SCOM/ SRSI
Quality of Service

Ana Aguiar

DEEC, FEUP

2018-19

Contents

- Resource Allocation Mechanisms
  - Queueing, dropping
  - Scheduling
  - Traffic shaping: leaky bucket, token bucket
- QoS service models: IntServ and DiffServ
- MPLS

Resource Allocation

- Resources: links, routers
- Resources shared among different packet flows

- Resource allocation deals with mechanisms to share available resources
  - According to service models
  - Under normal operation and in case of high load

Best Effort Networks

- Packets are treated individually
  - No notion of flow
- Best-effort networks treat all packets alike
  - No service differentiation
- Network and hosts do not “talk” to each other
- Need to react to or avoid congestion
  - Congestion control and avoidance mechanisms

Real-Time Communication

- Communication of audiovisual and multimedia data requires time correlation between sender and receiver
  – Guaranteeing throughput is not enough

- Other applications
  – Control applications: robots, automation, IoT
  – Emergency, safety (e.g. autonomous driving)
  – Interactive data applications, e.g. trading

Real-Time Communication

- Real-time applications require delay guarantees from the network
  - Best-effort networks offer no guarantees

- Hence the need for other service models that can co-exist with best-effort
  - Packets are treated differently, according to the service they require
  - Required service may not always be supported

What is Quality of Service (QoS)?

- Capability of the network to offer service guarantees different from best-effort
- Capability of the network to provide different types of service
- Enabling hosts to request service guarantees

QoS Components

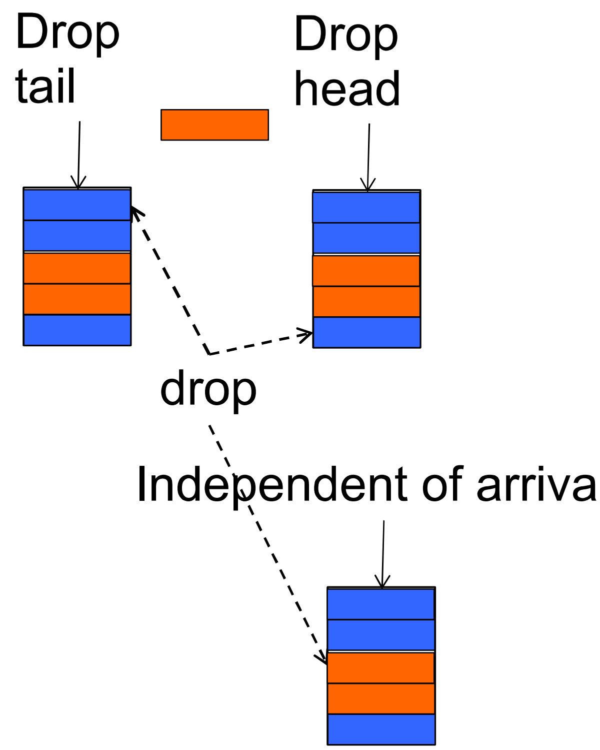
- QoS-capable network should provide
  - Abstraction of application requirements
  - Communication between hosts/ apps and network
  - Mechanisms for implementing service guarantees
    - Admission control
    - Policing of admitted flows
    - Resource allocation
    - Stateful routers

Resource Allocation Implementation

- A few important mechanisms to implement QoS
  - Queueing and dropping discipline
    - How to manage queues at routers and treat packets when queues start to fill
  - Traffic Shaping
    - How to guarantee traffic behaviour
  - Scheduling
    - How to implement a specific service at a router

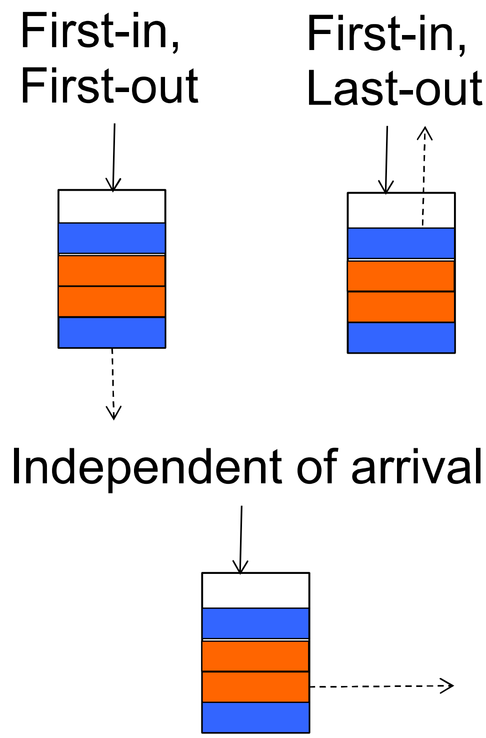
Dropping Discipline

- If queue is full and a new packet arrives, which packet will be dropped?
  - Newest (tail),
  - Oldest (head),
  - Lowest priority
  - Random, …
- Drop before filling up



