University of BRISTOL

Advanced Networks

(EENGM4211)

Rasheed Hussain

rasheed.hussain@bristol.ac.uk

Best contact : Microsoft Team

bristol.ac.uk

Slides courtesy: Dr D. Kaleshi & Dr G. Oikonomou as well as Internet

University of BRISTOL

Part 6:
Quality of Service

University of BRISTOL

Unit Outline

- Introduction
- Internet Routing and Switching
- IP Multicast
- Networking for Realtime Applications
- Routing in Wireless Networks
- Quality of Service

University of BRISTOL

Outline & Topics

- Motivation
- Some practical principles
- How do we apply the principles to make QoS work?
- Case study : IP Quality of Service
  - What? why?
  - Protocols

University of BRISTOL

Motivation

QoS Definition

From the end **user** point of view

QoS is a quantification of service/application-relevant measures of network effectiveness against acceptable levels of measures such as:

- Delay
- Jitter
- Packet Loss Rate
- Response time
- Throughput
- Service Availability

QoS from the service providers point of view:

Provide services to specific traffic classes such that QoS can be provided to end-users on a guaranteed/differential basis

→ Ability to provide better service to selected traffic

Different categories of traffic

- Low latency (e.g. streaming)
- Guaranteed delivery (e.g. e-commerce)
- Best effort (e.g. email)

Table 1 Internet Traffic Characteristics

IntServ Category [1]	*IntServ Subcategory*	*3GPP Category [2]*	*Examples*	*Constant Rate Necessary*	*Low Delay Necessary*	*Low Jitter*	*Low Delay Preferred*
Tolerant Real-Time		Streaming	Audio/video streaming	√	√		
Intolerant Real-Time		Conversational	IP telephony, teleconferencing	√	√	√	
Elastic (Non-Real-Time)	Interactive Burst	Interactive	Telnet, HTTP				√
	Interactive Bulk Transfer		FTP				√
	Asynchronous Bulk Transfer	Background	SMTP				

University of BRISTOL

Motivation

Why should we care for QoS?

- Because…
  - Customers have expectations,… and **receive bills**
  - Operators must fulfil those expectations to justify the bills
  - It is an interesting theoretical and practical problem
- Historically there was one service – voice, wired up per call
  - Easy to manage QoS – certain number of calls per hour, lasting so many minutes gives number of circuits needed
  - Other things easier with digital (codecs, tones, not pulses, etc.)
- Then there was fax – pretty much the same as voice
- Then there was packet switching – the industry tried to pretend this was just like voice too, but it isn't
- Then the Internet got in the way…

University of BRISTOL

Motivation

Benefits of using QoS

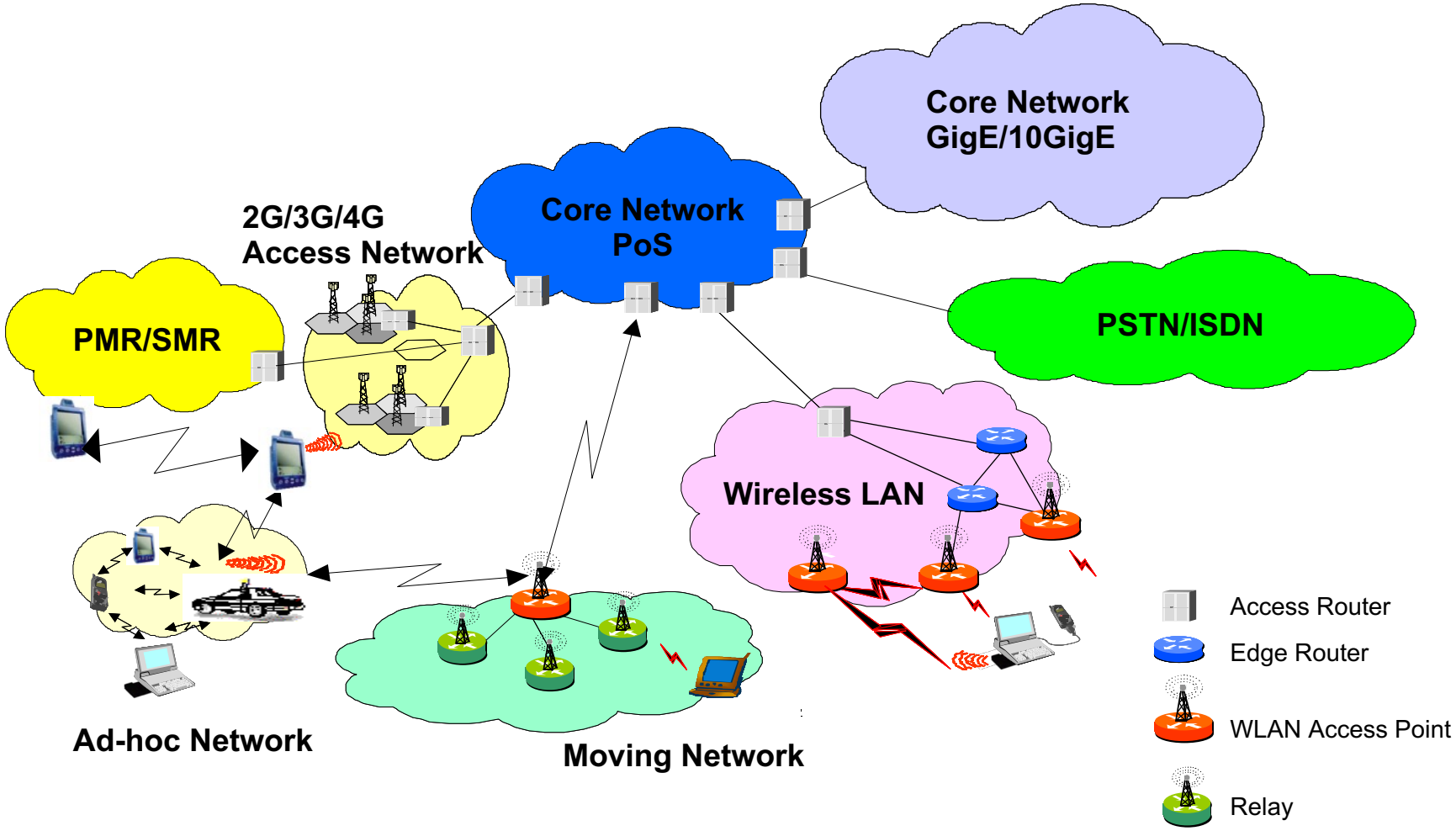
- Traffic gets to destinations quickly and reliably to meet application requirements
- Controlled network latency and jitter and improved loss characteristics
- Effectively manage existing bandwidth and allocate resources in the network (bandwidth, buffer, priority etc)
- Control and predictability beyond **“best effort”** concept

