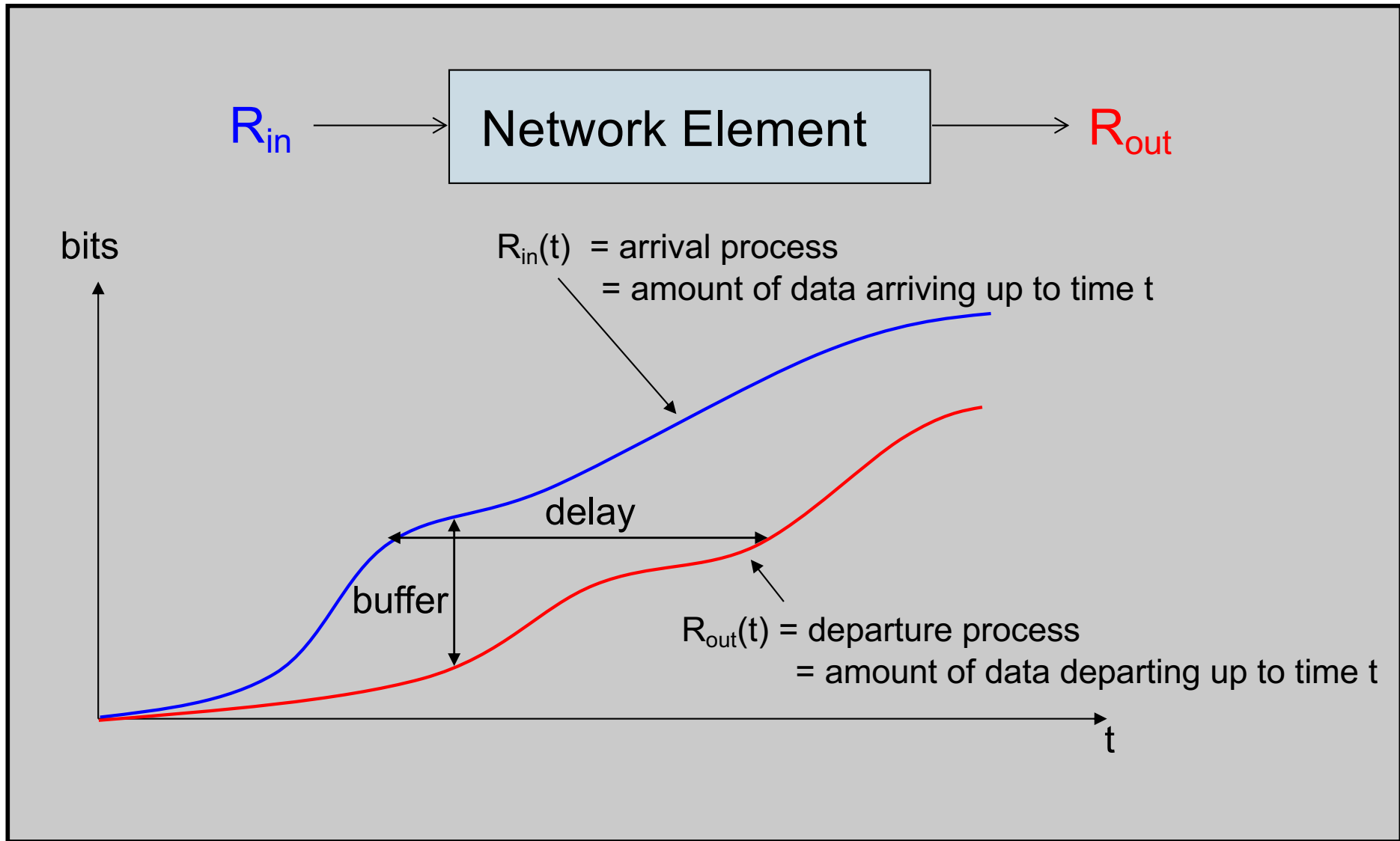


What is a Service Model?



- ❑ The QoS measures (delay, throughput, loss, cost) depend on offered traffic, and possibly other external processes.
- ❑ A service model attempts to characterize the relationship between offered traffic, delivered traffic, and possibly other external processes.

Arrival and Departure Process



Delay and Buffer Bounds