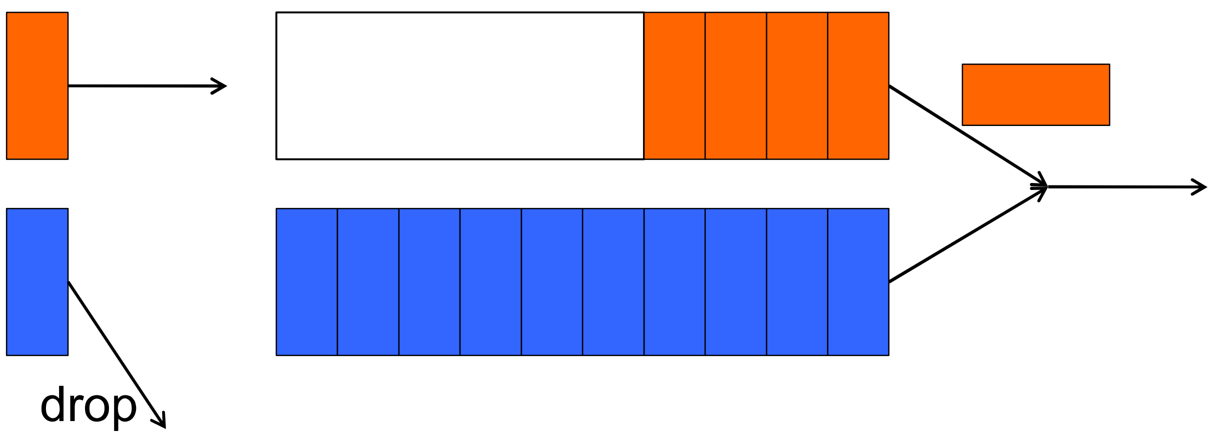
Queueing Discipline

- Which packet in the queue will be served next?
  - Oldest (FIFO)
  - Newest (FILO)
  - Highest priority
  - Depending on deadline
  - …
- Multiple queues



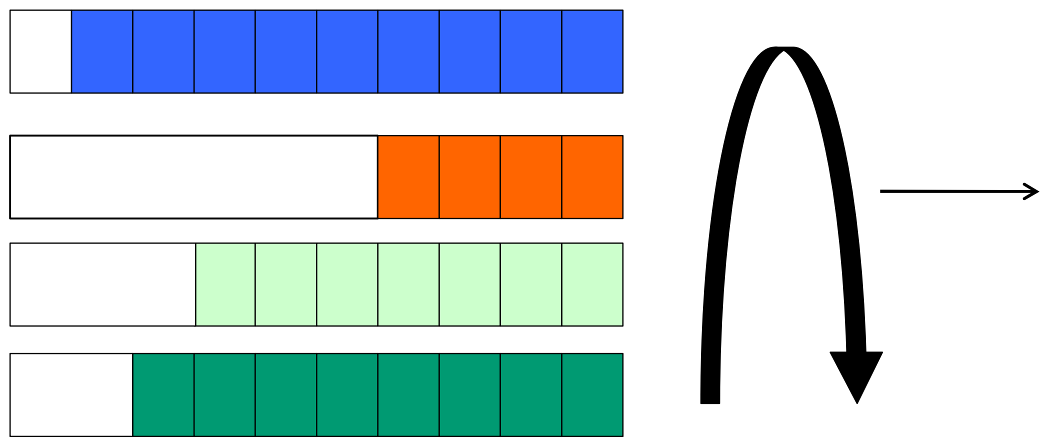
Scheduling: Priority Queueing

- Packets can have different priorities
- Higher priority packets are served first
  - Low priority packets can be starved
- We have other options



Scheduling: Fair Queueing

- Assume fair = equal
- One queue per flow
- Round-robin scheduling implements fair access (= transmission opportunity) to shared resource



Scheduling: Fair Queueing

- Fair queueing is a scheduling discipline
- Not all packets have the same length
  - Round robin is not fair in this case
- Simulate bit-by-bit service using a **virtual clock**

$$F_i = \max(F_{i-1}, A_i) + P_i$$

$$F_{i-1} \ \textit{finish time of packet } i-1$$

$$A_i \ \textit{arrival time of packet } i$$

$$P_i \ \textit{time necessary to transmit packet } i$$

- Choose queue to serve according to simulated packet virtual finish time

Scheduling: Fair Queueing (FQ)

- If n flows have data to send, each flow receives 1/n th of the bandwidth
  – Link is equally shared among backlogged flows

- If only one flow has data to transmit, it receives the whole bandwidth
  – Work conserving scheduling policy: as long as there is data to transmit, the link is not idle

Weighted Fair Queueing (WFQ)

- Extends FQ to support weights per queue
- Each queue receives service according to its weight
  - Instead of transmitting 1/n bits of each queue per clock tick, it transmits $w_i/\mathrm{sum}(w_i)$
  - FQ is a special case of WFQ, where $w_i=1$ for all i

Weighted Fair Queueing

- Enables the implementation of concepts of fairness other than fair ⇔ equal
- But does not specify how to do it
  - How to set weights is policy, not scheduling
  - Weight of a flow does not guarantee a minimum bandwidth to that flow, only a minimum percentage of available bandwidth

Policy != Mechanism that implements it

Exercise

- 3 input flows, 1 output
- Initially, all queues empty
- Name the transmission order for
  - Fair queueing
  - Weighted fair queueing for $w_1=2$, $w_2=w_3=1$

Packet	Size	Flow
1	100	1
2	100	1
3	100	1
4	100	1
5	190	2
6	200	2
7	110	3
8	50	3

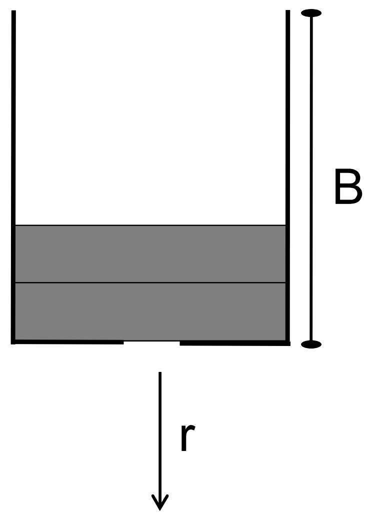
Exercise 10, Chapter 6, in “Computer Networks: a Systems Approach”, L. Peterson, B. Davie

Traffic Policing and Shaping

- Mechanisms:
  - Leaky bucket
  - Token bucket
- Shaping: control traffic entering a network
- Policing: identify traffic that does not conform
- Marking: mark packets that do not conform to certain shape

