Admission Control

Outline

- Economic principles
- Traffic classes
- Mechanisms at each time scale
  - Faster than one RTT
  - One RTT
  - Session
    - Signaling
    - Admission control
  - Day
  - Weeks to months
- Some open problems

Connection (flow) Admission Control (CAC)

- Can a call (flow, session) be admitted?
  - $\sum$*(bandwidth allocated for all connections)* $\leq$ *Link Rate*
  - Otherwise the call is inadmissible
  - What bandwidth to allocate to connections??
    - *Depends upon the traffic, traffic model assumed and the Queueing methodology deployed and model used to estimate the required bandwidth*
  - Procedure:
    - Map the traffic descriptors associated with a connection onto a traffic model;
    - Use this traffic model with an appropriate queuing model for each congestion point, to estimate whether there are enough system resources to admit the connection in order to guarantee the QoS at every congestion (or queuing) point.
    - Allocate resources if the connection is accepted.

CAC (continued ..)

- Depending on the traffic models used, the CAC procedures can be too conservative by over allocating the resources.
- This reduces the *statistical gains*

$$StasticalGain = \frac{Number\ Connections\ admitted\ with\ Statistical\ Multiplexing}{Number\ of\ Connections\ admitted\ with\ peak\ rate\ allocation}$$

- An efficient CAC is the one which produces maximum amount of statistical gain at a given congestion point without violating the QoS.
- The efficiency of the CAC thus depends on how closely the two steps (traffic model and queuing model) above model reality.
- Both the traffic and queuing models are well researched and widely published in the literature.

CBR and UBR Admission Control

- CBR admission control (Peak Rate Allocation)
  - simple

$$\sum_i PCR_i \leq Link\,Capacity$$

  - on failure: try again, reroute, or hold
- Best-effort admission control
  - trivial
  - if minimum bandwidth needed, use CBR test

CAC for CBR (with small jitter)

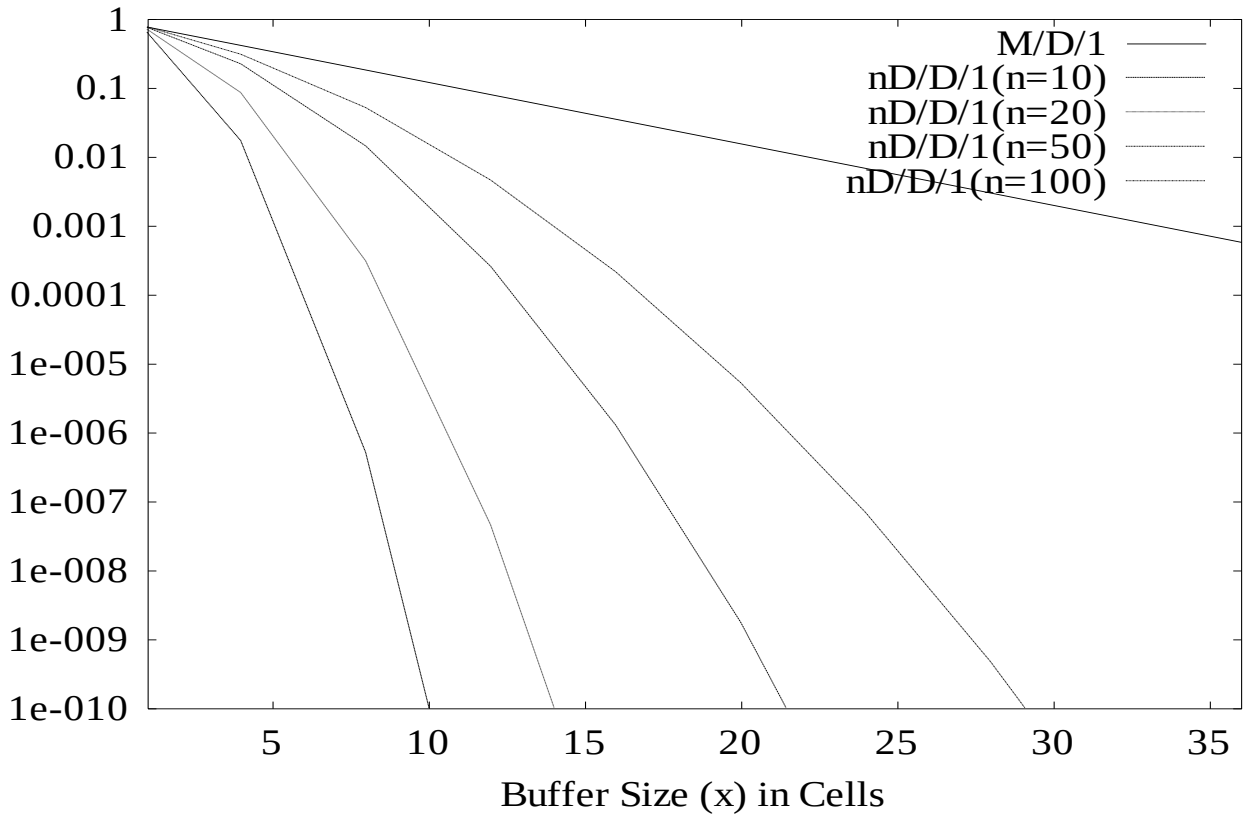
- Given the buffer size *B*, the link capacity *C* and the peak cell rate of the connection *PCRi,* determine a load ρ such that the probability of queue length exceeding *B* is less than ε, where ε is a small number such as 10^{-10}
- Using M/D/1 model:

$$P(BufferLength > x) \approx -\frac{1-\rho}{\ln(\rho)} \exp\left(-x\left(1-\rho-\ln(\rho)\right)\right)$$

- Using nD/D/1 model:

$$P(BufferLength > x) \approx -\frac{1-\rho}{\ln(\rho)} \exp\left(-x\left(\frac{2x}{n}+1-\rho-\ln(\rho)\right)\right)$$

Loss Probability versus Buffer Size



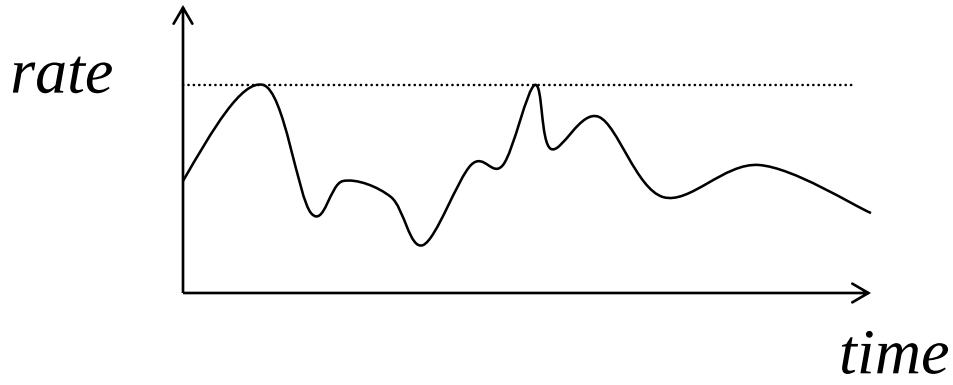
- ρ=0.9
- M/D/1 is conservative
- For large N, both give similar performance

VBR admission control

- VBR
  - peak rate differs from average rate = *burstiness*
  - if we reserve bandwidth at the peak rate, wastes bandwidth
  - if we reserve at the average rate, may drop packets during peak
  - key decision: how much to overbook
- Four known approaches
  - peak rate admission control
  - worst-case admission control
  - admission control with statistical guarantees
  - measurement-based admission control

