Issues in Flexible Resource Management for Datagram Networks

Sally Floyd
Lawrence Berkeley Laboratory
floyd@ee.lbl.gov

3rd Workshop on Very High Speed Networks

March 9, 1992

(Based in part on work by Van Jacobson,
Dave Clark, and other members
of the Internet End-to-End Research Group)

Goals:

- Support connections that require bandwidth guarantees (e.g., packet voice and video).

- Support ‘quality of service’ (e.g., interactive telnet vs. bulk data ftp).

- Support flexible link sharing.

Methods:

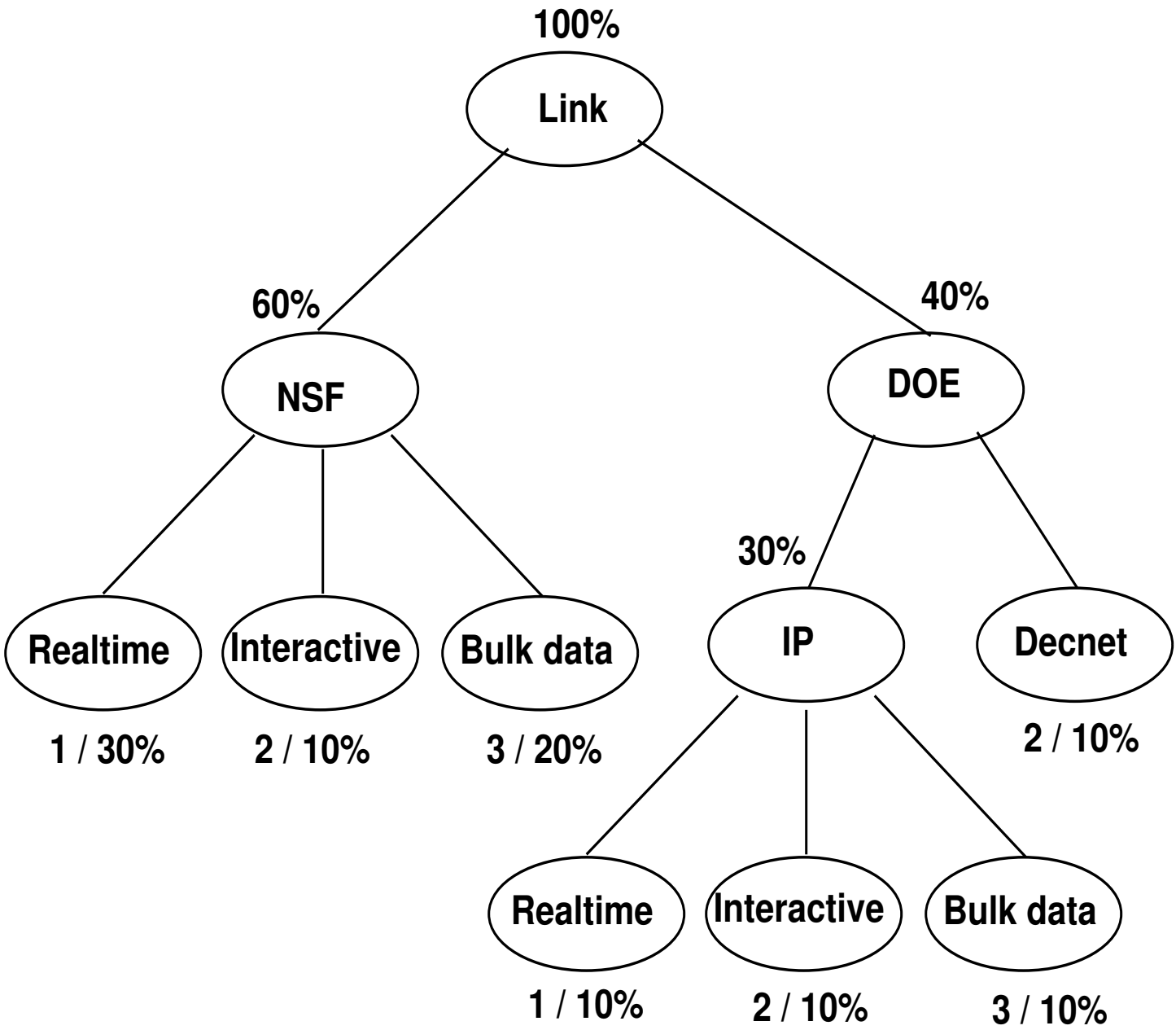
- Separate low-level mechanisms from high-level policy, to allow evolution.

