Fair Queuing

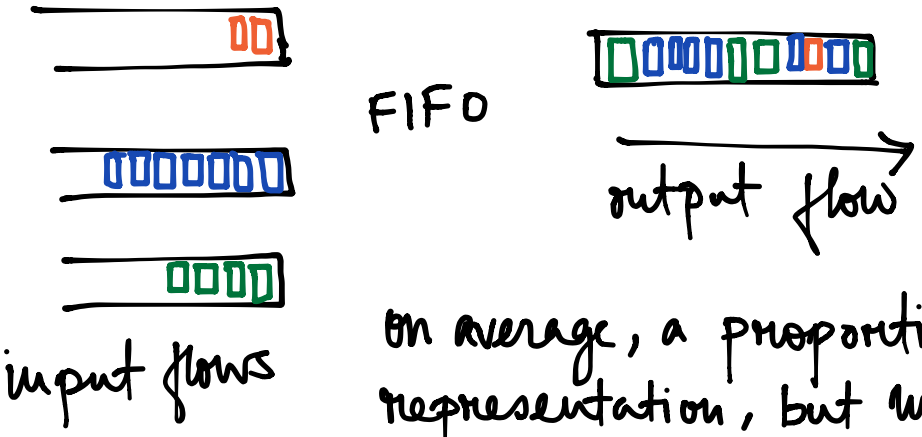
Provides a fair quality of service (QoS)

Sharing bandwidth between flows

- Weighted Fair Queuing

FIFO:

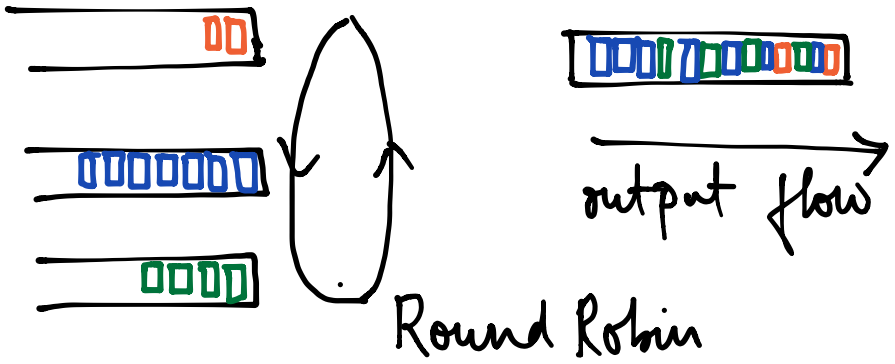
- queue packets are stored and served in a FCFS basis.
- discard packets when queue is full.



on average, a proportional representation, but may have a high variance

- difficult to maintain QoS
- malicious senders can block other flows

Round Robin Queuing



long-term average is still proportional, but each flow can guarantee QoS.

The output order is no longer the input order

– No single flow can throw out others

Problem: different packets might be of different size. Fairness should consider that too.

Solution: provide a bit level fairness

– Weight each of the queues s.t. the net bit flow is equal.

1 unit per packet → 6 packets
2 units per packet → 3 packets
6 units per packet → 1 packet

} equal bits.

Implementations vary - but the idea is same

Weighted fair Queuing is a generalization of the above approach.

Instead of the bit rate of each flow being equal, if we want to make it proportional to some pre-defined ratio