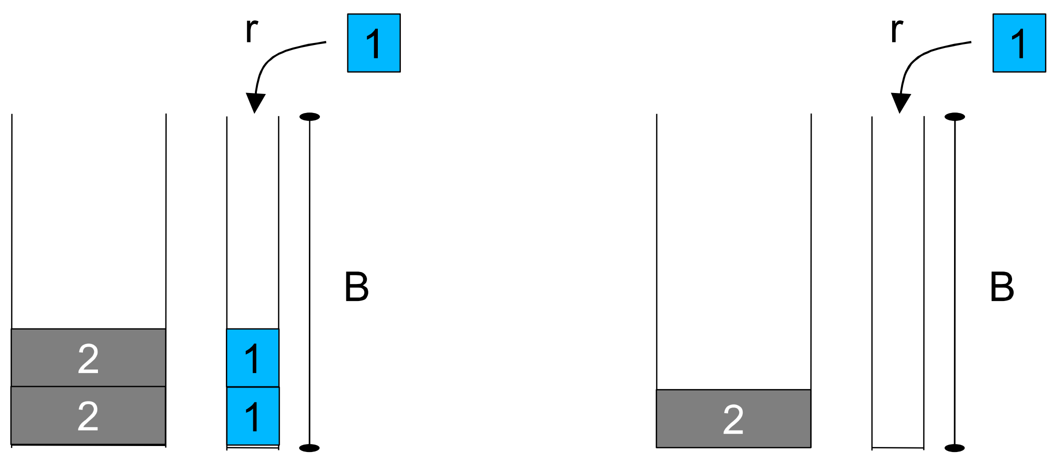
Leaky Bucket

- Single server queue
- Output data rate is constant
- Queue determines bursts that can be accommodated without losses
  - But output data rate is still constant
- Queue full => data loss



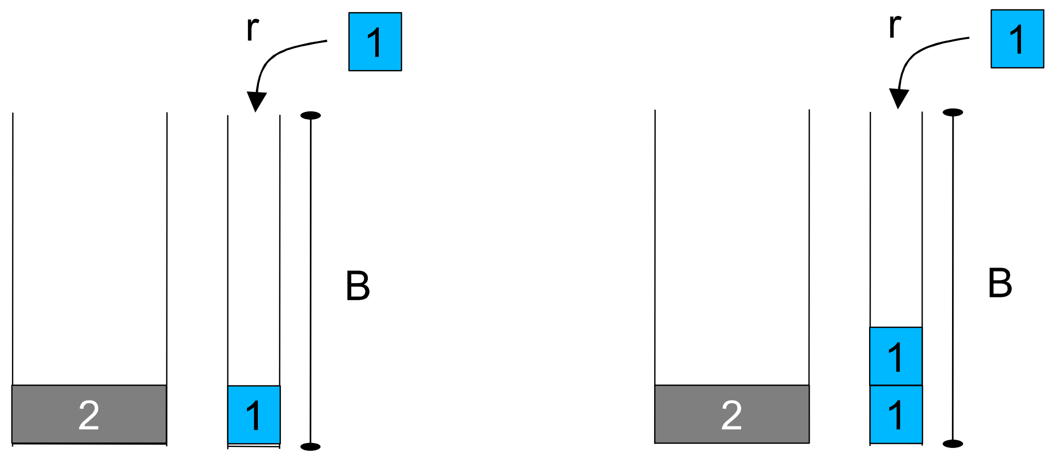
Token Bucket

- Tokens are inserted into bucket at a rate
- Data transmission consumes tokens from bucket
  - Average output data rate equals token rate



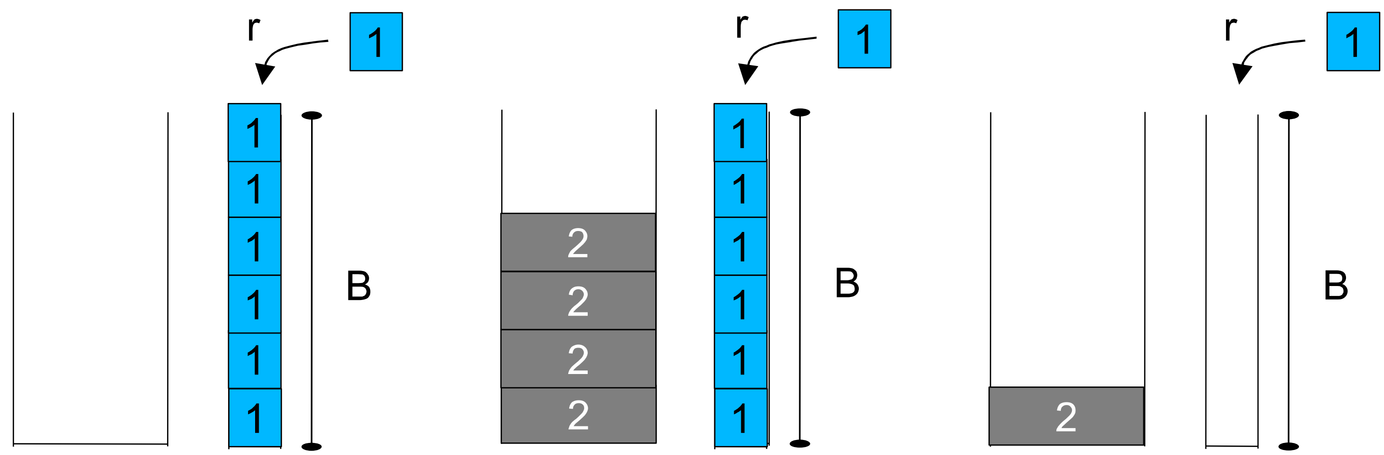
Token Bucket

- Data can only be transmitted for which enough tokens exist in the bucket



Token Bucket

- Size of bucket indicates maximum burst size



Token Bucket: Example

- Flow A
  - average rate = 1 Mbps
  - no variations
- Flow B
  - average rate = 1 Mbps
  - maximum bursts of 2 Mbps for at most 1 s, max 1/3 of the time

