Computer Networks

Lecture on

Queuing & Scheduling, Quality of Service

Plan of This Lecture

- What and what for is queuing & scheduling?
- Queuing disciplines – algorithms
- QoS approaches in IP networks

What Is Queuing & Scheduling?

Queuing – deciding if and where to store a packet

Scheduling – deciding what packets to get sent & when

Buffer-bloat – excessive delay due to excessive queue capacity

- Queuing delay potentially becomes much larger than propagation delay
- Leads to high delay variability – jitter
- Problem for:
  - real-time traffic like voice and interactive video
  - fast web-page loads

Queuing disciplines – queuing & scheduling algorithms
provide traffic management via flow priorities

Queues:

one or more

- per input interface
- per output interface
- per flow
  <protocol, source addr., source port, dest. addr., dest. port>
- per class of flows
- per client (identified by: interface, VLAN, VPN)

What for Deploy Them?

- Mixing
  - real-time and bulk traffic
  - more and less important traffic
  - higher and lower paid packet flows

- Protecting against – by smart decision which packet to drop
  - network congestion
  - flow starvation

- Shaping incoming flows
  - delaying packets that are arriving too fast
  - dropping packets that excide contracted bandwidth

More efficient network usage than Time/Frequency Division Multiplexing

Without them

all traffic is sent on a best-effort basis and routers handle all traffic on an equal footing

Queuing Disciplines

Which packet and when should be dropped?

- Tail-drop – arriving packet is dropped
  - while the output queue is full
  - just before it is full – at the congestion "knee" about 70% of queue capacity
  - Random Early Detection – to drop an occasional packet much earlier when the queue is less than half full
- Head drop – oldest packet is dropped – not used for router queues
- Random drop (randomly selected from the queue packets + the arriving) – seldom used
  - lower risk of starvation for one TCP connection

Which packet and when should be sent?

- FIFO (First-In, First-Out) queuing – one output queue
- Priority queuing
- Fair queuing
- Weighted fair queuing
- ...

Token Bucket Filters

Bandwidth limit is defined for a sender

Packet that excides the limit can be

- dropped – aka. traffic policing
- marked as non-compliant – marked packets can be dropped by congested routers
- delayed to satisfy the limit – aka. traffic shaping

Token bucket algorithm

- ➢ analogy to bucket of tokens B – size of the bucket T – number of tokens in the bucket
- ➢ token – represents unit of bytes
- ➢ incoming packet size → N – number of tokens representing the packet

- A token is added to the bucket every 1/r seconds
- If bucket is full, the token is discarded
- If a packet arrives than
  - if $N < T$ then $T = T - N$ and *normal packet processing*
    else *special packet processing*

Variants of the Algorithm

Leaky bucket as a meter

- the most often used
- output traffic can be bursty
  - burst allowance corresponds to the bucket size

