- Textbook: An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network, by S. Keshav, 1997, Addison-Wesley.
    - Useful chapters: Ch 5 layering, Ch6 System Design, Ch 9 Scheduling, Ch 11 Routing, Ch 13 Flow control, Ch 14 Traffic management [One-to-one mapping with Lecture slides].

Routing

Packet size

- min: to detect a collision (i.e. duration of transmission from two extreme stations at max length of cable extent, must overlap by enough bits to detect collision and therefore trigger CD part of CSMA/CD)
- max: to prevent *capture* and to make sure delay before eventual access is not too high.

Inter-domain

BGP (Border Gateway Protocol)

Intra-domain

Fibbing paper

Multi-protocol label switching

Segment

Multicast

- hard in BGP
- multicast in mobile routing

Mobile and telephone

Mobile

Telephone circuits: dynamic alternative routing

Flow / congestion control

open-loop vs closed-loop control

- window-based vs rate-based

TCP, packet loss (congestion vs mobile host moving)

- ECN
- AIMD

- RTO (Retransmission TimeOut), congestion window
- Go-Back-N
- buffering, fast retransmit, fast recovery; wireless

Scheduling

The scheduler allocates the delay, bandwidth and loss rate to the packets in the output queues.

work-conserving: whenever there is a packet to send, the scheduler will send it.

- the sum of mean queuing delay q [seconds] to clear all the packets weighted by link load utilization ρ is *independent* of the scheduling algorithm,

$$\sum_{i=1}^{n} \rho_i q_i = C,$$

- where mean link utilization $\rho_i = \frac{\lambda_i}{p_i}$, i.e. the ratio between the mean arrival pkt rate λ_i and the mean pkt service rate p_i [packets/s].
- Alternatively, $\mu_i = \frac{1}{p_i}$ is the mean per-packet service time [s/packet].

non-work-conserving: the scheduler may idle even if there are packets to send.

- allows smoothing of packet flows, or less jitter; downstream traffic more predictable since the output flow is controlled, i.e. less bursty traffic; less buffer space required.
- the sum > C, causing higher end-to-end delay; complex time synchronization required.

Fairness

max-min fairness is important for elastic traffic (best-effort service) since there's no economic incentive for routers to exceed the minimum bandwidth required. Whereas for inelastic traffic, which pays the network operator for a guaranteed bandwidth, fairness is not a concern.

Each flow is assigned m_i resource, which neither exceeds its demand x_i, nor exceeds an equal share of the remaining capacity (for unsatisfied flows), in the order of increasing flow demand,

$$m_0 = 0, m_i = \min(x_i, \frac{C - \sum_{j=0}^{i-1} m_j}{n - (i-1)}).$$

The max-min fair share allocates the small demanding flows what they want, and then evenly distributes the remaining capacity among the larger demanding flows. The process is repeated until all flows are satisfied.

Protection

misbehavior by one connection should not affect the performance received by other connections (or flows), where misbehavior includes sending packets at a rate faster than its fair share.

relationship: fair $\rightarrow$ protection, i.e., a fair scheduler automatically provides protection. However, the converse need not be true.

- because the fairness limits a misbehaving connection to its fair share. On the other hand, if connections are policed at the entrance to the network (conforming to a predeclared traffic pattern) they are protected from each other, but their resource shares may not be fair.

Scheduling algorithms

- FIFO $O(1)$
  - only per packet, not per flow (`src`/`dst` IP/port, protocol).
  - ✘ no fairness and protection.
  - ✓ easy to implement, and no extra state stored in RAM.
- Priority queueing
  - ✓ guaranteed-service.
  - ✘ starve low-priority flows, unless proper admission control and policing are used.
- Generalized Processor Sharing (GPS)
  - "bitwise round-robin" benchmark (Flow A: $g(A)$ = 1 bit $\rightarrow$ Flow B: 1 bit $\rightarrow$ ... $\rightarrow$ Flow A: 1 bit $\rightarrow$ ...).
  - For N connections with positive real weights ϕ_i, and
  - $S(i, t_1, t_2)$ is the amount of service received by connection i in the interval $[t_1, t_2]$,
  - for any connection i with data in queue, for any connection j,

$$\frac{S(i, t_1, t_2)}{S(j, t_1, t_2)} \geq \frac{\phi_i}{\phi_j}.$$

  - ✓ fair as each flow shares the remaining bandwidth in proportion to its weight.
  - ✘ not implementable since packets are not infinitesimal small, but is used as a reference.
    - Relative Fairness Bound: $RFB = |\frac{S(i,t_1,t_2)}{g(i)} - \frac{S(j,t_1,t_2)}{g(j)}|$; Absolute Fairness Bound: $AFB = |\frac{S(i,t_1,t_2)}{g(i)} - \frac{GPS(i,t_1,t_2)}{g(i)}|$,
    - where $g(i) = \min_k g(i,k)$ and the service rate in switch k is $r(k)$,