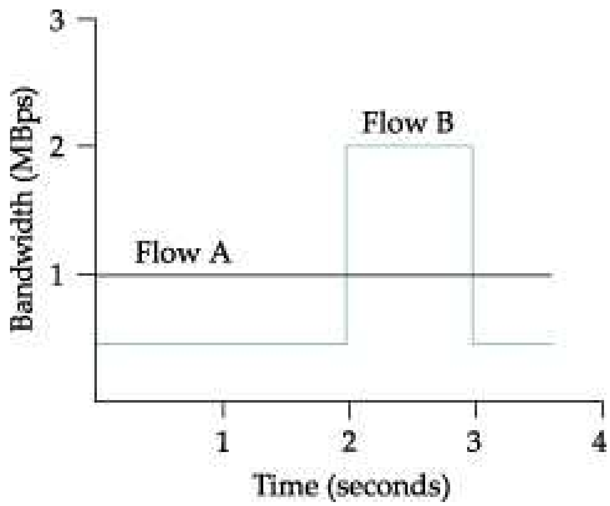


Image in “Computer Networks: a Systems Approach”, L. Peterson, B. Davie

Token Bucket: Example

- Flow A
  - r_A = 1 Mbps
  - B_A = MSS Bytes
- Flow B
  - r_B = 1 Mbps
  - B_B = 1 Mbit
  - Use only half the tokens during first 2 seconds
  - Use saved tokens during burst

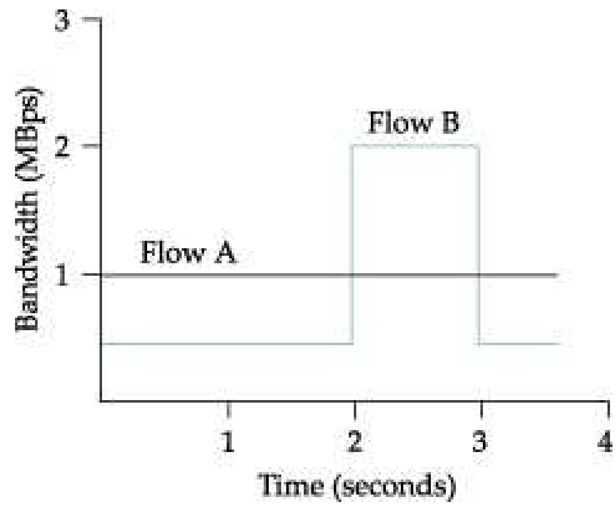


Image in "Computer Networks: a Systems Approach", L. Peterson, B. Davie

Exercise

- Assume processing rate R = 1 packet (of length P) per time unit.
- The input flow has a burst of 10 packets, followed by an idle period of 20 packet times (totally there are 30 packet times).
- Leaky bucket and Token bucket are described with the usual parameters where the bucket size $B = 5 \times P$.
  - What is maximum permissible burst without loss for the leaky bucket? What is the maximum delay for a packet?
  - Show the outputs of the flow shaped by a leaky bucket and a token bucket if the burst arrival of the 10 back packets observe the maximum permissible burst limit. Note the processing rate of a token bucket can be adjusted by the token generation rate.
  - If the maximum permissible burst limit is not followed by the leaky bucket algorithm, what will happen? How about the token bucket algorithm?

QOS SERVICE MODELS

QoS Paradigms

- Support requirements per flow

  Fine-grained QoS support

  Integrated Services (IntServ) and ATM

- Support requirements per class of flows

  Coarse QoS support

  Differentiated Services (DiffServ)

Integrated Services (IntServ)

- RFC 1633, 1994
- Service guarantees per flow
- Service based on end-to-end reservation
- Specification of
  - service classes
  - reservation protocol (RSVP)
  - mechanisms to implement reservation

Service Classes

- Guaranteed service (RFC 2211)
  - Guaranteed maximum delay per packet
  - Designed for intolerant applications
- Controlled load (RFC 2212)
  - Designed for applications that tolerate some loss
  - Provides flow isolation so that flows perceive network as lightly loaded
- Best Effort

Resource ReSerVation Protocol

- RFC 2205
- Signalling protocol for resource reservation

- Provides
  - means for hosts to describe traffic and requirements to intermediate nodes
  - means for routers to inform hosts whether they have enough available resources

Flowspecs

- Specifies flow traffic (TSpec) and flow requirements (RSpec)
- TSpec
  - Used in admission control calculations by the network
  - Should describe traffic as precisely as possible
  - Most flows have varying traffic patterns
    - E.g., recall variations in data rate of coded video
  - Network must accommodate simultaneous bursts
- RSpec
  - Guaranteed service: target or maximum delay
  - Controlled load: requires no additional specification
- Sent using Resource Reservation Protocol (RSVP)

Traffic Description

- Traffic description should be very precise
- Token bucket describes traffic
  - If a flow does not expect bursts, it should advertise a very small bucket size
  - Resources are reserved according to TSpec and RSpec

RSVP: Resource Reservation

- Sender sends TSpec to receiver in PATH message
- Each router saves the reverse path for the PATH message to use later in reverse-forwarding the reservation message

RSVP: Resource Reservation

- Receiver sends RESV message along reverse path, containing TSpec and its own RSpec
- Each router assesses whether it can provide the reservation (admission control)
  - RESV message is sent to previous router in path if yes
  - Error message is sent to requesting receiver if not

