

1 ☐ EECS 122, Lecture 27

Today's Topics:

Scheduling Best-Effort and Guaranteed Connections

QoS in ATM

Kevin Fall, kfall@cs.berkeley.edu

2 ☐ Where we are...

- The motivation for QoS
 - a desire to provide a better than best effort service
 - motivation by ISPs to charge for new services
- Need scheduling and traffic regulation
 - guaranteed and best-effort handled somewhat differently
- Haven't talked about scheduling details...

3 ☐ Scheduling of Best-Effort Traffic

- Ideal work-conserving scheduling discipline that achieves max-min fairness is called Generalized Processor Sharing (GPS)
- GPS serves an infinitesimally small amount of data from each queue in round-robin fashion
- GPS is not implementable, but achieves exact weighted max-min fairness, so usually compare real approaches to GPS

4 ☐ Max-Min Fairness using GPS

- A GPS server, m , that serves N sessions on a link is characterized by N positive real numbers
These numbers denote the relative amount of

service to each session. More precisely, if $S_{i,m}$ is the amount of session i traffic served by server m during an interval $[s, t]$, then

- (for session i continually backlogged)

5 ☐ What this means...

- So, for non-backlogged connections, they already receive what they ask for. The GPS server ensures that backlogged connections share the remaining bandwidth in proportion to the assigned weights (i.e. max-min fairness).
- For every backlogged connection i , each receives a service rate at switch m of:

6 ☐ Weighted Round Robin

- WRR approximates GPS reasonably well for connections with equal packet sizes
 - if different size packets, need mean packet size to be close to GPS [may be hard to know]; if not known, RR isn't fair
 - fair only over time scales longer than a round time; with large # connections or small weight, may be unfair over long periods
- Note WRR would be ok in ATM...

7 ☐ Deficit Round Robin (DRR)

- Modification to WRR so knowing mean packet size is not required
- Choose a quantum of bits to serve from each connection in order. For each HOL packet, if its size is \leq (quantum+credit) send and save excess, otherwise save entire quantum. If no packet to

send, reset counter (to remain fair)

- Easier implementation than WFQ

8 ☐ DRR Example

- Each connection has a deficit counter (to store credits) with initial value zero. Scheduler is config'd with quantum.
 - Assume quantum=1000, 3 connections (A-C) have packets of size 1500, 800, 1200
 - Round 1: A's counter goes to 1000, B's first packet is served (and its counter goes to 200), C's counter goes to 1000
 - Round 2: A's packet is served (and its counter goes to 500), C's packet is served (and its counter goes to 800), B's counter reset to zero

9 ☐ PGPS (Pkt-by-pkt GPS) and WFQ

- WFQ and PGPS independently discovered disciplines which do not require GPS's infinitesimal service assumption
- we have studied this already, and know it approximates max-min fairness
- If we combine with regulation/policing, we can do some interesting things...

10 ☐ Guaranteed Service Connections

- It turns out you can provide bandwidth and delay bounds using WFQ (wow!)
- Assuming a leaky-bucket constrained source i , assume its service rate is:
- Result (end2end trans+queue delay D):

11 Looking a bit closer...

- Result is from Parekh/Gallager, 1992
- Definitions:
 - $P_{\max}(i)$ = largest pkt on connection
 - P_{\max} = largest pkt allowed on network
 - $D^*(i)$ = end-2-end delay on connection i
- With $P_{\max} = 0$ [infinitesimal], gives bound of just $1/g(i)$ [like one scheduler w/rate $g(i)$]

12 An Example

- Assume connection has leaky bucket parameters (16KB, 150Kbps), 10 hops, all link bandwidths 45Mb/s. With largest pkt size of 8KB, what g will guarantee an end-to-end delay of 100ms, assuming a total propagation delay of 30ms?
 - Need max queue delay of $100 - 30 = 70$ ms, so we have

13 Example (cont'd)

- Solve this for g :
 - $0.7 = (16K \cdot 8)/g + (10-1) \cdot 8K \cdot 8/g + 10 \cdot 8K \cdot 8/45 \cdot 10^6$
 - g is about 13Mb/s [$>85x$ source avg rate!]
 - large packets can lead to substantial delays

14 Virtual Clock

- Similar to WFQ (scheduler stamps packets with finish time tags and services them in order of tags)
- Instead of GPS, emulates TDM; easier to compute finish number than WFQ

- If all connections are backlogged, WFQ and VC behave identically (ok for guar.)
- Read about it in Chapter 8...