- Aggregate connections in classes. Each class has a priority and a throughput allocation.

- Construct a hierarchy of classes.

- Avoid extensive per-conversation parameterization.

Example class hierarchy:



Priority/Allocation

Mechanisms:

- Classifier: Map packets to classes, using info in the packet header.

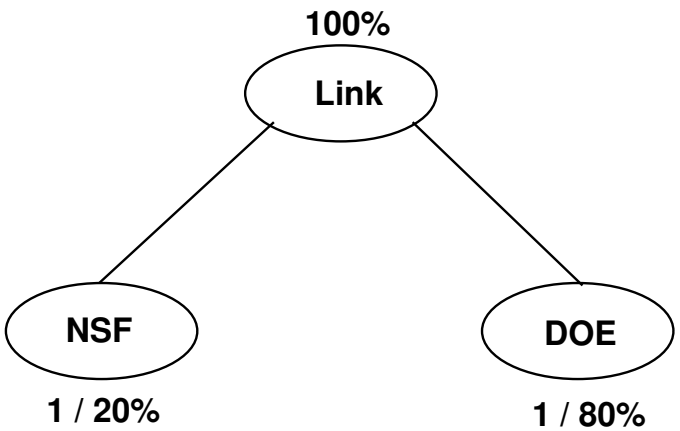
- Selector: Find a class that is either underlimit (under its throughput allocation) OR able to borrow. Highest priority first. Round-robin within same priority.

- Estimator: Compute a short-term estimate of the class's throughput. Average the inter-departure times between packets, using EWMA (exponential weighted moving average).

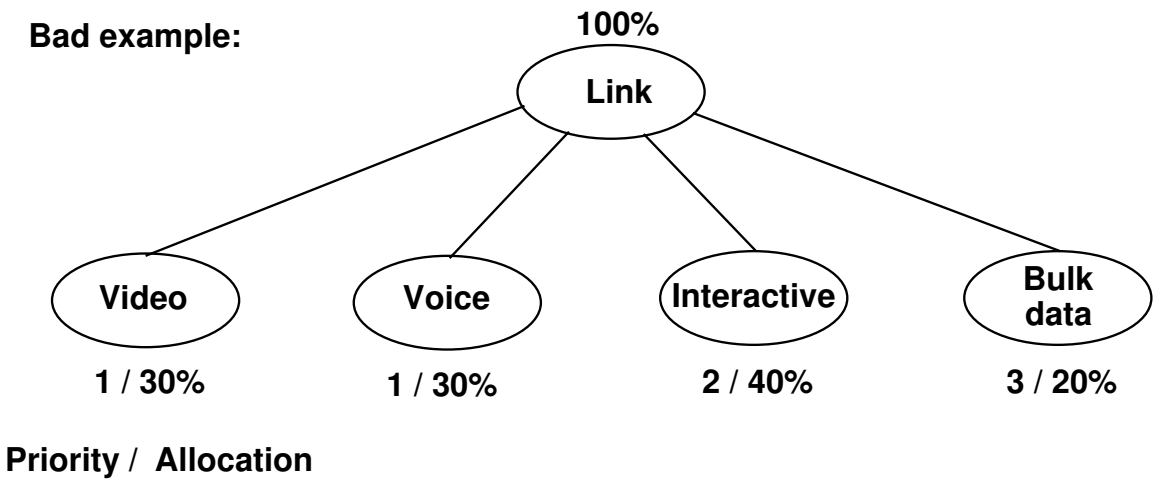
- Delayer: For a class that is overlimit AND unable to borrow, compute the next time this class is allowed to send a packet. A delayed class is limited to its allocated throughput.

