Fair Queueing: Demers, Keshav, Shenker [Demers89a]

CSci551: Computer Networks
SP2006 Thursday Section
John Heidemann

Key ideas

- want to have smart routers to control congestion
- routers can enforce fair allocation of bandwidth
- why?
  - can prevent malicious users from hurting others
  - can keep bulk traffic sources from unnecessarily delaying non-bulk traffic

Places to Implement Congestion Control

- applications (admission control)
  - ex: rate-adaptive applications (not admission control), telephone network ("all circuits are busy"), sometimes web sites show "busy signal" (503 unavailable?)
  - advantages: pre-emptive, maybe easier (just once per connection, not ongoing)
- source and destination (transport, end-to-end)
  - ex: TCP, DECbit
  - advantages: dumb (cheaper), faster? routers, easy to change end points—maybe more flexible
- routers ("gateways")
  - via adaptive routing (ex: load-sensitive routing)
    - advantages: xxx
    - disads: xxx
  - via queueing policy (ex: NOT FCFC (FIFO, drop tail), FQ Fair Queueing)
    - advantages: rtr has more information, and info about all the flows (if network is shared), can enforce fairness, deal with malicious users

Queueing Policies

- Many policies have been considered:
  - FIFO ("drop tail") [and also drop head]
  - round robin (per flow)
  - weighted round robin
  - fair queueing [this one]
  - token bucket
  - virtual clock
  - class-based queueing (per class of traffic)
  - stochastic fair queueing (statistical)

Alternatives

- how quickly do you provide feedback
- what kind of fairness do you provide
  - fair to whom: flows, src-dest pairs
    - be careful: if you have more flows, you get more bandwidth
    - although in principle you could change the definintion of who
  - how fair: min-max fair, guarantee
- how efficient you are (router go idle?)
  - ex. circuit switched networks doing TMDA let the router go idle
  - but FQ wants to keep rtr busy
- how much state you must keep
  - information per flow
  - *per flow state* is a big concern in big routers
- how do you signal congestion
  - dropping vs. explicit feedback (DECbit, ECN)

Fair Queueing

- fairness
  - to whom: *conversations* (now called *flows*: src-dest addr-port pairs)
  - quality: always (not just statistical)
  - definition: max-min
- state
  - $R(t)$—cumulative number of rounds serviced from time 0 to time t
  - S_i and F_i—start and finish rounds for flow i

Max-Min Fairness

- definition:
  1. no user gets more than request
  2. no better allocation
  3. "condition 2 is recrusively true if we remove the minimal user and reduce the total resource accordingly"
- another view:
  - $\mu_i = MIN(\mu_{fair}, \rho_i)$
- considering allocation
  - find the minimal allocation
  - give everyone that much if undercapacity
  - if over capacity, divide what's left equally
  - if capacity left over, repeat with anyone left who wants more

Examples of Min-Max Fairness

- offered load => accepted load (dropped)
- 1/3, 1/3, 1/3 => 1/3, 1/3, 1/3 (0, 0, 0)
- 1, 1, 1 => 1/3, 1/3, 1/3 (2/3, 2/3, 2/3)
- 1/3, 1/6, 1/6 => 1/3, 1/6, 1/6 (0, 0, 0)
- 2/3, 1/6, 1/6 => 2/3, 1/6, 1/6 (0, 0, 0)
- 1, 1/6, 1/6 => 2/3, 1/6, 1/6 (1/3, 0, 0)
- 1, 1, 1/6 => 5/12, 5/12, 1/6 (xxx)
- 1, 2/3, 1/6 => 5/12, 5/12, 1/6 (xxx)
- what about bad users? always request 1, always get 1/N
  - why bother to under request? hopfully people will ask for only what they need, OR/AND you could do billing AND this FQ paper talks about giving time bonuses to people who don't use their full share

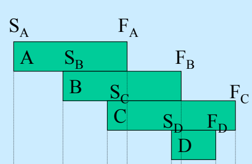
Developing Fair Queueing

- start with *bit-wise fair queueing*
  - not realizable in practice, but allows strong definitions of fairness
- generalize to *packet-level fair queueing*
  - same arguments for correctness apply, but can now be realized