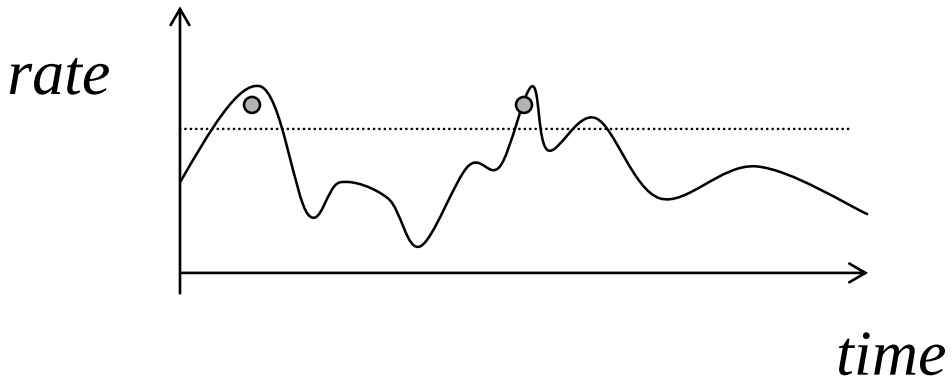
1. Peak-rate admission control

- Reserve at a connection's peak rate
- Pros
  - simple (can use FIFO scheduling)
  - connections get negligible delay and loss
  - works well for a small number of sources
- Cons
  - wastes bandwidth
  - peak rate may increase because of scheduling jitter



2. Worst-case admission control

- Characterize source by ‘average’ rate and burst size (LBAP)
- Use WFQ or rate-controlled discipline to reserve bandwidth at average rate
- Pros
  - may use less bandwidth than with peak rate
  - can get an end-to-end delay guarantee
- Cons
  - for low delay bound, need to reserve at more than peak rate!
  - implementation complexity



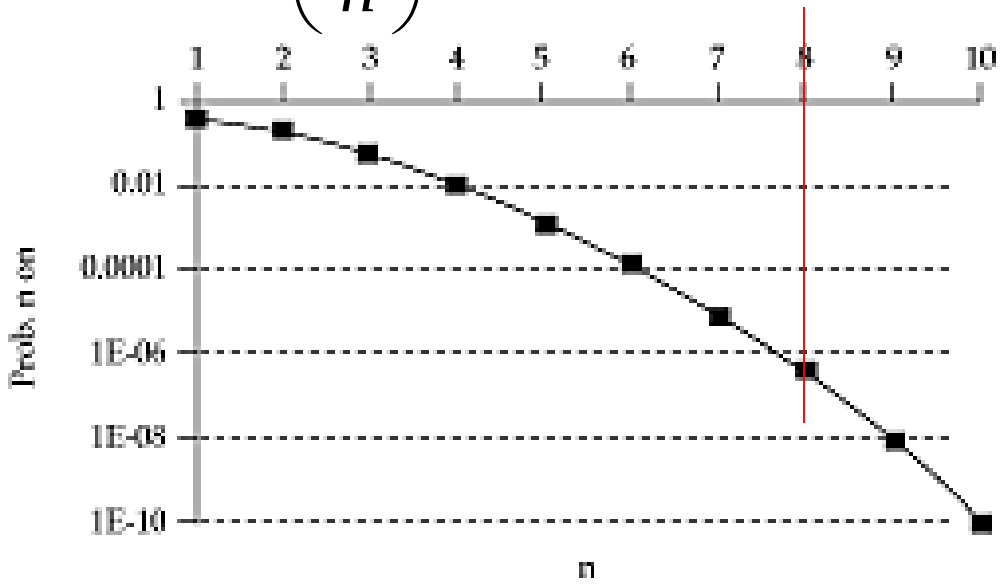
3. Admission with statistical guarantees

- Key insight is that as number of calls increases, probability that multiple sources send a burst decreases
  - sum of connection rates is increasingly smooth
- With enough sources, traffic from each source can be assumed to arrive at its average rate
- Put in enough buffers to make probability of loss low
  - Theory of large deviations quantitatively bounds the overflow probability
- By allowing a small loss, we can reduce the resources considerably

Example

- Consider an ensemble of 10 identical and independent sources, each of which is "on" with a probability 0.1. When "on" has a transmission rate of 1.0. What is the probability that they overflow a shared link of capacity 8?
- The probability that *n* sources are "on" out of 10 is given by

$$\binom{10}{n}(0.1^{n})(0.9^{10-n})$$



The probability of loss is less than 10^{-6}

For peak allocation we need a capacity of 10

By allowing loss, we reduced resources by 20%!!