Guidelines:

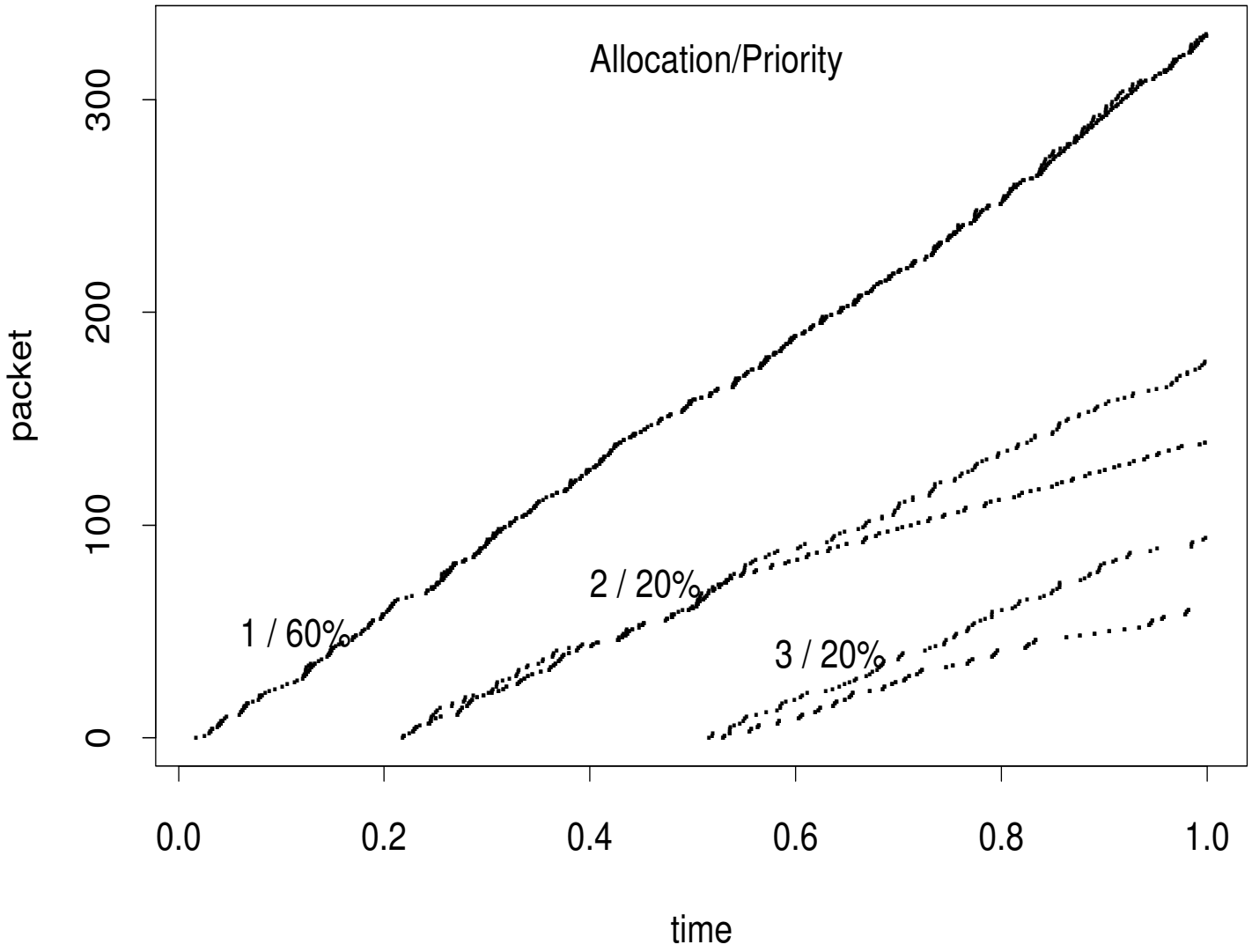
- Don’t allow borrowing from a parent class that is near its allocated bandwidth.



- The sum of the allocations to the leaf classes must not exceed the link bandwidth.