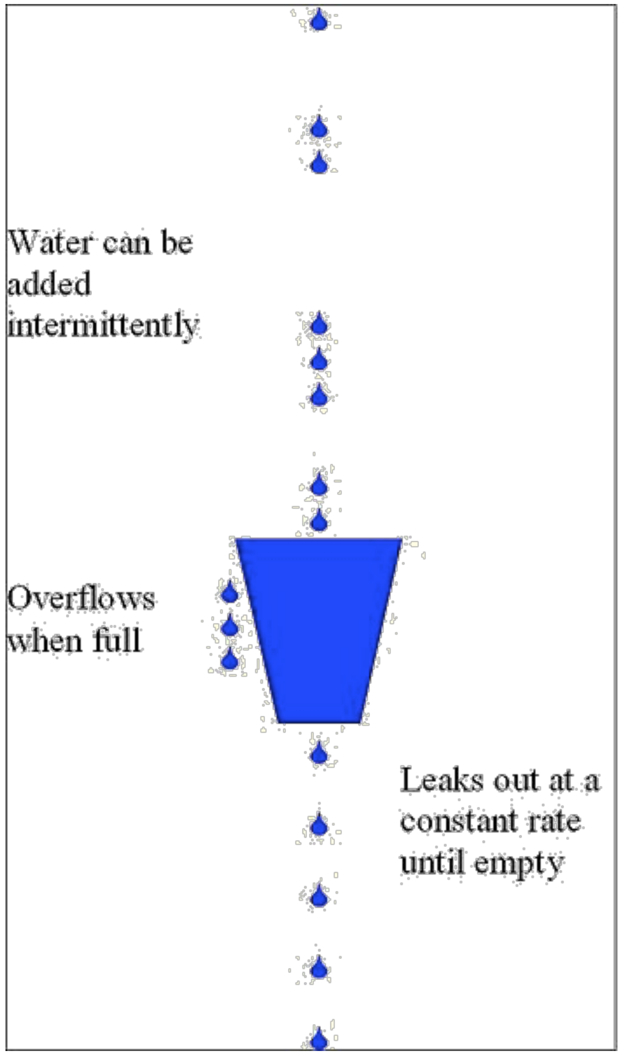
Leaky bucket as a queue

- is more appropriate to traffic shaping in the network interfaces of hosts
- can slowdown a flow needlessly when the network is free

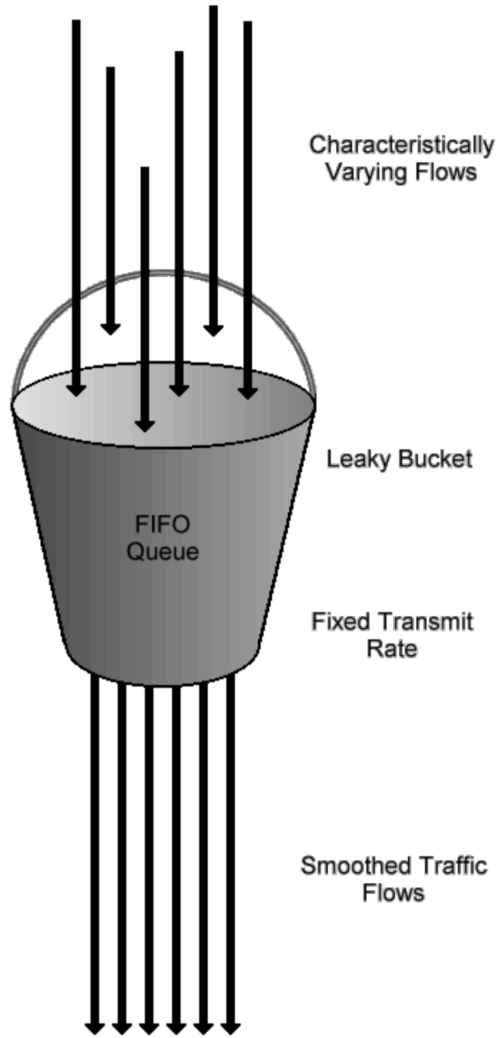
Hierarchical token bucket

- a bucket per traffic class

Leaky bucket as a meter



Leaky bucket as a queue



source: en.wikipedia.org/wiki/Leaky_bucket

Token Bucket Parameters

r – token fill rate ~ packets/s or bits/s

B – buffer capacity ~ packets or bits

A – action if algorithm condition is not satisfied

A ∈ {drop, mark, delay}

For flow(s) passed by the bucket

- **r** is the limit of bandwidth consumption
- **B** is the max burst size
- **r / B** is the max delay of the buffer ~ router

Two or more token-buckets could be applied for a flow, e.g.:

- to limit peak and average transmission rate

  not TB(r_{av}, B_{av}) and not TB(r_{peak}, B_{peak}) ⇒ drop

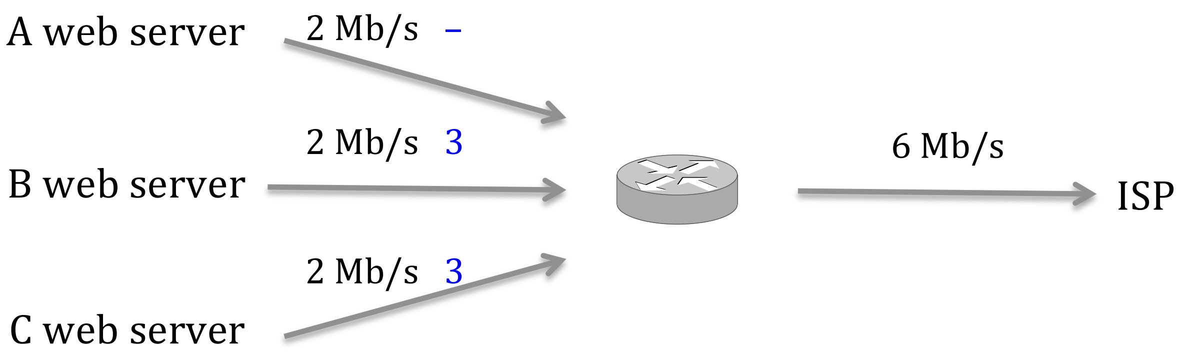
- to mark packet above committed rate and to drop packet above excess rate
  - TB(r_{CIR}, B_{CIR}, mark) next TB(r_{EIR}, B_{EIR}, drop)

Priority Queuing

- Several queues of different priorities
- Lower-priority packets are sent only when there is no waiting higher-priority traffic
- Can lead to complete starvation for the lower-priority traffic
- If higher-priority traffic is bandwidth constrained, than no risk
    - e.g. real-time traffic that consumes less than 10% of the available bandwidth
    - VoIP has intrinsic limitations
    - admission control can be applied – they drop packets
    - traffic shapers can delay some packets to enforce an administratively imposed bandwidth cap

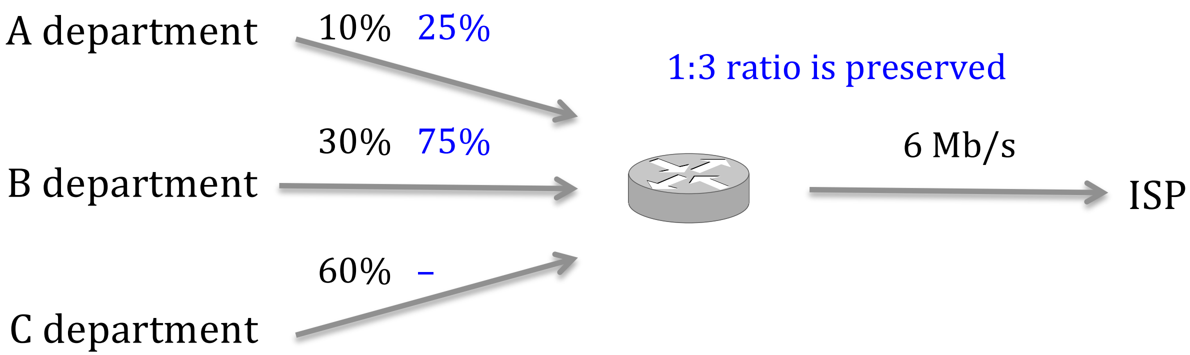
Fair Queuing

- Divides bandwidth among multiple senders
- As many queues as traffic classes
- Pops in round-robin manner from the queues
  – sending one packet from each in turn
- Traffic classes have equal share of output bandwidth if packets are of the same size!



Weighted Fair Queuing

Divides bandwidth among multiple senders according to pre-set percentages



If packets are of different size

- simple *fair queuing* or *weighted fair queuing* are insufficient
- many algorithms are proposed to deal with

QoS – Quality of Service

QoS parameters

- network availability
- bandwidth
- delay
- jitter
- loss rates

They are crucial for real-time and many business applications

ISP's problems

- How to charge endpoints for their QoS requests, when the endpoints are not direct customers?
- How to compare special traffic like multicast with standard unicast traffic?
- Supporting QoS by network devices creates costs
  - QoS per flow is very expensive
  - QoS per traffic class – not so