3. Admission with statistical guarantees (contd.)

- Assume that traffic from a source is sent to a buffer of size *B* which is drained at a constant rate *R*
- If source sends a burst, its delay goes up
- If the burst is too large, bits are lost
- *Equivalent bandwidth (EBW)* of the source is the rate at which we need to drain this buffer so that the probability of loss is less than *L* (and the delay in leaving the buffer is less than *d)*
- If many sources share a buffer, the equivalent bandwidth of each source decreases (why?)
- Equivalent bandwidth of an ensemble of connections is the sum of their equivalent bandwidths

3. Admission with statistical guarantees (contd.)

- When a source arrives, use its performance requirements and current network state to assign it an equivalent bandwidth
- Admission control: sum of equivalent bandwidths at the link should be less than link capacity
- Pros
  - can trade off a small loss probability for a large decrease in bandwidth reservation
  - mathematical treatment possible
  - can obtain delay bounds
- Cons
  - assumes uncorrelated sources
  - hairy mathematics

Effective Bandwidth

- This model maps each connection's traffic parameters into a real number *EBWi*, called the *Equivalent Bandwidth* or *Effective Bandwidth* of the connection such that the QoS constraints are satisfied.
- Thus, the effective bandwidth is derived as a *source property* and with this mapping, the CAC rule becomes very simple:

$$\sum EBW_i \leq Link\,Capacity$$

- For a connection with an average rate *SCRi* and peak rate as *PCRi*, the effective bandwidth is a number between the *SCRi* and *PCRi*. That is,

$$SCR_i \leq EBW_i \leq PCR_i$$

- There are many methods and models published in the literature

Properties of EBW

- *Additive Property*: Effective bandwidths are additive, i.e., the total effective bandwidth needed for *N* connections equals to the sum of effective bandwidth of each connection
- *Independence Property*: Effective bandwidth for a given connection is only a function of that connection's parameters.
  - due to the independence property, the effective bandwidth method could be far more conservative than a method which considers the true statistical multiplexing (i.e., the method which considers the presence of other connections)
- With the effective bandwidth's method, the CAC function can add (or subtract) the effective bandwidth of the connection which is being set-up (or torn down) from the total effective bandwidth. *This is not easily possible with any method which does not have the independence property*

EBW (First Approach by Roberts)

- Assumes fluid sources and zero buffering (so that two simultaneously active sources would cause data loss)
- Let each source has a peak rate *P*, mean rate *m* and link capacity is *C* and required cell loss is smaller than 10^{-9}
- The *heuristic* to estimate the EBW of a source is:
  - *EBW = 1.2m + 60m(P-m) / C*
- First term says EBW is 1.2 times of mean rate
- Second term increases EBW in proportion to the gap between peak and mean (an indicator of source burstiness). This is mitigated by the large link capacity.
- Expression is independent of cell loss!!

EBW (Second approach by Gibbens and Hunt)

- on-off sources with exponentially distributed ‘on’ and ‘off’ periods
- Let a source mean “on” period be $1/\mu_i$ and mean “off” period be $1/\lambda_i$. When the source is “on”, it is assumed to produce information at a constant rate γ_i
- Let B be the buffer size; *CLR* is the cell loss ratio required and