E.g. if □ □ □ = 2 : 1 : 3 in the previous example

Sent = 12 : 3 : 3

= 12 units : 6 units : 18 units

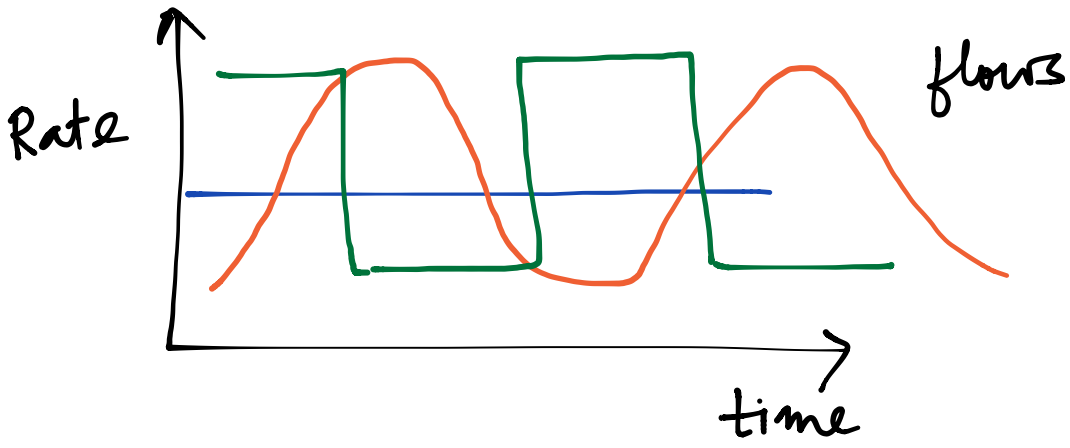
= 2 : 1 : 3

Downside : difficult to implement as is

Traffic Shaping

Real traffic is bursty, so the ratio of the different flows can't be determined beforehand. The averaging has to be done on the fly.

— depends on the app, device, and stochastic network delays



- the pattern is not regular
- not known beforehand.

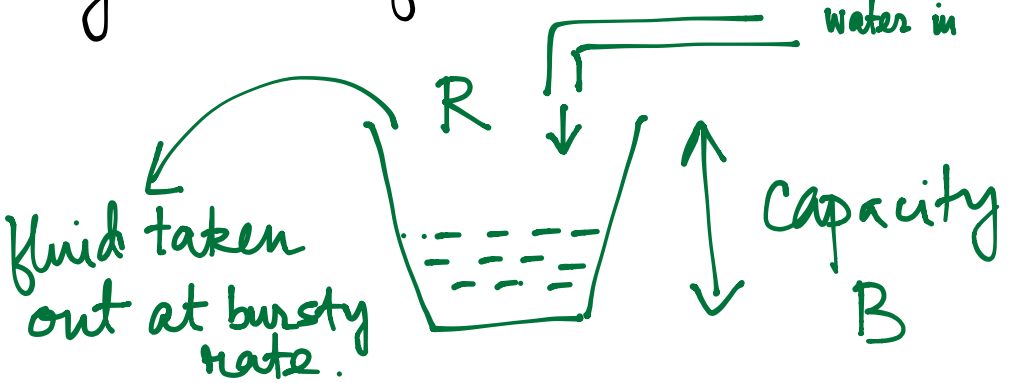
Token Bucket

traffic shaper with two parameters

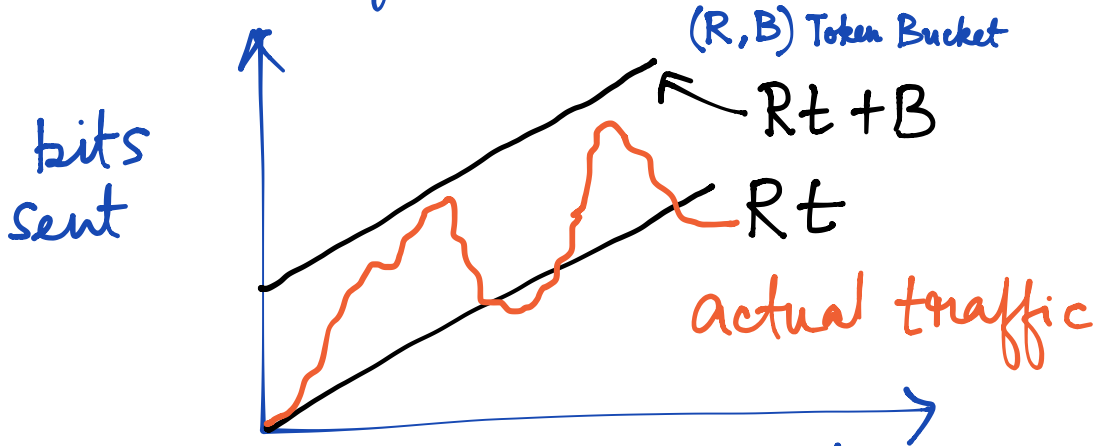
(R, B)