University of BRISTOL

Section 1

Defining the problem
Managing flows and resources

Let's look at the Internet



University of BRISTOL

QoS: the problem

- QoS is only a problem when the communications system is **oversubscribed**
- Guaranteed QoS:
  - If you know the whole state of the system over time
    - This isn't possible
  - Events and flows are predictable
    - No errors, no congestion
    - But… they aren't and we can't
  - Applications and users behave

- Question: How can we improve the assurance that an application will receive a requested throughput, delay, error rate?

QoS: some questions

Why can't it be done in the best effort Internet?

Because:

- Routes change all the time.
- Traffic cannot be predicted or planned.
- Routers/switches and links are congestion points that cannot be controlled or managed.

How can this be fixed?

- end-to-end transport protocol guarantee delivery but cannot control delay or throughput
- Can asynchronous packet-switching technology provide guarantees of service? How?
  - e.g. OSPF takes into account the link load.

  Why is this not sufficient?
    - Response time; risk of instability in the network routing, etc.

University of BRISTOL

Internet Principles

- Global management framework for naming and addressing
- Common, limited, set of internetworking functions
- Local optimisations within a global framework
  - Backbone vs. border vs. local area – to make routing work
  - Communications layers – to separate concerns
- End-end principle
  - Functions placed at low levels of a system may be redundant when compared with the cost of providing them
  - Such functions cannot be completely and correctly implemented except with the knowledge and help of the application executing at the end points of the communication system.
- Fate-sharing principle
  - An end-to-end protocol design should not rely on the **maintenance of state** (i.e. information about the state of the end-to-end communication) inside the network.
  - Such state should be **maintained only in the end-systems**, in such a way that the state can only be destroyed when the end-system itself breaks.