15 ☐ Delay-Earliest Due Date (EDD)

- Delay-EDD
 - assign scheduling deadlines so that even with all connections at peak rate, worst-case delay in traffic descriptor is met
 - bandwidth reservation independent of delay bound, but must use peak rate regulator thereby giving up stat muxing gain
 - deadline is time at which it should be sent had it been received according to traffic contract (slower than peak rate)

16 ☐ Jitter-EDD

- Delay-jitter regulator precedes the EDD scheduler
 - packets receive same delay at each hop (except the last one), so total jitter is reduced to that of last hop
 - can provide end-to-end bw, delay, & jitter bounds
- Incorporates Delay-EDD for delay guarantees, so in that case must reserve at peak rate

17 ☐ Rate Controlled Scheduling

- Provide bw, delay, and jitter bounds
- Two components: regulator & scheduler
 - regulator determines eligibility time for pkts
 - scheduler selects among eligible packets
- By selecting which regulator and scheduler, can implement a wide range of overall service disciplines
 - e.g.: rate-controlled static priority

- rate-jitter regulator & multi-level FCFS prio scheduling

18 ☐ RC Scheduling Implementation Issues

- Implement a regulator
 - if we can tolerate some granularity in time stamps, can use a calendar queue
 - for delay-jitter regulation, also need clock synchronization and timestamps in each pkt
- Implement a scheduler
 - if FCFS or multi-FCFS, just queues
 - if sorted/deadlines, etc need priority queue

19 ☐ Packet Drop Strategies

- Assumption is that guaranteed connections rarely drop any packets (due to admission control), but best-effort flows must deal with this
- Similar characterizations as schedulers:
 - degree of aggregation
 - choice of drop priorities
 - early or overloaded drop
 - drop position

20 ☐ Degree of Aggregation

- Essentially a choice of per-flow or per-class state maintenance
 - per-flow: more protection on overload
 - per-class: less protection, easier to implement
- Can achieve min-max buffer allocation if always drop from the largest queue
- If using WFQ-like scheduler, drop packet with largest finish number, even w/out per-flow queuing

21 ☐ Drop Priorities

- Mark some packets as higher priority
 - on overload, drop lower priority
 - (maybe even do this before overload)
- Loss bits may be set by source or policer (or both)
- What to drop (note cell versus frame in ATM)... several switches uses PPD/EPD [basically doing frame dropping]

22 ☐ Drop Early or on Overload

- Early drop works for responsive sources
 - early random drop and RED
 - ERD not as effective as RED in controlling misbehaving users
- RED substantially improves performance of network of cooperating TCP sources
 - probability of drop is roughly proportional to its throughput share
 - RED has no bias against bursty sources

23 ☐ Drop Position

- Which packet to drop when dropping?
 - head, tail, random, [entire queue]
- Tail drop
 - most straightforward to implement
 - no modification to queue head/tail pointers
- Head drop
 - better for dupack detection (because “hole” will be served earlier; don’t have wait for whole queue)

24 ☐ QoS Summary

- Guaranteed and best-effort service
 - guaranteed is set up by traffic descriptor, should rarely

- or never drop packets
- best-effort may drop packets, but scheduling should be fair
- Leaky-bucket traffic model
- Many scheduling disciplines
 - GPS, WFQ, Delay/Jitter-EDD, etc
- What happens in the real world?