R: average rate of R bits/sec

B: bursts of B bits over R



Time diagram



Token Bucket is a specification for limit

Shaping

Modifying the traffic near the source to fit within (R, B) specifications

- Run Token Bucket (R, B) at source
- Pass the tokens to the network when there are tokens

automatically ensures that never more than $Rt + B$

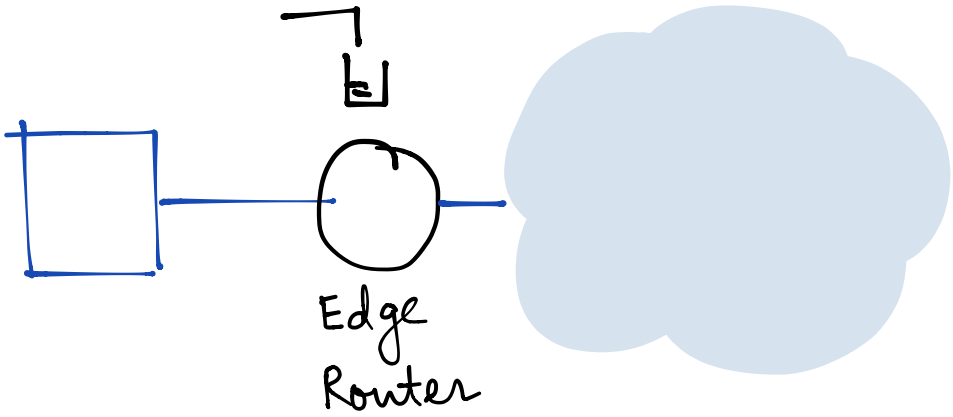
Running TB is a contract between the user and the provider.

Policing

Polices (verifies) if the traffic profile meets the (R, B) tb standard

Happens at the network

- Run (R, B) token bucket at the network edge
- Let packets into the network as long as there are tokens
- discard packets otherwise



Token bucket endnotes

- TB helps regulate the traffic
- Users have the flexibility of choosing their TB (R, B).

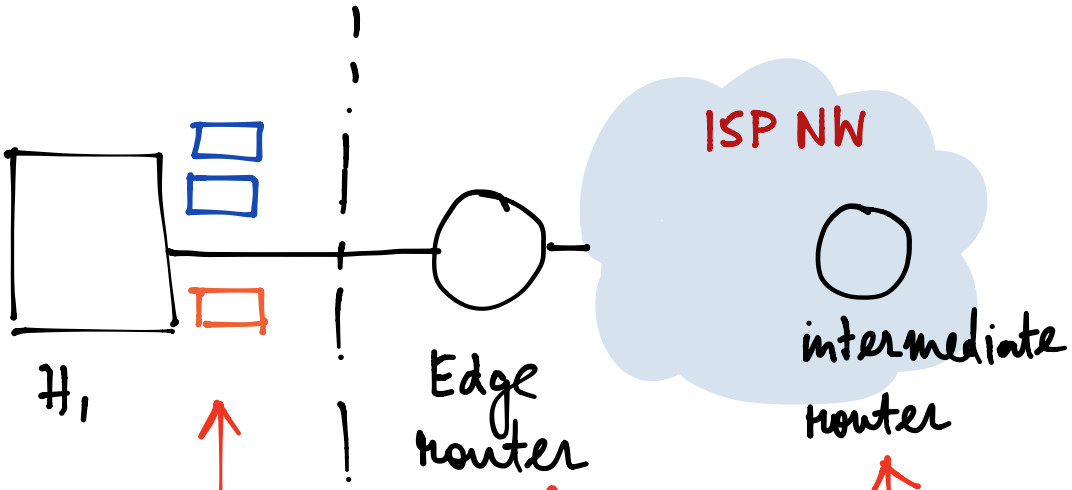
TB limits the flow, preferential treatment is given via methods like WFQ

Differentiated Services

Treating different flows differently in the network.

priorities can be provided by the ISP on a payment basis

Architecture of differentiated service



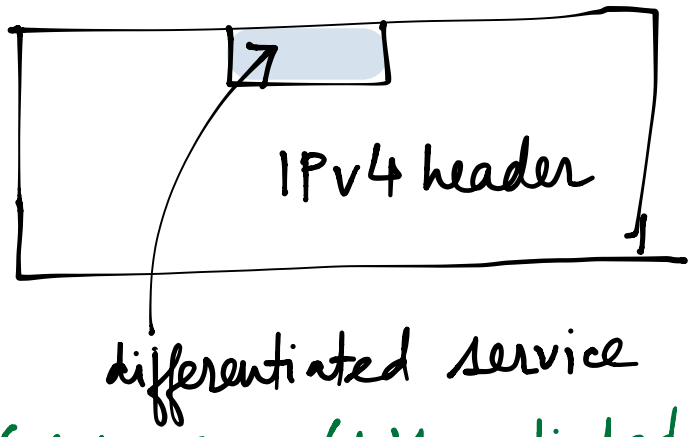
(1) user marks packet for desired service