University of BRISTOL

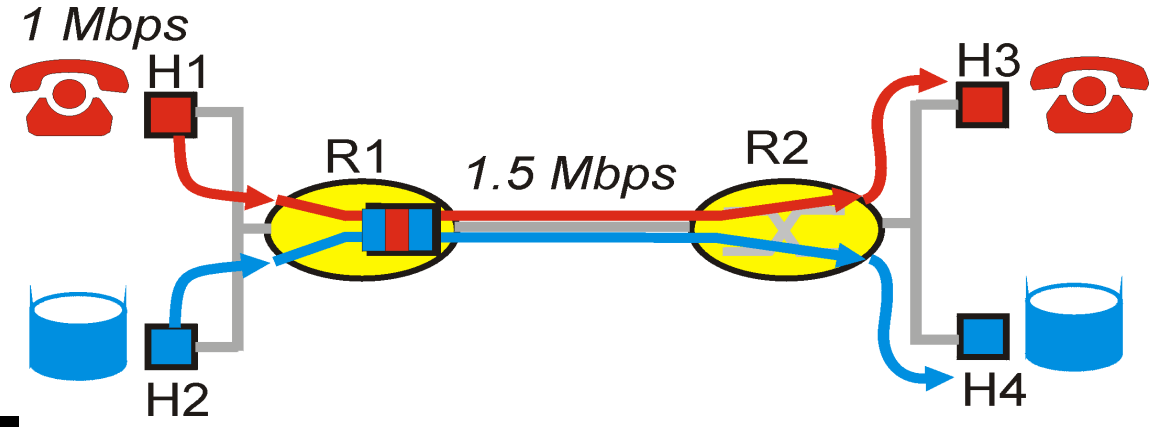
Definitions

- A flow: is a sequence of packets sent between a source/destination host pair that follow the same network path.
- Allocate bandwidth: (on output links) the proportion of the link capacity used by a flow
- Manage queues of packets according to allocation of resources
  - Throughput, loss, delay
- Signals between routers
  - Flow specification
  - Traffic specification
  - Service requests

QoS Principle 1 / 4: Use a marking and servicing protocol

Example: 1Mbps IP phone and an FTP connection share 1.5 Mbps link.

- FTP sender does not know what maximum rate is tolerable → bursts of FTP data traffic can congest router and cause audio loss
- Want to give priority to audio over FTP → how can the router tell which application a packet belongs to?

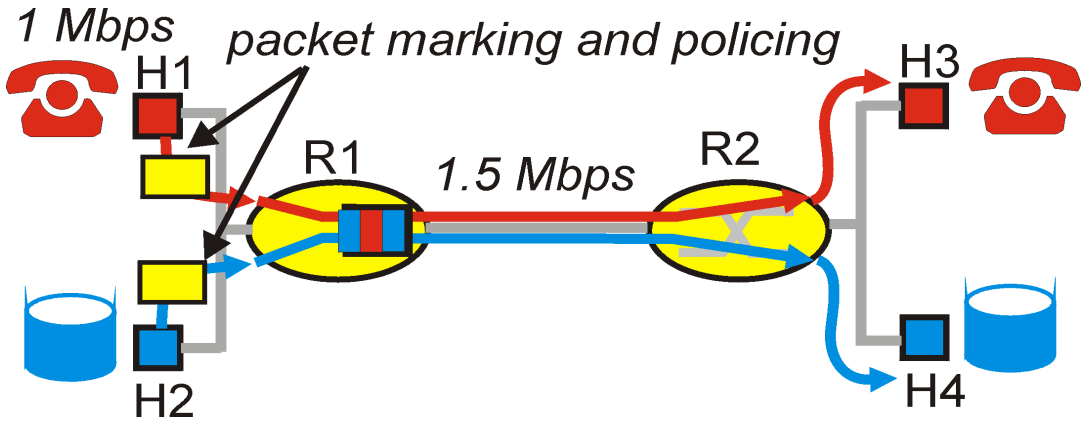


Principle 1

Packet Marking needed for router to distinguish between **different classes**; and new router policy to treat packets accordingly

QoS Principle 2 / 4: Policing: Make flows behave

- What if applications misbehave (audio sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations
- Marking and policing at network edge



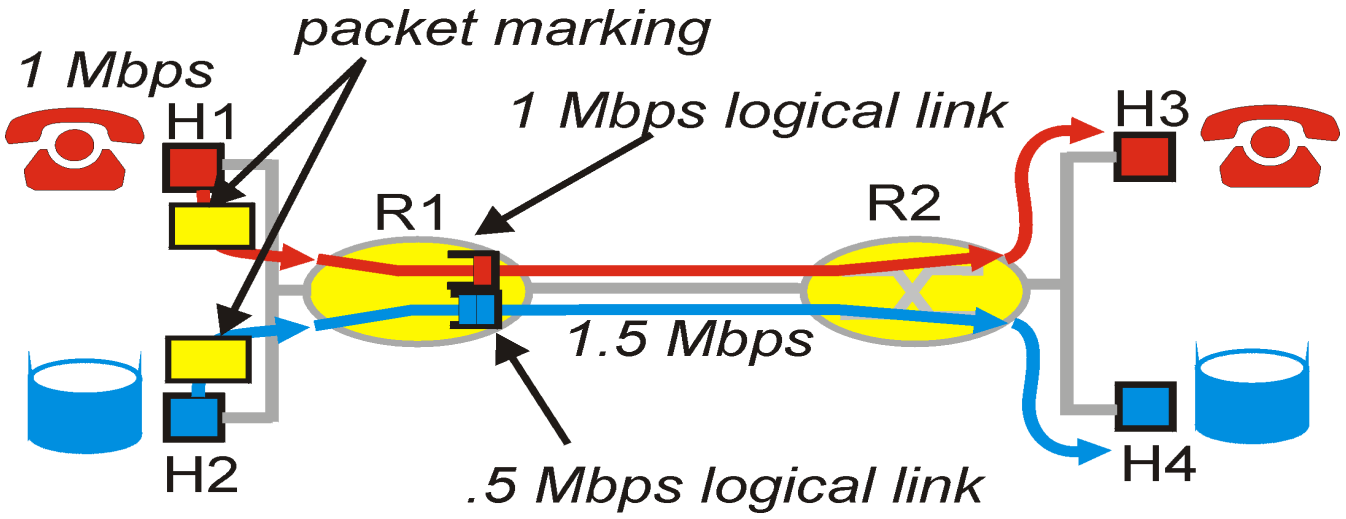
Principle 2

Provide protection (*isolation*) for one flow from others

QoS Principle 3 / 4: Maximise efficiency

Allocating *fixed* (non-sharable) bandwidth to flow:

- *Inefficient* use of bandwidth if flows don't use their allocations

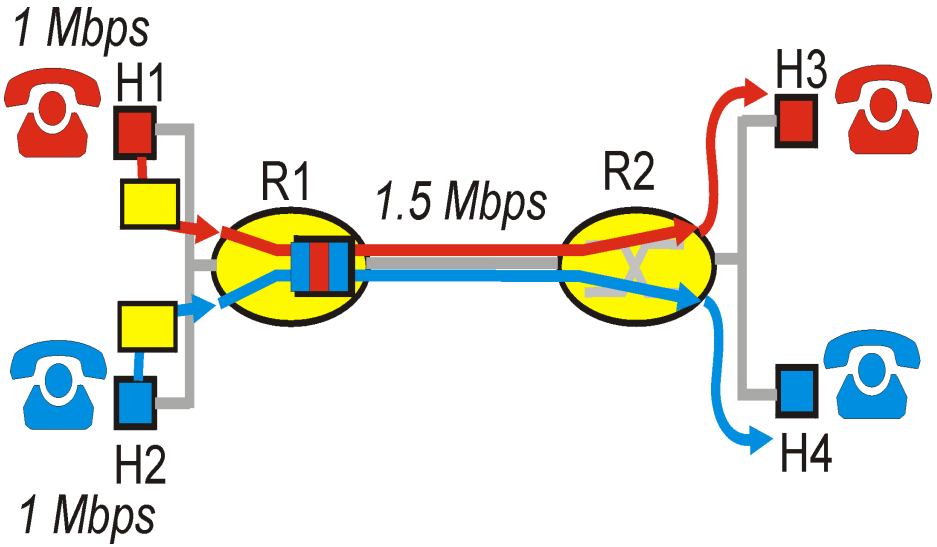


Principle 3

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

QoS Principle 4 / 4: Don't over-allocate

Fact: cannot support traffic demands beyond link capacity!



Principle 4

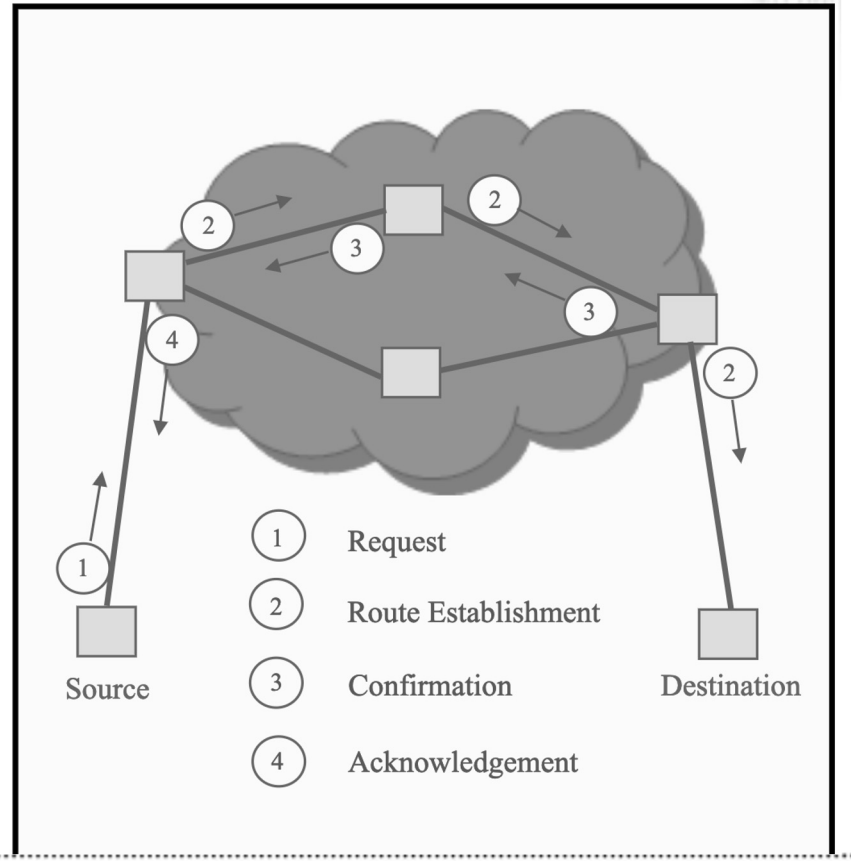
Call Admission: A flow declares its needs. The network may block the call (e.g. busy signal) if it cannot meet the requirements.