Very high throughputs of today's interfaces minimise the need for QoS services

IntServ – Integrated Services

Integrated Services – means simultaneous transmission of voice, video, and other data

IntServ architecture

- developed by the IETF in the late 1990's
- provides a fine-grained QoS – per flow which needs some QoS guarantees
- defines
  - router QoS mechanisms
  - RSVP – Resource ReSerVation Protocol
- is not supported by backbone ISPs
  - difficult to scale for rising number of QoS reservations
- difficult to preserve reservations between administrative domains
- supported by access networks
  - they are more prone for congestion
- supported by many corporate intranets, e.g. for VoIP

RSVP

- Created to support multicast services, e.g. teleconferencing
  - thus receivers request reservations
- Supports unicast flows
- Reservations are for one direction
  - bidirectional traffic needs two reservations
- Reservation is identified by source & destination IP addresses & port numbers
- It is not necessary that every router along the path be RSVP-aware

RSVP messages

- PATH message
  - carries sender requirements
  - creates a reservation context at each router
  - collects router addresses – the path
    - IP routers do not guarantee the same path for two directions
    - additional record is created in the routing table
  - reservation can be refused – admission control decision
    - it the router cannot grant the parameters

- RESV message
  - confirms the reservation & receiver's readiness
  - goes back the path
  - is repeated periodically, e.g. every ~30 s

Alternative to IntServv – Content-Delivery Networks

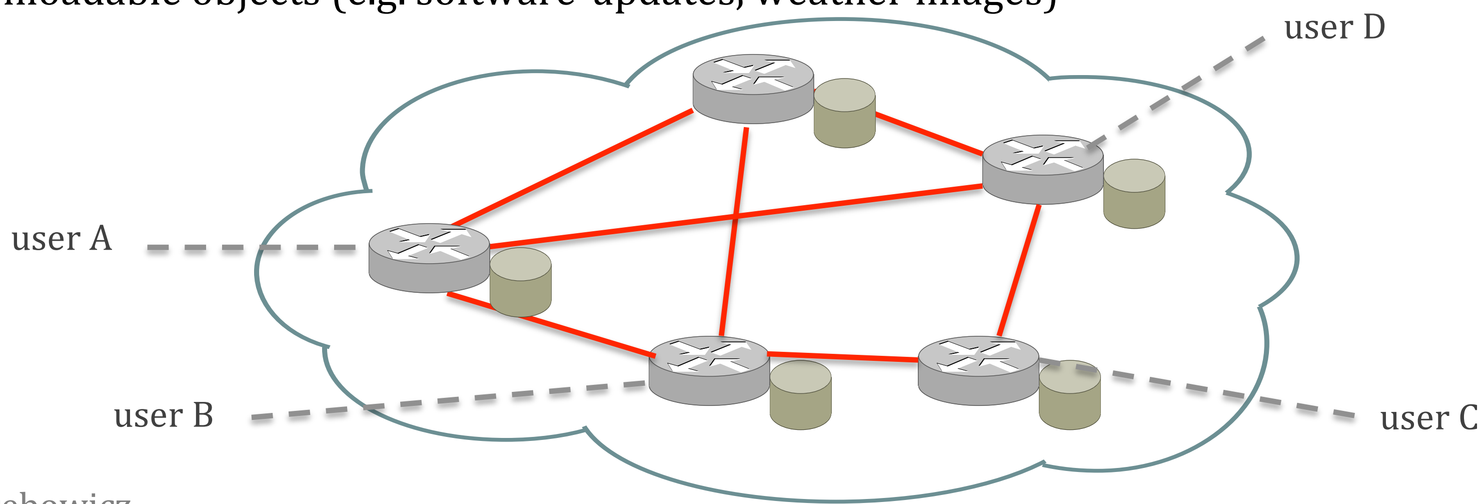
CDN's operator is a large customer for the ISPs

CDN nodes

- overlay network – over the ISP networks
- interconnected by virtual private links of granted QoS parameters – in most deployments
    - however user-to-CDN link has not granted QoS
- can support QoS reservations & multicast for CDN users