RSVP Resource Reservation

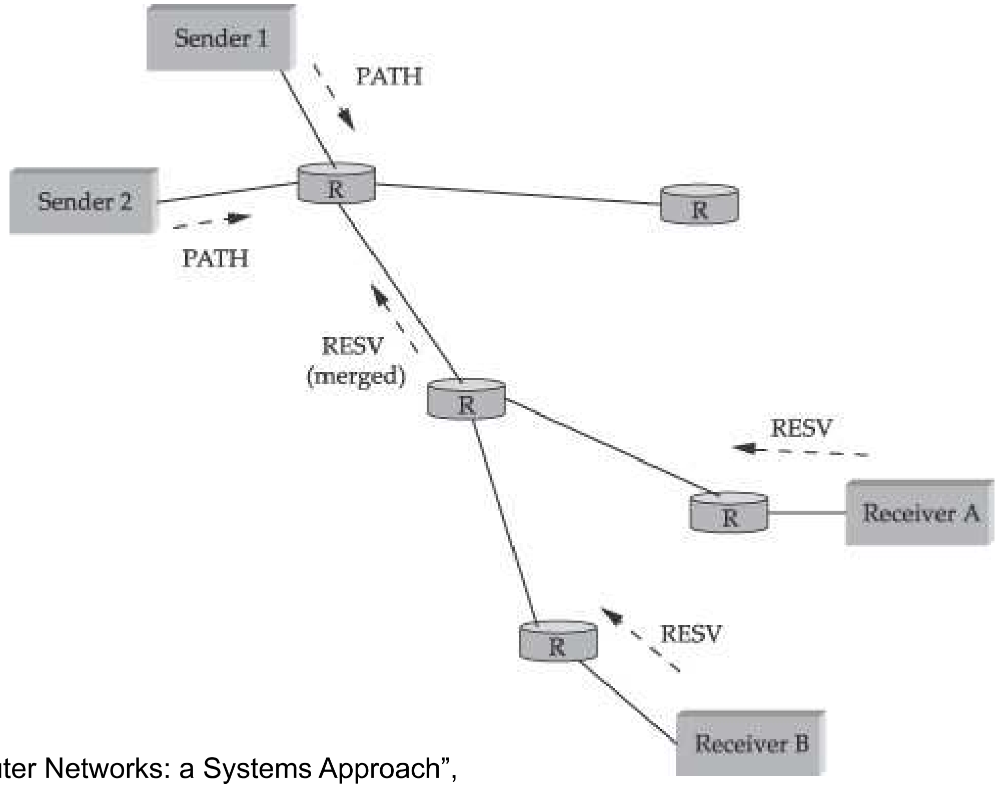


Image in “Computer Networks: a Systems Approach”,
L. Peterson, B. Davie

RSVP: Soft state

- Routers keep per flow soft state
  - state must be refreshed periodically to be kept alive
  - state is discarded in absence of refresh message
  - maintains robustness of connectionless network in case of hosts or routers crashing or rebooting

- Need to periodically refresh state makes adaptation of reserved resource easy

RSVP: Route Changes

- PATH message is re-sent every 30 s
  - Recall: RESV messages also periodically re-sent
- Routers adapt their reverse path according to new PATH message
- If route has changed, next RESV message will be passed along new route and trigger reservation along it

RSVP: Multicast Support

- Multicast
  - More receivers than senders expected
  - Receivers may have different requirements

- Reservation requests are started by receiver

RSVP: Multicast Reservation

- Multicast trees have common path segments to different receivers
- RESV messages are only forwarded if existing reservations are not sufficient
- Otherwise, RESV messages from different receivers are merged

Admission Control

- Mechanism to decide whether a new flow can be accepted
- Both Guaranteed Service and Controlled Load require admission control
- Admission of a new flow should not cause any on-going flow to receive less than requested service
  - Decision depends on FlowSpec and available resources

Policing

- Guaranteeing that an accepted flow does not exceed his TSpec
  - Misbehaviour can lead to not meeting service reservations for other flows
  - Packets out of specification can be always dropped or forwarded anyway as long as they do not interfere with other flows
  - Can you think of a mechanism to implement this?

Packet Classification

- Delivering different service to different packets requires packet classification
- Use source and destination addresses, ports, and protocol number field
  - These fields must be parsed at each router

Policy

- Policies are sets of rules that define the behaviour of mechanisms
  - For example, flows from certain hosts (e.g. plant controller, video conference system) have higher priority, or can preempt others in admission control

Packet Scheduling

- Scheduling is not specified in the architecture or service model

- IntServ is commonly associated with WFQ
  - Each guaranteed service flow has own queue
  - Controlled load services shared one queue

- Implementing different services in a node is very complex and scales badly

IntServ: Scalability

- IntServ has drawbacks
- Routers must
  - calculate admission control for flows
  - classify packets
  - manage different queues and a scheduling algorithm
  - maintain state for all flows
- Reservation maintenance messages are periodically sent by each sender and receiver