25 ☐ The Real World

- Today, there are really two contenders for supporting QoS: Internet and ATM
- ATM QoS has a several-year jump start, but is considerably more complex (and applies only to ATM, of course)
- Internet QoS is a simpler model, but must apply to all sorts of link technologies, and so poses a significant challenge
- Expect to see Internet QoS on ATM...

26 ☐ QoS in ATM: Service Categories

- Constant Bit Rate (CBR)
 - periodic, constant bit rate, like TDM
- Variable Bit Rate (VBR-rt & VBR-nrt)
 - variable periodic sources (real-time and not)
- Available Bit Rate (ABR)
 - like best-effort, but with flow control
- Unspecified Bit Rate (UBR)
 - completely best-effort

27 ☐ ATM Traffic Parameters

- Peak Cell Rate (PCR):
 - maximum cell transport rate

- Sustainable Cell Rate (SCR)
 - average allowable, long-term transfer rate
- Maximum Burst Size (MBS)
 - maximum back-to-back cell burst size
- Minimum Cell Rate (MCR)
 - minimum cell transport rate

28 ☐ ATM QoS Parameters

- Peak-to-peak Cell Delay Variation (CDV)
 - essentially a jitter measurement
 - bound on this is the CDV “tolerance” (CDVT)
- Maximum Cell Transfer Delay (maxCTD)
 - end-to-end delay bound
- Cell Loss Ratio (CLR)
 - bound on fraction of lost cells

29 ☐ Use of Parameters by Category

- CBR: uses PCR as maximum rate and maxCTD as bound on delay of cells
- VBR-rt: uses PCR, SCR, and MBS; cells delayed above maxCTD not valuable
- VCBR-nrt: uses PCR, SCR, and MBS
- ABR: uses PCR and MCR
 - uses feedback (RM cells) for flow control
- UBR: PCR for information only

30 ☐ QoS Components in ATM

- Usage Parameter Control (UPC)
 - regulator used to set CLP bit on overload
- Traffic Shaping
 - leaky bucket control via GCRA algorithm

- Cell Loss Priority Control (CLP)
 - drop policy based on CLP bits
- Connection Admission Control (CAC)
 - per-VC admission control at call setup
- ABR Flow Control via RM cells

31 ☐ Shaping and UPC

- Traffic shaping and UPC are procedures for regulating and policing traffic at the ATM ingress
- Reference algorithm is called GCRA
 - continuous leaky bucket algorithm
 - defines relationship between PCR and CDVT as well as SCR and BT (burst tolerance, derived from PCR, SCR and MBS)
 - notation GCRA(I,L); I=increment, L=limit

32 ☐ Cell Loss Priority

- CLP bit in ATM header may be set by either end station or by shaping/UPC mechanism
- For ATM networks which are sensitive to it, can be used to direct load shedding on network during congested periods

33 ☐ Connection Admission Control

- No algorithmic in the ATM spec, so this is an area for some innovation by switch manufacturers
- Two main approaches:
 - measurement based
 - analytic, based on assumed statistics and traffic parameters

34 ☐ Admission Control

- Many algorithms in the literature
- One important class uses *equivalent capacity*
- Equivalent capacity is the amount of capacity (bandwidth) required to handle a set of statistically multiplexed flows with a bound on the probability of buffer overrun (oversubscription)

35 Source Models

- Equivalent capacity (and other models) try to estimate statistics of sources by making certain assumptions about their behavior
- One common assumption is that traffic is independent (e.g. independent on/off periods)
- These independence assumptions help to form tractable equations

36 Equivalent Capacity

- We want to find C such that
- So, for assumed two-state sources with exponentially distributed burst and idle periods [memoryless], we can attempt to compute how many flows are likely to be bursting at the same time, but even this is hard, so...
- Assume a Gaussian distribution on the aggregate bit rate (params m, σ)

37 Equivalent Capacity

- With all of these assumptions, we get:

- Upshot: admission control not such an easy problem. Furthermore, there is reason to believe these sorts of assumptions may not be valid...