Widely used for distribution of

- live & on-demand streaming media
- social media site content
- cloud applications (e.g. e-commerce)
- downloadable objects (e.g. software-updates, weather images)



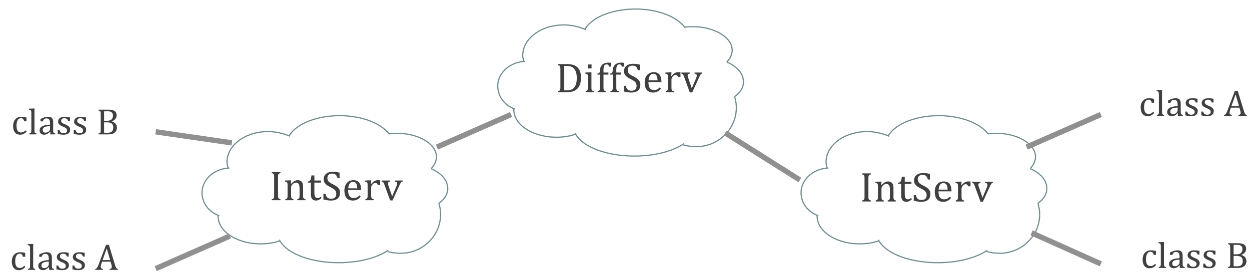
DiffServ – Differentiated Services

The idea – to replace IntServ's many reservations with a small number of service classes

- a few premium
- one best-effort for everyone's bulk traffic

- Reservation is not granted – to a significant degree depends on statistics
- Easier to preserve reservations between administrative domains, than for IntServ
  - A DiffServ domain may consist of a single ISP or several ISPs
    set of routers agreeing on a common DiffServ policy
- Packets are marked for premium service only at the border routers of a DiffServ domain
- Interior routers execute the same scheduling

Just an example:



DiffServ Code Point

000 000	default PHB (best-effort delivery)	
001 ddd	AF class 1 (the lowest priority)	*Assured Forwarding*
010 ddd	AF class 2	
011 ddd	AF class 3	
100 ddd	AF class 4 (the best)	
101 110	Expedited Forwarding	
101 100	Voice Admit	committed rate << interface rate
11x 000	Network control traffic (RFC 2597)	
xxx 000	Class Selector (traditional IPv4 Type-of-Service)	

Excess traffic is marked with drop precedence

ddd	drop precedence
010	do not drop
100	medium
110	high

AF classes can be bought as gold, silver and bronze ISP services

average rate, committed and excess burst (bucket) capacities are negotiated with ISP

Summary

- What is queuing & scheduling?
  - What for deploy them?

- Queuing disciplines
  - Token bucket filters
    - Variants of the algorithm
    - Token bucket parameters
  - Priority queuing
  - Fair queuing
  - Weighted fair queuing

- QoS *Quality of Service*
  - IntServ *Integrated Services*
    - RSVP
  - Alternative to IntServ – Content-Delivery Networks
  - DiffServ *Differentiated Services*

Questions

1. Which algorithm can be used to impose an absolute bandwidth limit?
2. Which algorithm can be used to divide equally the outbound bandwidth among several senders?
3. Which algorithm can be used to divide the outbound bandwidth among several senders according to a predetermined ratio?
4. Which algorithms should be used to enable real-time traffic together with bulky data over the same network?
5. What are the results of excessive queue capacities in routers?
6. Why should we drop packets before the output queue is full?
7. How does the token bucket algorithm work?
8. How does the priority queuing mechanism work?
9. How does the fair queuing mechanism work?
10. How does the weighted fair queuing mechanism work?
11. Why IntServ is not scalable?
12. Which mechanisms is more expensive for an ISP: IntServ or DiffServ? Explain why.
13. How does the Resource ReSerVation Protocol work?
14. Why does RSVP exchange PATH and RESV messages every 30 s?
15. What is a Content-Delivery Network?