Basic functional blocks of QoS

- Pieces of the Puzzle
  - Call Admission/ Resource Allocation
    - SLA (Service Level Agreement)
    - Advertisement of required bandwidth by user to the network
    - Provisioning of bandwidth by network for user
  - Flow Classification
    - Ability of the network to identify incoming packets (flows) and assign them a pre-defined level of service
  - Policing and Shaping
    - Ability of the network to monitor flows to ensure conformance to advertised traffic characteristics and provisioned resources
  - Congestion Avoidance Mechanisms
    - Ability to monitor buffer utilization levels and regulate flow rates to alleviate congestion on network links
  - Scheduling Mechanisms
    - Queueing mechanisms to provision differing levels of service

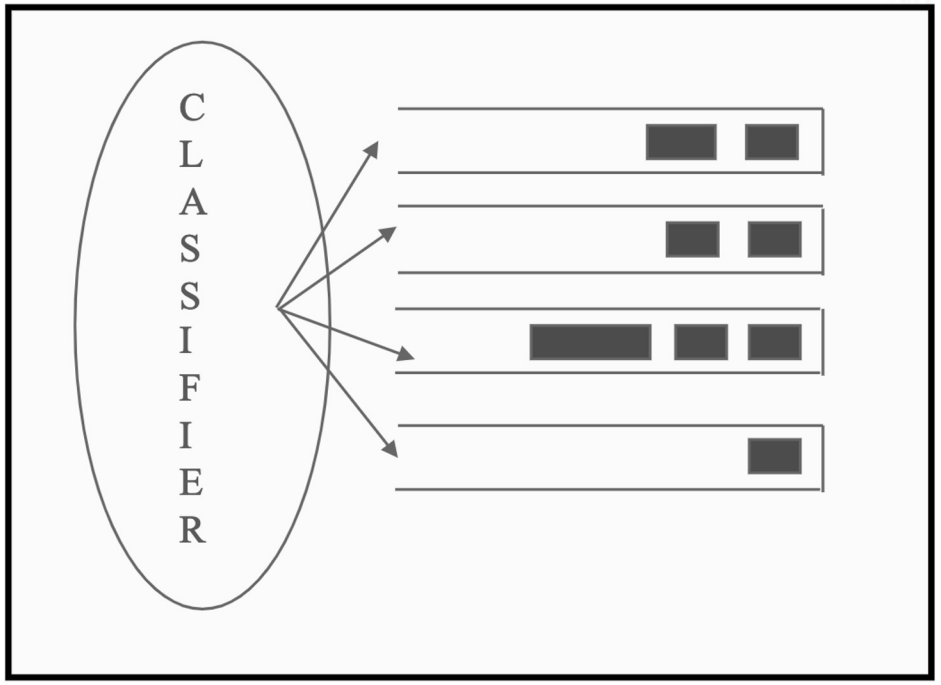
Call admission

- Control Plane
  - Signal the network on type of connection and QoS requirements
  - Network is responsible for pro-active Bandwidth Management
    - Establishing the route of the connection
    - Reserving enough resources to meet QoS requirements
      - Stochastic reservation
      - Virtual pipes
    - Rejecting a call if it does not have enough resources to meet the call
  - Examples:
    - ATM
    - IntServ (RSVP)
    - MPLS



Flow Classification

- *Need a method to identify packets/cells in order to provide differential treatment*
- Examples of Classification Criteria
  - VC number
  - MPLS Label
  - Type of Service
  - Protocol
  - Address
    - Source IP Address
    - Destination IP Address
  - Port Number
    - Source port
    - Destination port
  - Incoming interface