IntServ: Scalability

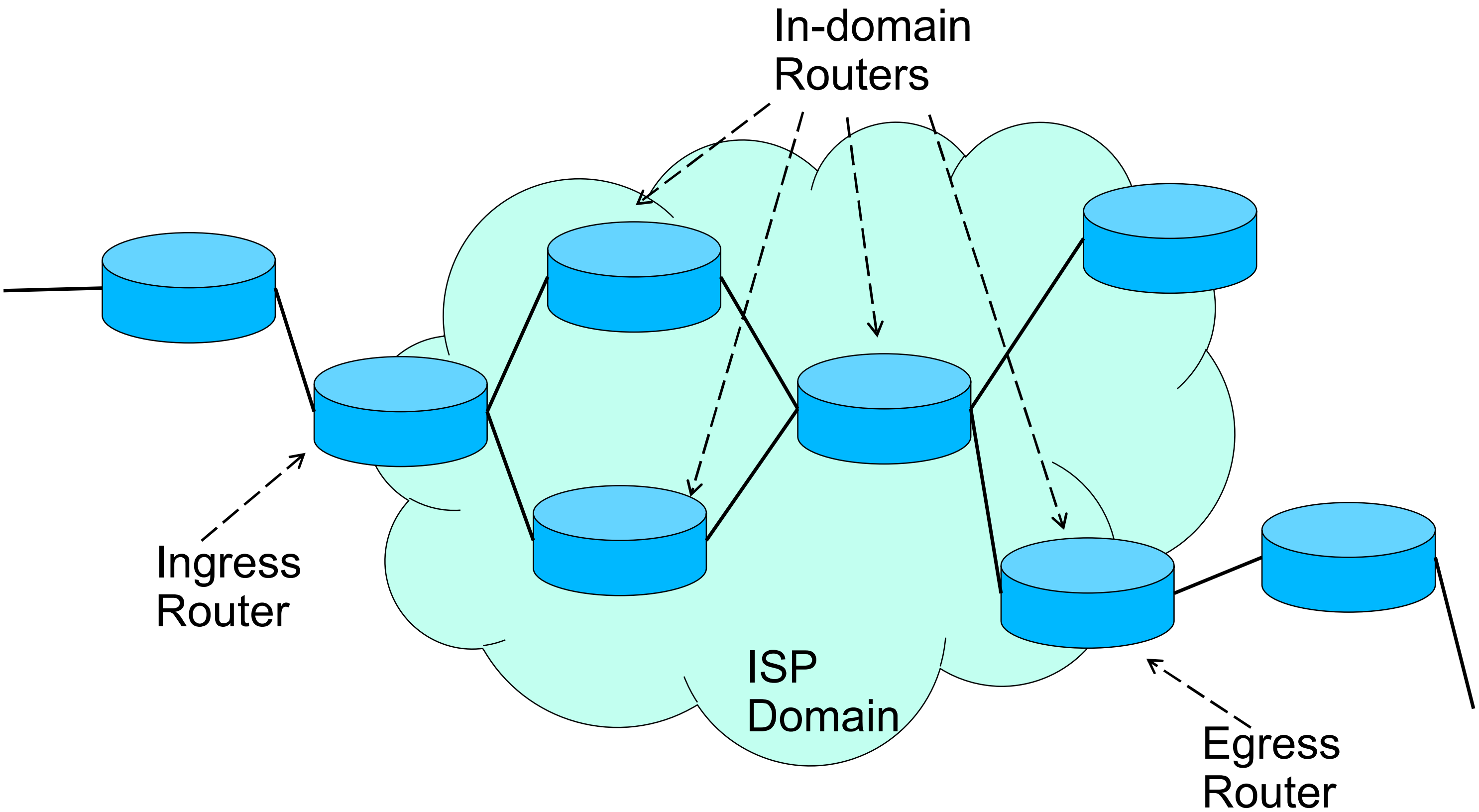
- Take a 10 Gbps link and suppose it is filled up with 64 kbps audio streams
- 156 250 flows
  - To keep state, classify packets, admit or decline
  - And all the other functionalities mentioned
- IntServ is often considered unscalable
  - Is it nowadays still so?

Differentiated Services

- Provides coarser quality of service inside one administrative domain (AD)

- Few pre-defined QoS classes
- Flows aggregated according to service class
- Each packet carries the indication of which service class it belongs to
- Service provided per hop
- Good fit to operator business model

DiffServ in a Domain



DiffServ Architecture

- Service agreement regarding traffic aggregates
  - Service Level Agreement (SLA) between ISP and clients
- Edge routers do traffic conditioning
  - Classify and mark packets according to SLA
  - Do traffic policing (ingress) and shaping (egress)
- Domain routers implement per-hop behaviour
  - Queueing and scheduling

DiffServ: Per Hop Behaviour

- DiffServ implements service differentiation through different per-hop behaviours

- PHB defined for each packet: fully stateless

- Use header field to differentiate packets
  - Re-define Type of Service (TOS) byte in IP header
  - 6-bit DiffServ Code Point (DSCP) field defines the per hop behaviour

Expedited Forwarding (EF) PHB

- Packets forwarded as soon as possible
  - Building block for low loss, low delay, low jitter

- Rate of EF packets arriving at a router should be lower than router capacity
- Requires some form of admission control
  - At edge router, before marking packets as EF
  - Example admission control $\sum\lambda_{EF} < \min_i \{r_i\}$, r_i is the rate of link i in the network

Expedited Forwarding (EF) PHB

- Implemented by queueing policies
  - Priority queueing with strict priority for EF and AF served with WFQ

  OR

  - Weighted Fair Queueing with much larger weight for EF queues
    - WFQ can guarantee that other traffic will not be starved at the cost of slight service inaccuracy
    - EF traffic may not always get lowest possible delay

Assured Forwarding (AF) PHB

- AF PHB offers services that are extensions to RED
  - RED In and Out (RIO)
  - Weighted RED (WRED)