$$(\log CLR)/B = \zeta \in [-\infty, 0]$$

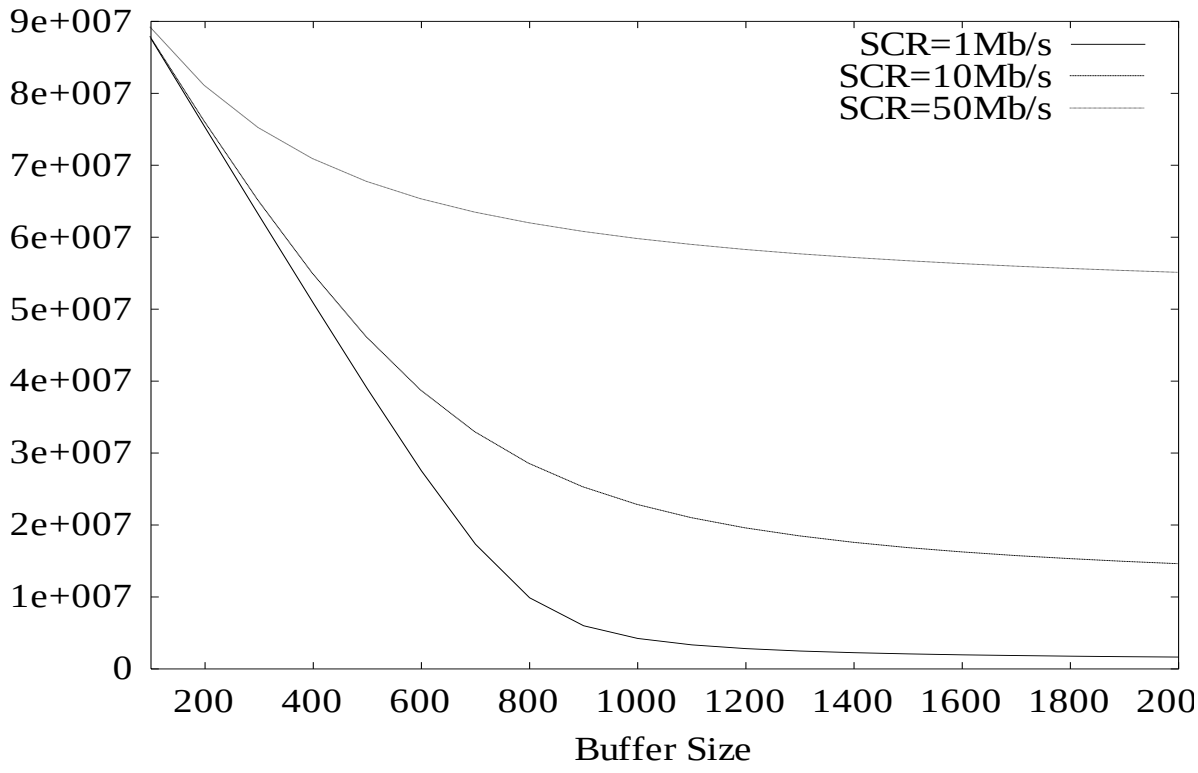
- The Effective Bandwidth is given by:

$$c_i = \frac{(\zeta\gamma_i + \mu_i + \lambda_i) - \sqrt{(\zeta\gamma_i + \mu_i - \lambda_i)^2 + 4\lambda_i\mu_i}}{2\zeta}$$

Example

- Let traffic descriptors are SCR, PCR=100Mb/s, CLR=10^{-7} and ABS (Average Burst Size)=50 cells

$$\mu_i = PCR / ABS \qquad \lambda_i = \mu_i . SCR_i / (PCR - SCR) \qquad \gamma_i = PCR$$



EBW Observations

- Equation implies that for large *B*, $\zeta \to 0$ and EBW (*ci*) equals to the mean rate of the source

$$\lambda_i \gamma_i / \left(\lambda_i + \mu_i \right)$$

- For a small buffer *B*, $\zeta \to -\infty$ and the effective bandwidth of the source will be , the peak information rate

$$c_i = \gamma_i$$

- The queue length distribution is assumed to be asymptotically exponential of form:

$$P(\text{Queue Length} \geq B) \approx e^{-f(c_i)B}$$

EBW for Self-similar traffic (By Norros)

- Let *m* is the mean bit rate of the traffic stream, *a* is the coefficient of variation, *B* is the buffer size, *H* is the Hurst parameter of the stream *(0.5≤H≤1)*, *CLR* is the target cell loss ratio.
- The EBW is given by

$$C = m + \left(H^H (1-H)^{(1-H)} \sqrt{-2\ln(CLR)}\right)^{1/H} a^{1/(2H)} B^{-(1-H)/H} m^{1/(2H)}$$

- Note that this equation does not follow the asymptotic exponential queue length distribution

Multi-class CAC

- In the real world, the traffic flow consists of multiple QoS classes, where, the services may be partitioned and queued separately
- To guarantee QoS, a certain amount of bandwidth (or capacity) is reserved for each of the service categories.
- With effective bandwidth approach, this assignment becomes very simple.
  - Let N_j be the number of sources for class j and let α_j be the effective bandwidth of a source belonging to class j. Let there be K such classes. Then, the CAC for multi-class traffic should check that the total estimated capacity is less than the service rate. That is,

$$\sum_{j=1}^{K} N_j \alpha_j \leq LinkCapacity$$

4. Measurement-based admission

- For traffic that cannot describe itself
  - also renegotiated traffic
- *Measure* 'real' average load due to ensemble of connections
- Users tell peak
- If peak + measured average load < capacity, admit
- Over time, new call becomes part of average
- Problems:
  - assumes that past behavior is indicative of the future
  - how long to measure?
  - when to forget about the past?