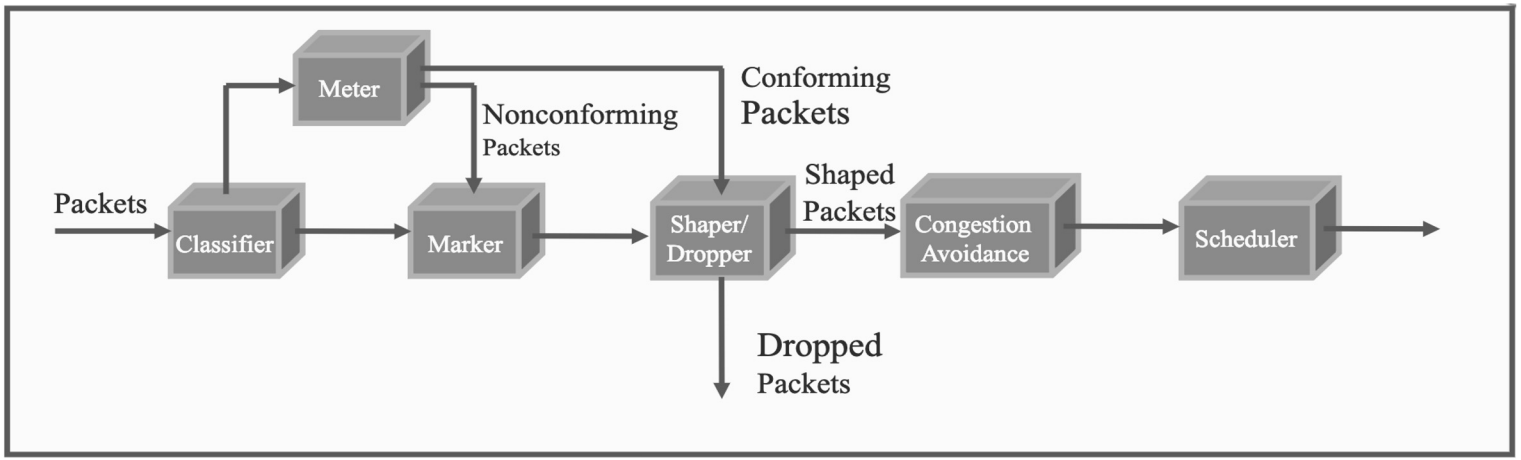


University of BRISTOL

Section 2

Ways of implementing QoS
General purpose approaches

Techniques for QoS



- Classifier: Selects packets based on portions of packet header
- Marker: Marks/Remarks the packet header based on traffic class
- Meter: Checks compliance to traffic profile and passes result to Marker and Shaper/Dropper
- Shaper: Allows for delaying of packets in buffer to enforce compliance with traffic profile
- Dropper: Drops traffic that does not conform with traffic profile
- **Congestion Avoidance:** Checks buffer levels and stochastically drops packets
- **Scheduler:** Allows for differential queueing and servicing of packets

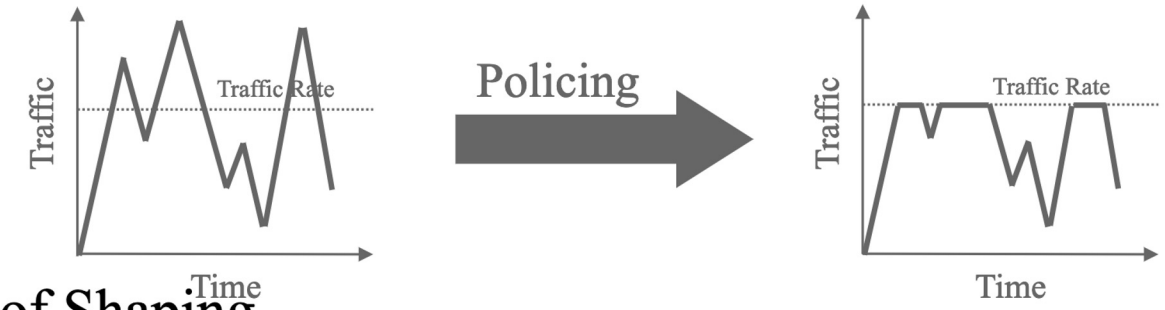
University of BRISTOL

Policing and Shaping

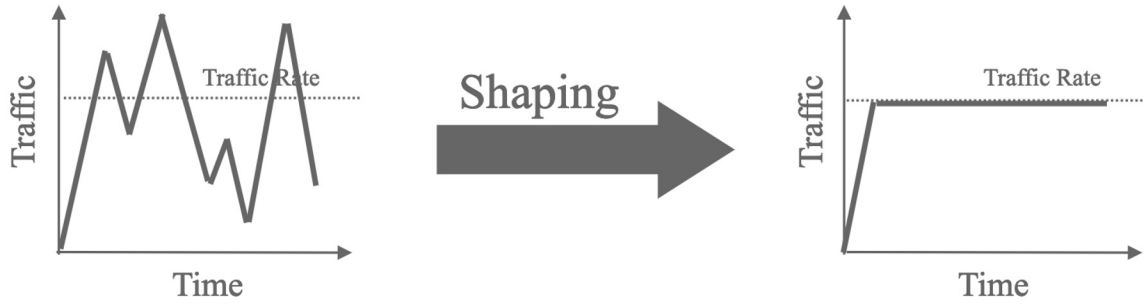
- *Regulate the rate at which a flow is allowed to inject packets into the network*
- Using an appropriate time scale for measurement is important for three metrics
  - Average rate
    - Limit the long term-average rate (packets per time interval) of the flow
  - Peak rate
    - Limit the maximum number of packets that can be allowed into the network over a short interval of time
  - Burst size
    - Limit the number of packets that can be allowed into the network over an extremely short interval of time
    - Limiting case (as time interval approaches zero) defines the number of packets that can be instantaneously sent into the network