DiffServ Building Blocks

Ingress routers

- Classification
- Policing

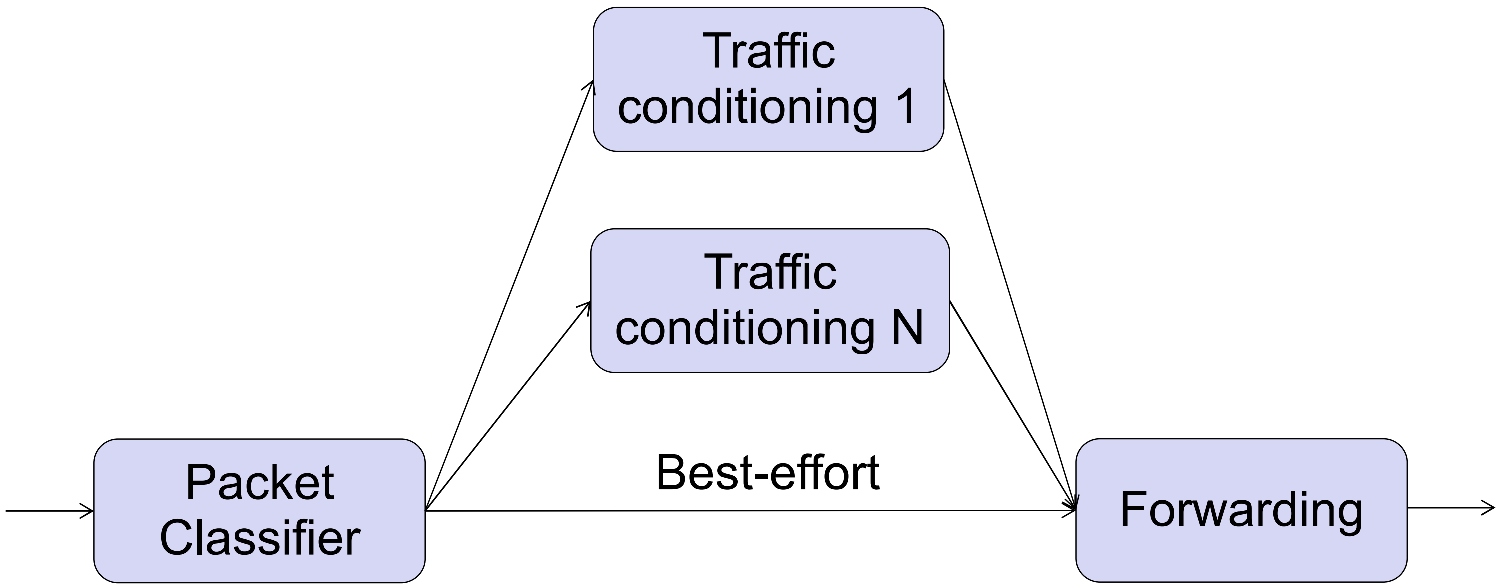
Domain routers

- Flow isolation
- Queueing
- Scheduling

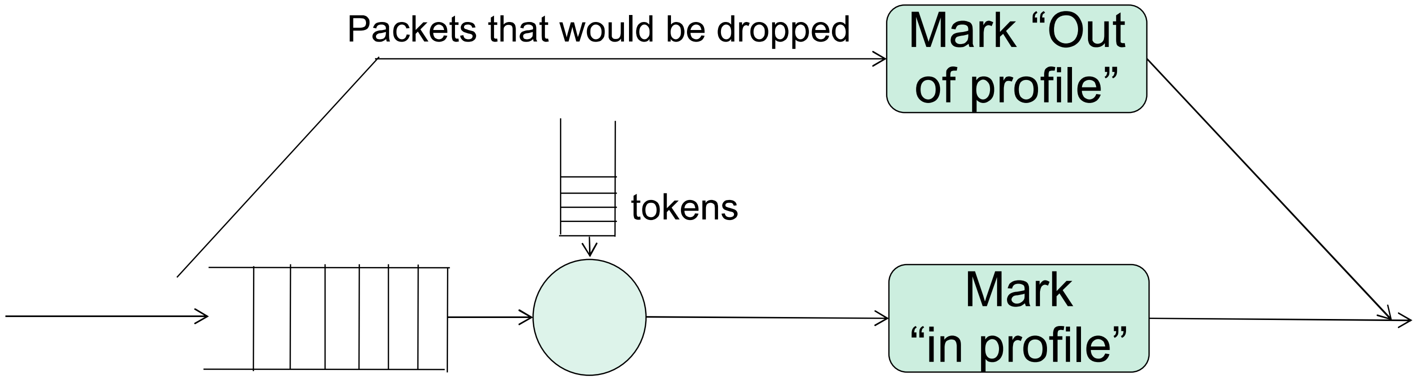
Egress routers

- Traffic shaping

Ingress Edge Router



Traffic Conditioning



Classification

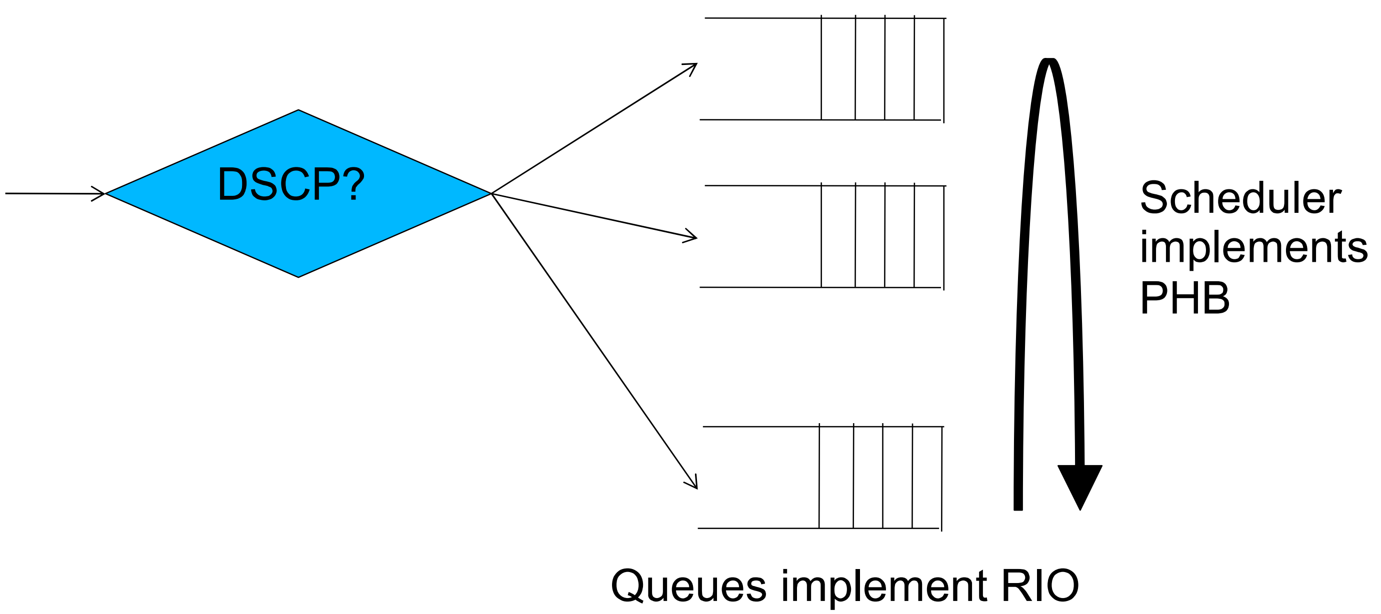
- Classifier at ingress router identifies and marks packets according to the service class