$$g(i,k) = \frac{\omega(i,k)}{\sum_k \omega(j,k)} r(k).$$

- Weighted Round Robin (WRR) $O(1)$
  - use case: high-speed ATM networks, with fixed-size packets and short round times.
  - normalized weight $\hat{\omega} = \frac{\omega}{\mu}$, where μ is the mean packet size of the flow.
    - ✘ must know mean packet size in advance.
    - ✘ huge state stored in RAM; new/short-lived flows need default average.
  - □ fair only over time scales longer than a round time.
    - At a shorter time scale, some connections may get more service than others (100%, ..., 0%, 0%).
    - If a connection has a small weight, or the number of connections is large, this may lead to long periods of unfairness.
- Deficit Round Robin (DRR) $O(1)$

    - associate each flow with a deficit counter (dc), initialized to 0.
    - for non-empty queue, packet at the head is
        - served and `dc = dc + quantum - pkt_size`, if `pkt_size` $\leq$ `dc + quantum`;
        - kept in the queue and serve one quantum `dc += quantum`, otherwise. [build up credit]
        - quantum choice: at least the largest packet size of the flow (work-conserving).
    - reset `dc` to 0 when the queue is empty.
    - ✓ no need to know the mean packet size in advance, hence no problem for new/short-lived flows.
    - ✘ huge state stored in RAM for `dc` for each flow.
    - □ fair only over time scales longer than a frame time.
    - use case: fair longer, given small packet size and small number of flows.

- Weighted Fair Queuing (WFQ) $O(\log n)$
    - smallest finish-number $F(i,k,t)$ from GPS across queues served first,
    - where flow i with weight ω_i, packet k at time t with size $P(i,k,t)$,
    - the length of a round, that is, the time taken to serve one bit from each active connection, is proportional to the number of active connections.
    - define the round number $R(t)$ to be the number of rounds that service GPS has completed.
        - increases at a rate inversely proportional to the number of active connections.

$$
\begin{aligned}
F(i,k,t) &= \begin{cases} R(t) + \frac{P(i,k,t)}{\omega_i}, & \text{if connection } i \text{ is inactive} \\ F(i,k-1,f) + \frac{P(i,k,t)}{\omega_i}, & \text{if connection } i \text{ is active} \end{cases} \\
&= \max\{F(i,k-1,f), R(t)\} + \frac{P(i,k,t)}{\omega_i}.
\end{aligned}
$$

    - ✓ no need to know the mean packet size in advance.
    - ✘ storage for finish numbers $F(i,k,t)$ for each flow
    - ✘ processing time to find the smallest finish number.

algo.	work-conserving?	max-min fair?	protection?	w/o mean pkg size?
FIFO/FCFS	✓	✘	✘	✓
Priority	✓	✘	✘ starve	✓
GPS	✓	✓	✓	✓
WRR	✓	long time scale ✓	long time scale ✓	✘
Deficit RR	✓	medium and	long time scales ✓	✓
WFQ	✓	✓	✓	✓

Queue management

drop from T/H, smart/random drop, flush, Active Queue Management (e.g. Random Early Detection), ECN.

- y2012p8q10

Data centers

QJump, a switch priority queueing, combined with end systems traffic regulation, by exploiting its regular topology & traffic matrix (src-dst) & source behaviour.

Data center, QJump paper.

Optimization

In general, we have two coupled problems, i.e.,

- Traffic engineering (routing) problem: given fixed rate $\{x_i\}$, minimize the network congestion with variable routing path R_{li}, i.e., the fraction of flow i on link l,

$$\min_{R} \quad \sum_{l \in L} D(u_l), \qquad \text{where } u_l = \frac{\sum_i R_{li} x_i}{c_l}. \tag{routing}$$

- Congestion control (system) problem: given fixed path R, maximize the utility of the system with adaptive source rates $\{x_s\}$,

$$\max_{x_s} \quad \sum_{s \in S} U_s(x_s), \qquad \text{s.t.} \sum_{s \in l} x_s \le C_l, \forall l \in L. \tag{system}$$

Routing

- W: set of src-dst pairs.
- r_w is the fixed rate of sd pair w [bps].
- P_w is the set of path between the sd pair w.
- x_p is the pkt rate on path p [bps].
- C_l is the capacity of link l [bps].
- $F_l = \sum_{\text{all } p \text{ with } l} x_p$.