Policing and Shaping

- Effect of Policing



- Effect of Shaping



University of BRISTOL

Scheduling problem

- Assume we have K flows, each with its own queue.
- Problem:
  - (vers.1) In what order and for how long do I serve queue ki?
  - (vers.2) In what order and for how many bits I serve from queue ki?
  - (vers.3) In what order and when do I serve a packet from queue ki?
- Scheduling solutions:
  - FIFO (FCFS)
  - Simple priority queuing (also called strict priority)
  - Round-robin, Weighted Round-robin & Deficit Round-robin
  - Weighted Fair Queuing (WFQ)

University of BRISTOL

Simple priority queuing

- *K* queues:
  - $1 \leq k \leq K$
  - queue $k + 1$ has greater priority than queue k
  - higher priority queues serviced first

✓ Very simple to implement

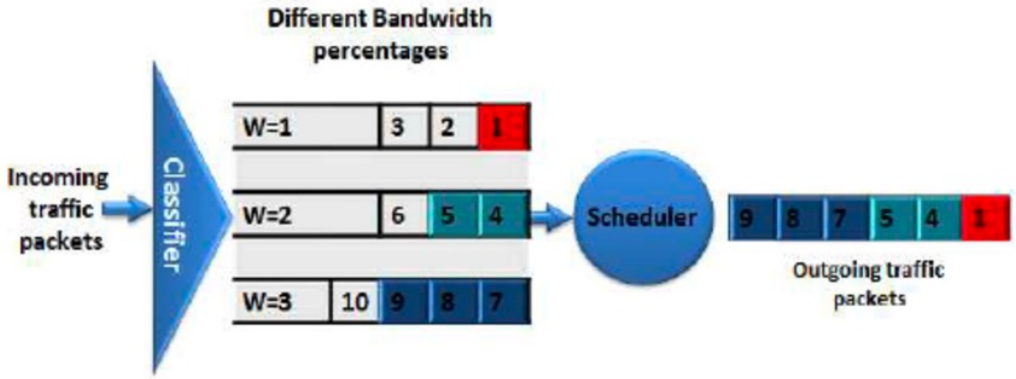
✓ Low processing overhead

- Relative priority:
  - no deterministic performance bounds

✘ Fairness and protection:
  - not max-min fair: starvation of low priority queues

Weighted round-robin (WRR)

- Simplest attempt at **generalized processor sharing (GPS)**
- Queues visited round-robin in proportion to weights assigned
- Different mean packet sizes:
  - weight divided by mean packet size for each queue
- Mean packets size unpredictable:
  - may cause unfairness



- Service is fair over long timescales:
  - must have more than one visit to each flow/queue
  - short-lived flows?
  - small weights?
  - large number of flows?

University of BRISTOL

Deficit round-robin (DRR)

- DRR does not need to know mean packet size
- Each queue has deficit counter (dc): initially zero
- Scheduler attempts to serve one quantum of data from a non-empty queue:
  - packet at head served if size ≤ quantum + dc
    dc ← quantum + dc – size
  - else dc += quantum
- Queues not served during round build up “credits”:
  - only non-empty queues
- Quantum normally set to max expected packet size:
  - ensures one packet per round, per non-empty queue
- Works best for:
  - small packet size
  - small number of flows

If weights are assigned to the queues, then the quantum size applied for each queue is multiplied by the assigned weight.

University of BRISTOL

Weighted Fair Queuing (WFQ)

- Principle:
  - Model of continuous bit-by-bit transmission to produce finish-numbers for packets in queue
  - Serves packets round-robin
- Finish-number:
  - the time packet would have completed service under bit-by-bit scheduling
  - packets tagged with finish-number
  - smallest finish-number across queues served first
- Round-number:
  - execution of round by bit-by-bit round-robin server
  - finish-number calculated from round number
- If queue is empty:
  - finish-number is:
    number of bits in packet + round-number
- If queue non-empty:
  - finish-number is:
    highest current finish number for queue +
    number of bits in packet

Weighted Fair Queuing (WFQ)

$$F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)$$

$F(i,k,t)$: finish-number for packet k on flow i arriving at time t

$P(i,k,t)$: size of packet k on flow i arriving at time t

$R(t)$: round-number at time t

$$F_\varphi(i,k,t) = \max\{F_\varphi(i,k-1,t), R(t)\} + \frac{P(i,k,t)}{\varphi(i)}$$

$\varphi(i)$: weight given to flow i

- Rate of change of $R(t)$ depends on number of active flows (and their weights)
- As $R(t)$ changes, so packets will be served at different rates