- Packets can be classified based on
  - sending domain
  - transport protocol
  - … (header fields)

Policing

- At ingress router, measure amount of traffic arriving for each flow aggregate
  - Use token buckets
- Mark packets as in or out of profile
  - Routers in the network can decide whether they can handle out of profile packets or not
  - Enables better resource sharing than dropping
  - RIO, WRED (later on)

Domain Router



Flow Isolation

- Service delivered to one class should be independent of service delivered to another

- Packets in different service classes are placed in different queues at each router

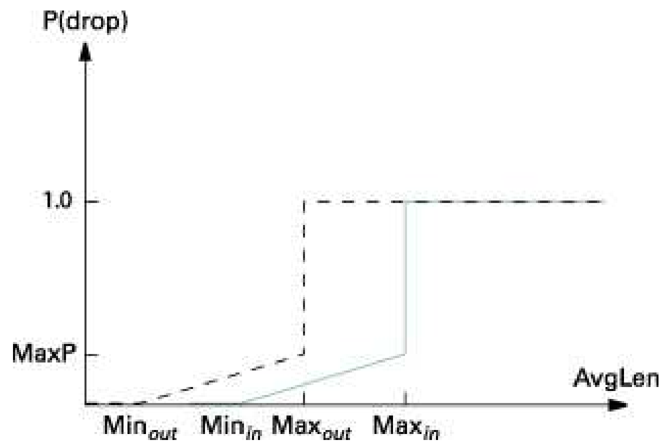
DiffServ with Priority Queueing

- EF packets have highest priority
- AF has several queues with different service

- If packets of an aggregate use different queues, packet re-ordering can happen

RED In and Out (RIO)

- In or Out refers to the traffic profile agreed with the sender
  - Incoming traffic will be marked as In or Out of profile
  - Traffic exceeding agreed volume will follow Out curve drop probability, and be dropped earlier at the domain routers



RIO Considerations

- RIO will try to keep congestion low so that In profile packets are rarely dropped
- Nevertheless, performance depends on overall bandwidth utilisation
  - AF and EF co-exist in the domain
- Performance depends on parameterisation and admission control

Weighted RED

- More than two drop probability curves can provide more than two classes of service

DiffServ with WFQ

- Different weights are used to provide different levels of service
- Difficulty lies in setting the queue weights
  - Queue weights for premium traffic can be set very high, but there is a certain margin of error in service
- WFQ can be combined with WRED
  - Packets in each queue can have different drop probability curves

DiffServ Example

- EF class whose packets are forwarded ASAP
  - Because of this, only in-profile EF packets can be admitted into the domain
- 4 AF classes with 3 drop levels each
- 1 Best Effort class

- No standard for bandwidth allocation
- No packet re-ordering within a class

Traffic Shaping

- At egress routers
  - Traffic arriving from network is bursty and irregular
  - Leaves domain shaped according to leaky bucket
    - Regular data rate
    - Queue size determines acceptable burstsize/ dropping
- Guarantees that domain is “well-behaved” towards others

EF vs AF PHB

How would you classify the discussed applications?

VoIP, VoD, WWW, file transfer, control…

Admisssion Control

- Before accepting a flow for a requested service, verify that the network can deliver that service
  - IntServ and EF PHB require admission control
- How aggressive should it be?
  - Too conservative can lead to low resource usage
  - Too aggressive can lead to unfulfillment of promised service agreements
  - Depends on the goal of running a network
    - Serve as many flows as possible?
    - In terms of throughput or delay?
  - Maximise total throughput? Minimise delay?
  - …

DiffServ vs IntServ

- Type of service that can be delivered
- Granularity
- Setup of the service
- Service based on reservation vs provisioning
- Scope of service: local vs end-to-end

MULTI-PROTOCOL LABEL SWITCHING

Multi-protocol Label Switching

- Multi-protocol Label Switching (MPLS)
  - Between link and network layer
  - Supports multiple protocols underneath and above
- Labels (“shim”) are added to packets according to Forwarding Equivalence Class (FEC)
- Label Switching Routers (LSR) forward according to label

MPLS

- Labels can be added to packets or flows
- Packets can have more than 1 label
  - To enable multiple routing levels
- LSRs forward according to “last in” label
- LSRs can swap labels
- Class of Service can be inferred from the label

MPLS

- Hop-by-hop routing
  - Regular routing, e.g. OSPF
  - LSRs create labels and distribute them among peers
  - LDP: Label Distribution Protocol
- Explicit routing
  - Ensures that a certain FEC follows a pre-determined path
  - Use reservation protocol to distribute labels (RSVP-TE)
    - Label allocated by receiving node
    -

NEXT WEEK: NETWORK NEUTRALITY