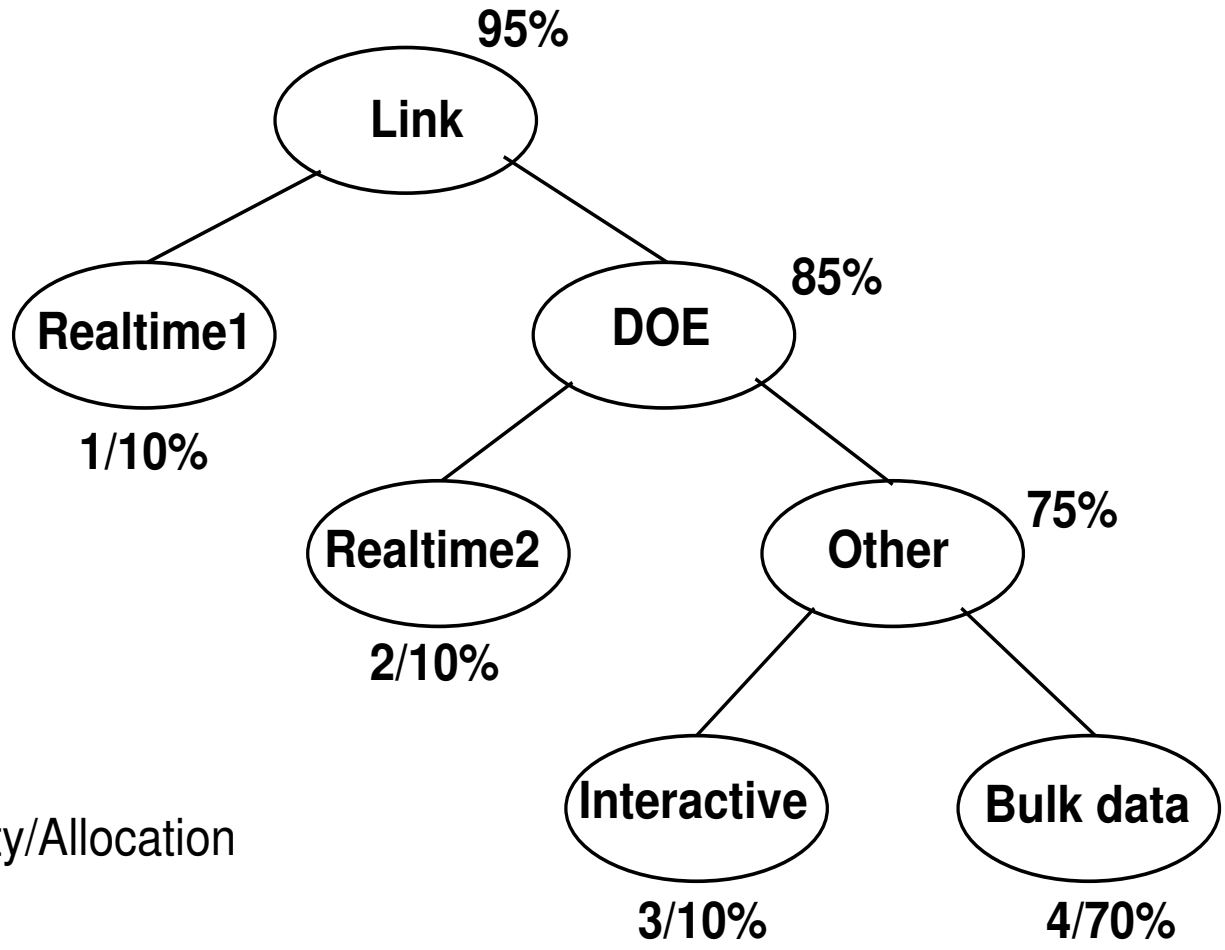
Simulations:



Analysis:

- Assuming that the guidelines are followed, then we can prove that no class will be denied its allocated throughput, and there is a limit on the time that a class has to wait to receive its allocated throughput.