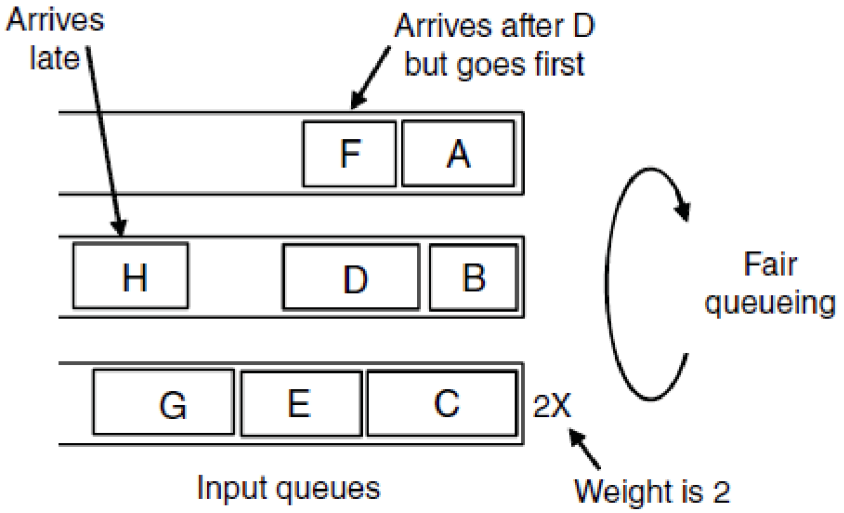
- Flow completes (empty queue):
  - one less flow in round, so
  - *R* increases more quickly
  - so, more flows complete
  - R increases more quickly
  - etc. …
  - **Iterated deletion** problem
    - Transmit avalanche ….
- WFQ needs to evaluate *R* each time packet arrives or leaves:
  - processing overhead

University of BRISTOL

Weighted Fair Queuing (WFQ)

- WFQ is the only scheduler that also proposes packet dropping policy
- Buffer drop policy:
  - packet arrives at full queue
  - drop packets already in queued, in order of decreasing finish-number
- Can be used for:
  - best-effort queuing
  - providing guaranteed data rate and deterministic end-to-end delay
- WFQ used in “real world”
- Alternatives also available:
  - self-clocked fair-queuing (SCFQ)
  - worst-case fair weighted fair queuing (WF2Q)

Weighted Fair Queuing



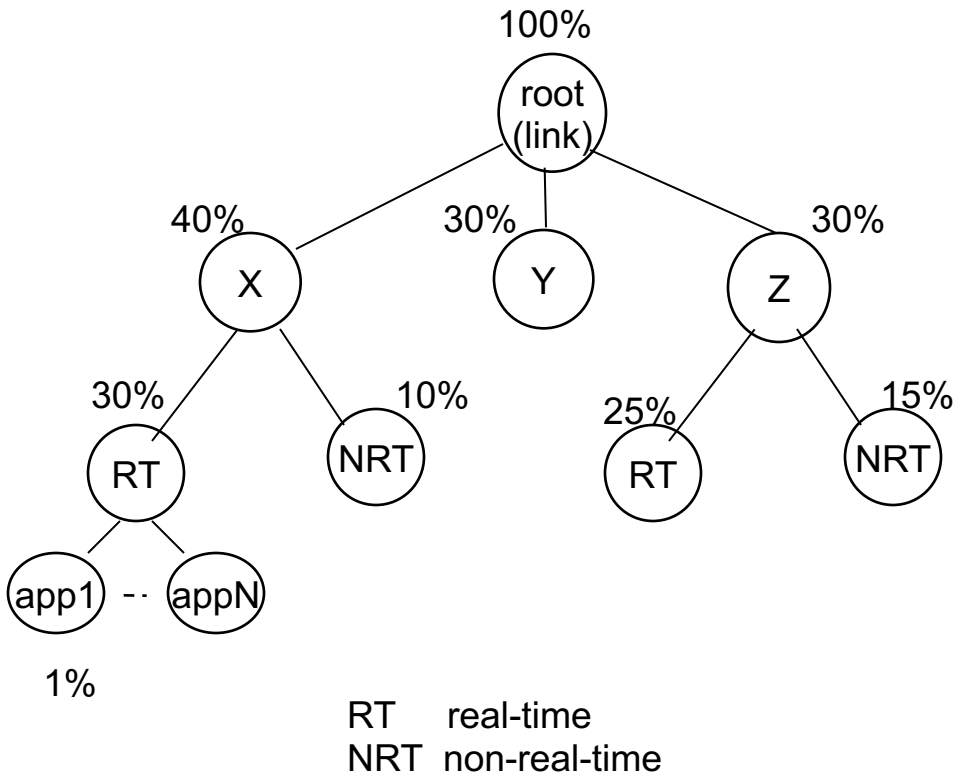
Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

$$\text{Finish}(i) = \text{Max}(\text{Arrive}(i), \text{Finish}(i-1)) + \text{Length}/\text{Weight}$$

https://courses.cs.washington.edu/courses/csep561/10au/lectures/lecture-qos.pdf

Class-Based Queuing

- Hierarchical link sharing:
  - link capacity is shared
  - class-based allocation
  - policy-based class selection
- Class hierarchy:
  - assign capacity/priority to each node
  - node can “borrow” any spare capacity from parent
  - fine-grained flows possible
- Note: this is a queuing mechanism: requires use of a scheduler



University of BRISTOL

Summary

- Quality of Service
- Scheduling
- Queuing
- Policing
- Classification
- Scheduling methods