(2) polices the packet for QoS (via TB)

(3) Provides priority accordingly (via WFQ)

① Packet marking

Use bits in IPv4/IPv6 header



6-bit DSCP (differentiated services code point)

Service / meaning	DSCP	Traffic need
default forwarding/ best effort	0	elastic (P2P)
assured forwarding/ enhanced effort	10-38	average rate (streaming video)
expedited forwarding/ real time	46	low loss/delay (VOIP, gaming)
precedence/ network control	48	high priority (Routing Protocol)

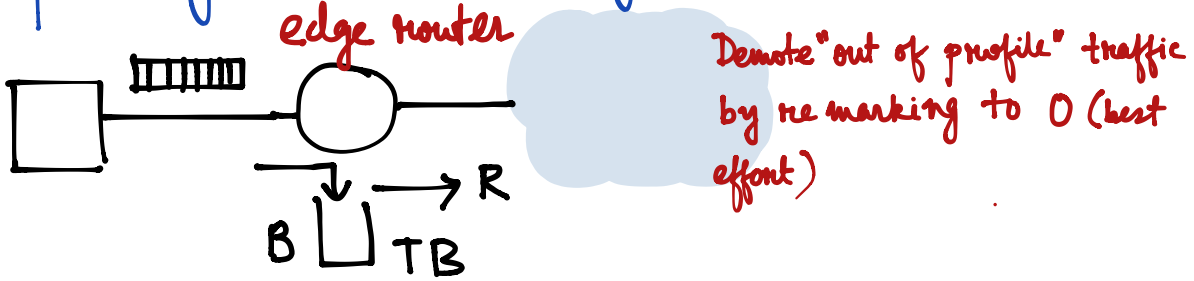
- Marking packets is done by user
- depends on network policies, e.g., VOIP/gaming - expedited?
- The App can set it - can be overwritten by the OS or router/middleboxes
- use heuristics, e.g., port #

---

(2) Policing packets

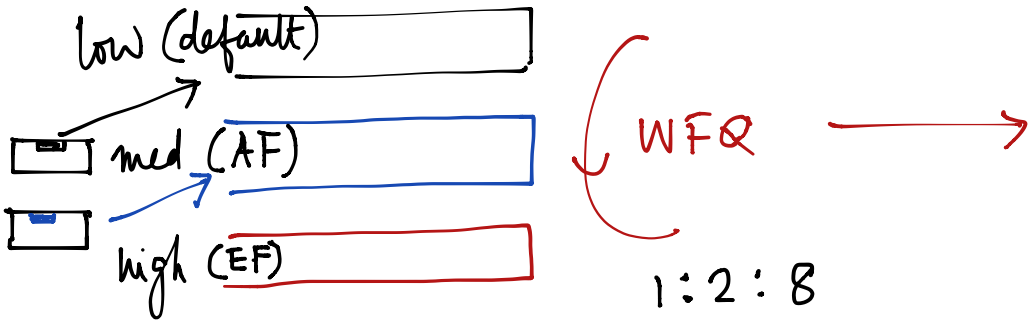
ISP checks if incoming traffic meets service contract

- not more expedited traffic than agreed
- only allowed markings (e.g., no network control traffic - supposed to be internal, router in ISP designs the routing protocol)
- policing is done using token bucket



③ Forwarding Packets

- ISP use WFQ instead of FIFO
- different incoming flows (queues) correspond to the kinds of service
- DSCP values are used to map packets to the right queue



- the queue sizes for the priority classes should be reasonable – otherwise it can block all other traffic

Challenges

- QoS is an end-to-end guarantee
  - no benefit until both parties upgrade
- tied to pricing – to take care of priority overloading

Rate and Delay guarantees

So far, QoS is local (to the ISP)

- want a guarantee across the network
- "hard QoS" guarantee
- e.g., like a circuit switched network [telephone call]

Aside: TCP's connection establishment doesn't quite do this – it is between the end parties but the transmission path is not reserved
- packet losses still happen

Can we do this at all?

difficulties:

- can setup a circuit – expensive
- can we do statistical multiplexing and implement hard guarantees?