Given (routing) on link $l = ij$, it's equivalent to the following,

$$\sum_{\text{all links } l} D_l(\sum_{\text{all} p \text{ with } l}), \qquad \text{s.t.} \sum_{p \in P_w} x_p = r_w, \forall w \in W; x_p \ge 0,$$

where D_l is the congestion function of link l, which is a non-decreasing function of the total flow F_l through it,

$$D_l(F_l) = \frac{F_l}{C_l - F_l},$$

- as F_l approaches the capacity C_l of the link, the congestion function D_l approaches infinity.

For each sd pair w and each k_w paths, evaluate marginal utilities $\frac{\partial D(x)}{\partial p_i}$ equal, where $i \in [1, ..., k_w]$.

Congestion control

Proportional fairness

For adaptive-rate flows, the vector of rates X_s is proportional fair if for any other vector of rates X'_s,

$$\sum_{s \in S} \frac{x'_s - x_s}{x_s} \leq 0.$$

the sum of relative improvements possible for each individual flow is non-positive.

- It keeps a balance between the utilitarian $\sum_{s \in S}(x'_s - x_s) \leq 0$ (efficiency) and egalitarian $(\min_{s \in S} x'_s - \min_{s \in S} x_s) \leq 0$ (fairness).
- When the demand exceeds the capacity, high demanding flows are flavored due to a smaller proportional loss.

(A) For the network problem, we are trying to maximize the revenue, which is in diminishing returns w.r.t the rate x_s,

$$\max_{x_s \geq 0} \quad \omega_s \log x_s, \quad \text{s.t.} \sum_{s \in l} x_s \leq C_l, \qquad (\text{network})$$

is the proportional fairness of rates x_s per unit charge, via aggregated rate x_r by replacing single user with w_r users.

- The more source pays ω_s [$/s] (willingness-to-pay), the more bandwidth x_s [bps] is allocated to it.
- ω_s [$/s] $= p_s$ [$/bit] $\cdot\, x_s$ [bits/s].

(B) For the user problem, we are trying to maximize the user's utility, while decreasing the cost involved, for each source s,

$$\max \quad U_s(x_s = \frac{p_s}{\omega_s}) - \omega_s, \quad \text{s.t. } \omega_s \geq 0. \qquad (\text{user})$$

Together with (A) and (B), we are solving the intractable system problem, since the source utility functions are not known by the network in advance,

$$\max \quad \sum_{s \in S} U_s(x_s), \quad \text{s.t.} \sum_{s \in l} x_s \leq C_l, \forall l \in L. \qquad (\text{system})$$

Under the decomposition of (system) problem into (network) and (user), we derive $x_r = \frac{\omega_r}{\sum_{j \in r} p_j}$ from the (network) Lagrangian.

- The Lagrange multipliers $\mu = p_j$ are the shadow prices of all the sources s that use it. (See Optimization paper for more details.)

The dual problem is reduced to differential equation, the change in bandwidth allocation at source s includes linear increase of ω_s and the multiplicative decrease based on the shadow price,

$$\frac{d}{dt} x_s(t) = k[\omega_s - x_s(t) \sum_{l \in L(s)} p_l(t)],$$

where shadow prices (congestion signals) $p_l(t) = g_l[\sum_s x_s(t)]$ is accumulated along the path $L(s)$ from the source s to the destination,

- g_l is the price charged by link l per unit flow through it [$/bps].
- The total flow through link l is $\sum_{l \in L(s)} x_s(t)$, i.e. the sum of the flows from all sources s passing through link l.
- It's a distance-related (RTT dependence) cost.

Traffic management

QoS + Connectivity.

Elastic vs inelastic

Traffic classes and synchrony (see Flow control section)

- Guaranteed-service, or inelastic (delay sensitive and unusable for low bandwidth, i.e. utility is 0)
    - Synchronous, or real-time, e.g. gaming, video conferencing.
    - open-loop flow control with admission control (RTP over UDP).
- Best-effort, or elastic (benefit from throughput, but not delay-sensitive)
    - Asynchronous, or non-real-time, e.g. file transfer, email, web browsing.
    - closed-loop flow control with congestion control (TCP), with rate adaptive to network conditions.

Utility function, Traffic model (measure or math), time scale, renegotiation, admission control, capacity planning and macroscopic QoS.

Peak load pricing

Signalling

Related: multicast (RSVP : Resource reSerVation Protocol)

IntServ vs. DiffServ

System design

layering, randomness and hysteresis

Seven Layering