Term Definitions

- R(t): cumulative number of rounds through t
  - a round is sending one bit of each flow
- dR(t)/dt: rate you service rounds
- S_i, F_i: start and end times of a sender pkt i
- to allow *promptness*
  - B_i: bid (request for bonus if you haven't sent much)

Bit-wise Fair Queueing

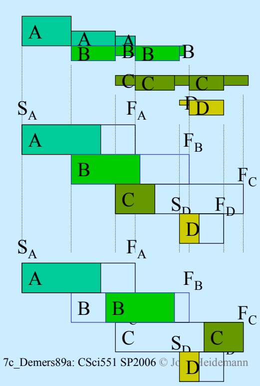


$N_{ac}(t)$:0 1 2 3 2 3 2 1 number of active flows

dR/dt:0 1/1 ½ 1/3 ½ 1/3 ½ 1

R(t) :0 1 2 3 4 5 6 6.5 7 7.5 8 8.33 .....

Bit-wise Fair Queueing



redrawn so that area corresponds to number of bits

bit-level scheduling (not realizable)

packet-level scheduling based on arrival times (not realizable)

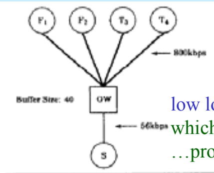
one realizable pkt-level schedule

Modeling Fair Queueing

- Analytic modeling (section 2.3)
  - evaluate mean queueing delay
- Simulation modeling
  - compare a variety of "flow control" alternatives: window, TCP, DECbit

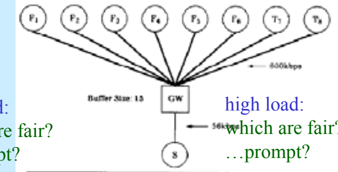
Sample of Sim Results

[Demers89a, scenarios 1 & 2]



low load:
which are fair?
…prompt?

Quantity	Policy	FTP 1	FTP 2	Telnet 3	Telnet 4
Throughput (packets)	G/FCFS	1746	1746	99	96
	G/FQ	1746	1746	102	94
	JK/FCFS	1747	1745	102	104
	JK/FQ	1746	1746	105	103
	DEC/DEC	1745	1746	131	106
	DEC/FQbit	1746	1746	102	94
Average Roundtrip Time	G/FCFS	1.43	1.43	1.36	1.35
	G/FQ	1.43	1.43	.079	.080
	JK/FCFS	1.43	1.43	1.38	1.38
	JK/FQ	1.43	1.43	.084	.088
	DEC/DEC	.297	.299	.215	.214
	DEC/FQbit	1.43	1.43	.080	.090



high load:
which are fair?
…prompt?

Quantity	Policy	FTP 1	FTP 2	FTP 3	FTP 4	FTP 5	FTP 6	Telnet 7	Telnet 8
Throughput (packets)	G/FCFS	18	1154	1169	3	1149	15	31	3
	G/FQ	178	830	591	600	615	421	96	98
	JK/FCFS	583	583	586	566	582	587	3	0
	JK/FQ	374	579	546	594	599	601	87	96
	DEC/DEC	582	582	582	582	582	582	99	90
	DEC/FQbit	582	582	587	587	587	582	105	97
Average Roundtrip Time	G/FCFS	403	2.18	2.16	-	2.18	149	11.5	-
	G/FQ	16.8	3.31	4.88	4.83	4.83	4.47	.079	.078
	JK/FCFS	1.85	1.93	1.93	1.85	1.93	1.85	-	-
	JK/FQ	1.75	1.78	1.13	1.80	2.20	2.16	.091	.085
	DEC/DEC	.859	.859	.859	.859	.859	.859	.778	.782
	DEC/FQbit	.860	.860	.860	.860	.860	.860	.089	.082

Limitations Fair Queueing

- problem of per-flow state
  - Internet backbone routers have many, many flows per second => can't keep per flow state
- recent approachs
  - police at *edges* of network (*core* is dumb but fast)
    - "core stateless" fair queueing [Stoica et al]
  - or can we keep *less than per-flow state*
    - maybe only state about "big" flows
    - active area of research

Other questions/observations?

- xxx