We will discuss some approaches to do this

Admission Control

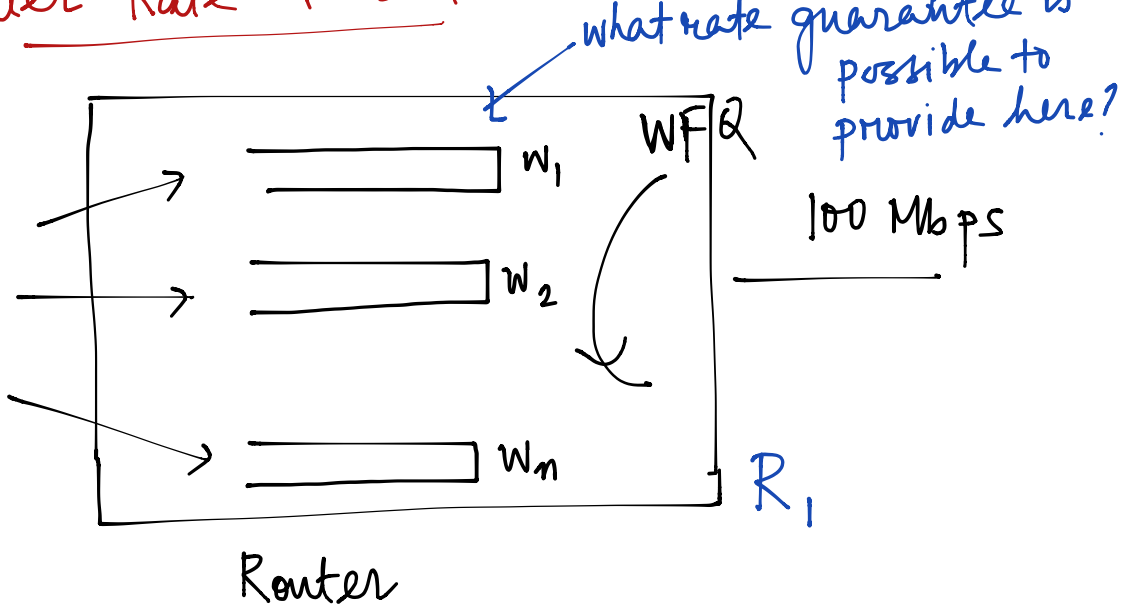
Requirement: a flow needs rate at least R Mbps and delay $\leq D$ msec

decision: to admit or not? [phone call]

rejecting should be infrequent

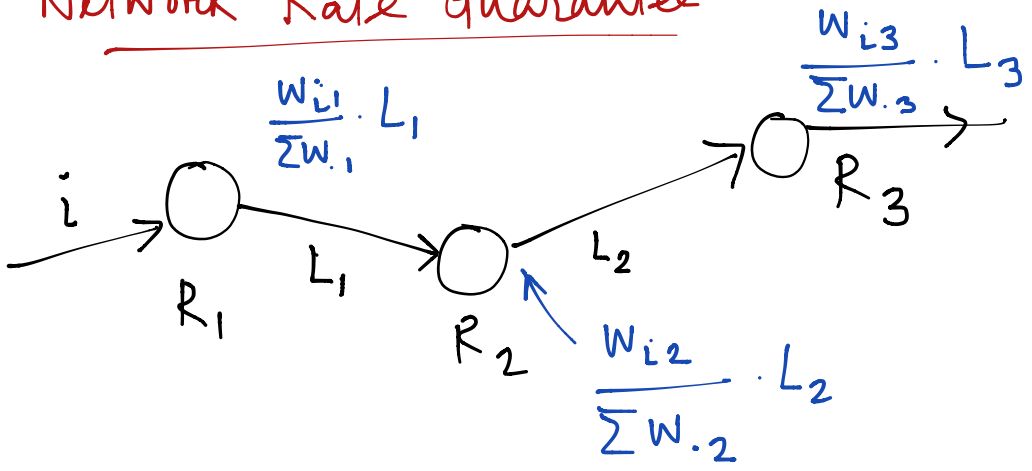
need control over admission of users to the network such that these guarantees are possible to provide.