Bounded delay for low-priority classes:



Assume that utilization has been low, and then each of the four classes would like to use all of the link. Class 'Realtime1' sends until the parent class 'Link' is over limit. Then class 'Realtime1' is restricted to its allocation. Class 'Realtime2' sends until the parent class 'DOE' is over limit, and so on. Finally, classes 'Realtime1', 'Realtime2', and 'Interactive' are restricted to their allocations, and class 'Bulk data' gets to send.

Delay allocation

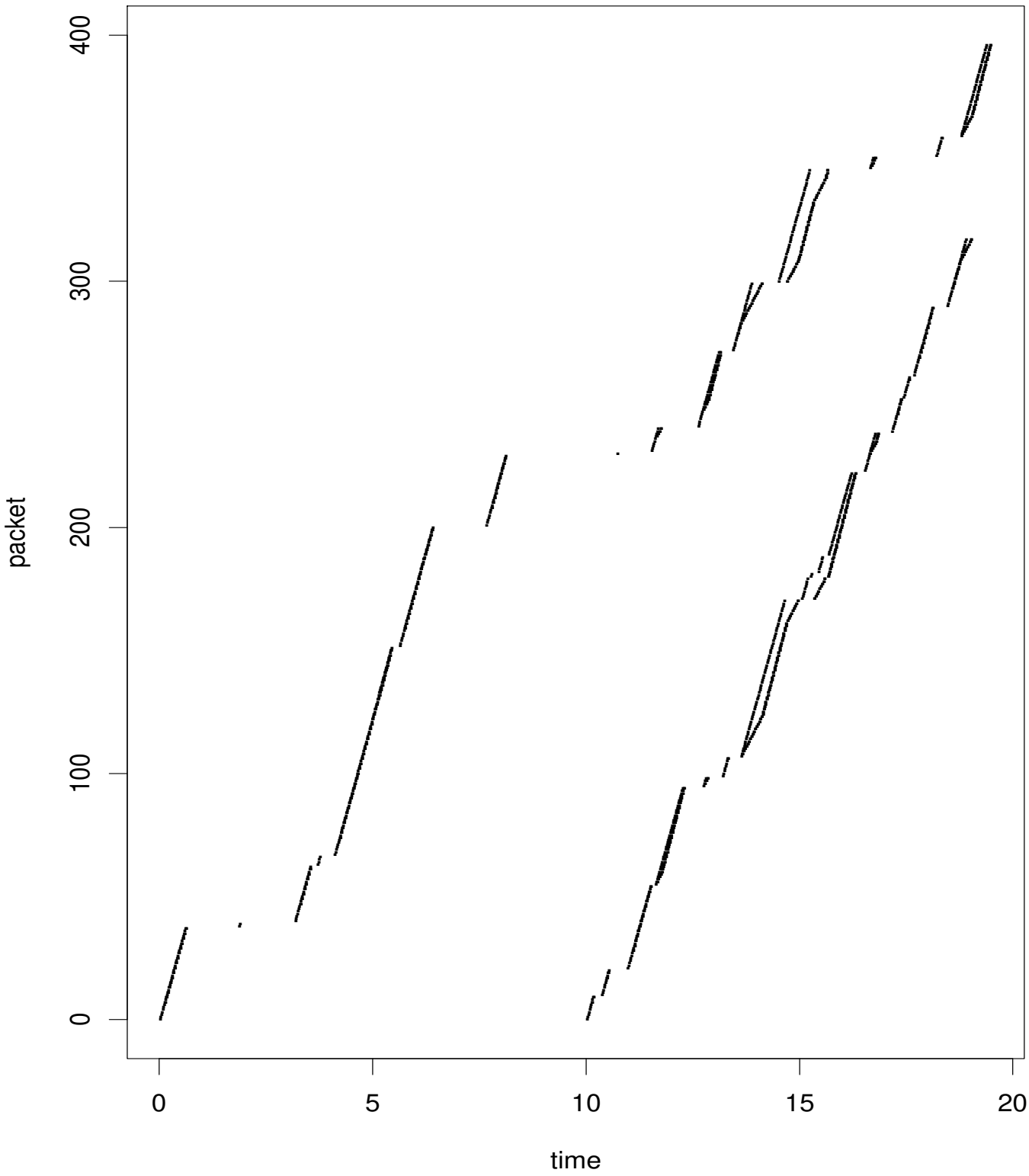
- Causes of delay or distortion:
  - An arriving packet waiting for the current packet to finish;
  - A class waiting for classes of equal or higher priority;
  - A delayed class (a class that is over its allocation and unable to borrow) which is rate limited to its allocated bandwidth.