Router Rate Guarantee



Easy to see that flow i gets $\frac{w_i}{\sum_{k=1}^{n} w_k} \times 100$ Mbps.

Network Rate Guarantee



The net rate is the min of all these rates

- to guarantee R Mbps we need

$$\min_{x \in \text{Routing path}} \frac{W_{ix}}{\sum W_{\cdot x}} \cdot L_x \geq R$$

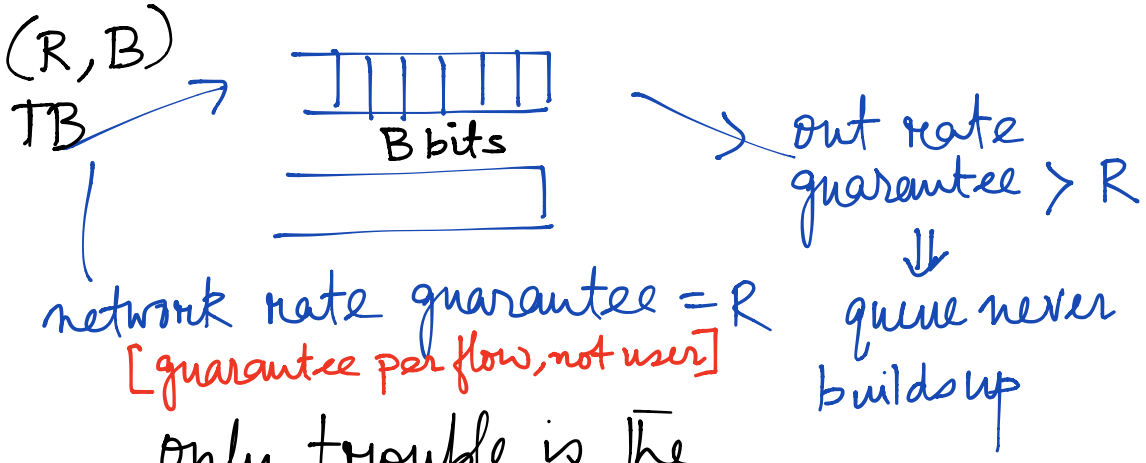
Equality has technical challenge

Delay Guarantee

- Queuing delay — transmission and propagation delays are unsurmountable
- depends on the traffic flow — if other flows get more than their minimum rate, queue

grows and delay grows

— need a max rate of all flows
  — traffic shaping (token bucket)



only trouble is the burst — due to TB, burst can be at most B

— this is consumed by the network min rate

$$\min_{x \in \text{Routing path}} \frac{W_{ix}}{\sum W_{\cdot x}} \cdot L_x = S > R$$

— hence max delay is $\leq \frac{B}{S} < \frac{B}{R}$

Putting things together

Given: A network with

1. (R, B) TB shaped traffic flow
2. WFQ routers with proper weights
3. Sharing of output link via statistical multiplexing

Guarantee:

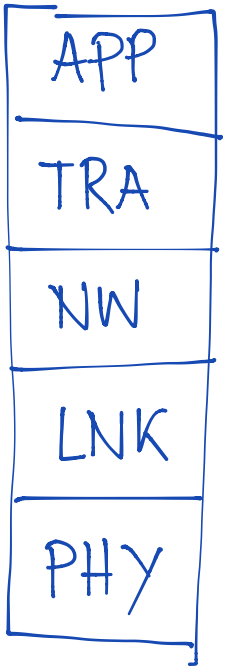
1. min rate = R Mbps
2. max delay = $(t_x + \text{prop delay}) + \frac{B}{R}$

irrespective of how other flows behave
(worst case guarantee)

Summary of our learning

— A lot! can't put in a minute

— highlights



- lot detailed mechanisms drive the smooth transfer of traffic (email, whatsapp, P2P, or Web) on the Internet
- layered structure is to help divide (distribute) tasks but the solution has to involve all

Acknowledgements

- the course TAs
- Notability App on iPad
- Xournal
- Kazam screencast } linux

] Videolecture special

Stay safe & hydrated