For a class that requires low delay:

- Assign high priority.

- Use an admissions control procedure (or other procedure for congestion aviodance) so that the class does not exceed its allocation.

Simulations with two voice connections in one class



Conformance testing

- For classes with admissions controls.

- The class-based resource management mechanism can be used for conformance testing, as an alternative to leaky bucket, by assigning a connection to its own class at the edge of the network. The resource management mechanism is capable of:
  - Controlling the average rate over an interval I (using the allocated throughput and disabling borrowing),
  - Controlling the peak rate (using the delay-queue mechanism), and
  - Controlling the interval over which the peak rate is permitted (using class parameters).

- Rathgeb [JSAC 4/91] compares leaky bucket and a slightly-different EWMA mechanism with four other policing mechanisms, and reports that leaky bucket and EWMA mechanism are the most promising methods.

Open questions: admissions controls

Plans for simulations with classes that have admissions control:

- For a class with few connections, use the traffic parameters for each connection to compute the ”equivalent capacity” for the class. [Guerin et al., 1991]

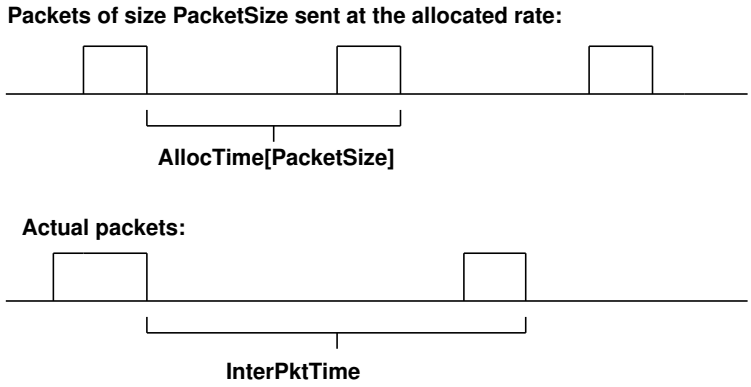
- For a class with many connections, estimate the class’s throughput based on current traffic. Possible methods:
  - Measure the distribution of the number of packets arriving in time T. [Saito et al.]
  - or: Measure the average throughput for the class, along with the variance, using EWMA. Use these to compute the ”equivalent capacity” for the class [using Guerin].
  - or: Use quick simulation methods to compute the probability of buffer overflow if the traffic is increased [Kesidis and Walrand, 1992].

Open questions: Traffic parameters

- The specification of a connection's traffic parameters is still an open question.
  (Peak rate, average rate, burst interval, sensitivity to delay?)

Computing the average rate:

AllocTime[PacketSize]: the time between packets when the class transmits packets at the allocated rate.



InterPktTime: the actual time between packets.

- Method A:
  IDLE is InterPktTime - AllocTime[PacketSize].
  (1) AvgIDLE is positive when the class is over its allocated rate, and negative otherwise.
  (2) The average rate of the class is easily computed IF all packets are of the same size.

- Method B:
  IDLE is InterPktTime/AllocTime[PacketSize] - 1.
  Condition (1) holds.
  (2) The average rate of the class is easily computed.

Advantages of the class-based resource management scheme

- Priority and throughput allocation are independent.

- Admissions control is optional, and is used if desired WITHIN a class. The priority/throughput guarantees for classes are independent of the admissions control procedures.

- The ‘borrowing’ of unused throughput is controlled by the class hierarchy. (Among other things, this permits link-sharing.)

- Guaranteed service could be provided by priority-1 classes, if desired.

- Per-conversation and per-packet state is low.