Link-sharing and Resource Management Models for Packet Networks

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

April 23, 1997

(Joint work with Van Jacobson.
Credits to other members
of the Internet End-to-End Research Group)

Overview of talk:

- Overview of CBQ (class-based queueing)
- Statement of link-sharing goals
- Formal link-sharing guidelines
- Priority scheduling in a link-sharing framework
- Relationship to other work
- Conclusions

And if there is extra time:
Implementation issues, Analysis of delay.

Policy goals for CBQ:

- 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 bandwidth allocation.
- Construct a hierarchy of classes, to simultaneously capture QoS and link-sharing constraints.
- Avoid extensive per-conversation parameterization.

Examples of link-sharing:

- Sharing a link between higher-priority (e.g., realtime) and lower-priority traffic.

- Sharing a link between protocol families (e.g., IP and SNA).

- Sharing a link between traffic classes with different responses to congestion (e.g., Mbone and TCP).

- Controlling the bandwidth used by a particular traffic class (e.g., WWW).

Specialized places for link-sharing.

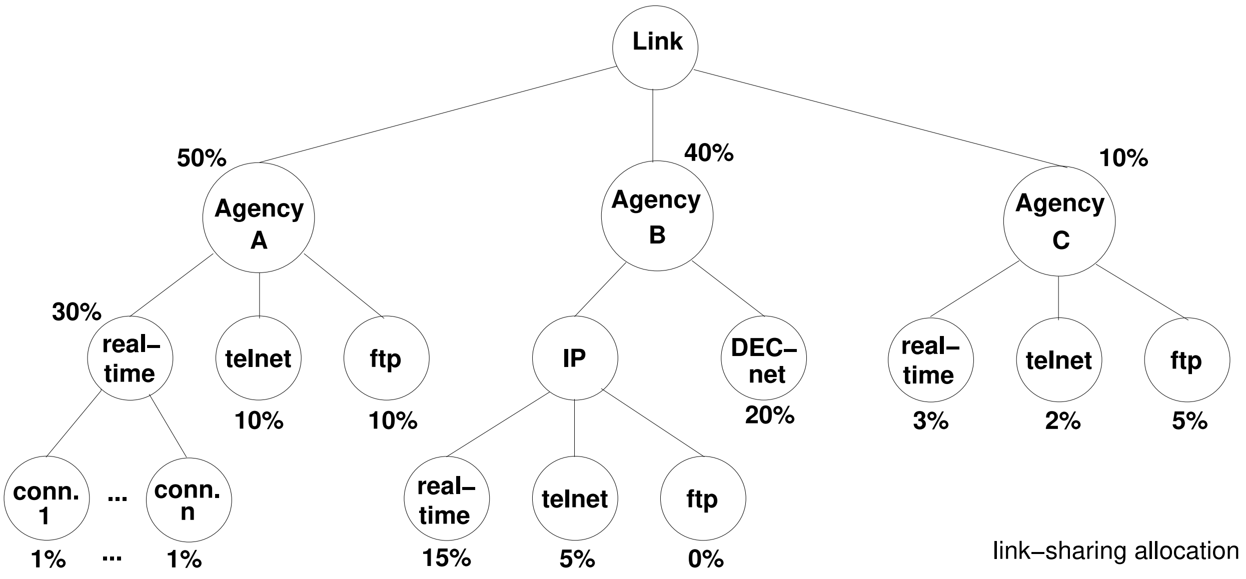
- ISDN lines.

- The Trans-Atlantic FAT pipe shared by three organizations.

- Firewalls (where traffic classification already takes place).

Link-sharing is in part a local issue, to provide local control over the distribution of bandwidth.

Example class hierarchy:



- Link-sharing between organizations, protocol families, and/or traffic types. We believe that needs met by link-sharing are fundamental (in the absence of free, infinite bandwidth).

- Hierarchy allows:
  - simultaneous QoS and sharing constraints;
  - flexible multiplexing via controlled ”borrowing”;
  - delegation.

- Different links in the network (e.g., an ISDN line to the home, a link in a backbone network) will have different class structures.

Link-sharing and Differential Services in the Internet

- The emphasis in the research community up until now has been on providing real-time services for individual audio or video flows (similar to the model of a telephone call).

- There is an increasingly-recognized need for longer-term controlled bandwidth management, and for higher levels of aggregation in the interior of the Internet.

Mechanisms:

- Classifier: Map arriving packets to classes, using information in the packet header. CBQ makes no assumptions about how traffic is assigned to classes.

- Estimator: Compute a short-term estimate of the class's bandwidth.

- Selector: Find the class that is allowed to send the next packet. (In CBQ, look for the highest priority class, then use round-robin within classes of the same priority.)

- Delayer: For a class that is over its link-sharing allocation and contributing to congestion, compute the next time this class is allowed to send a packet. A delayed class is rate-limited to its allocated link-sharing bandwidth.

Link-sharing goals:

- Each interior or leaf class should receive roughly its allocated link-sharing bandwidth over appropriate time intervals, given sufficient demand.

- If all leaf and interior classes with sufficient demand have received at least their allocated link-sharing bandwidth, the distribution of any `excess' bandwidth should not be arbitrary, but should follow some set of reasonable guidelines.

Non-goals:

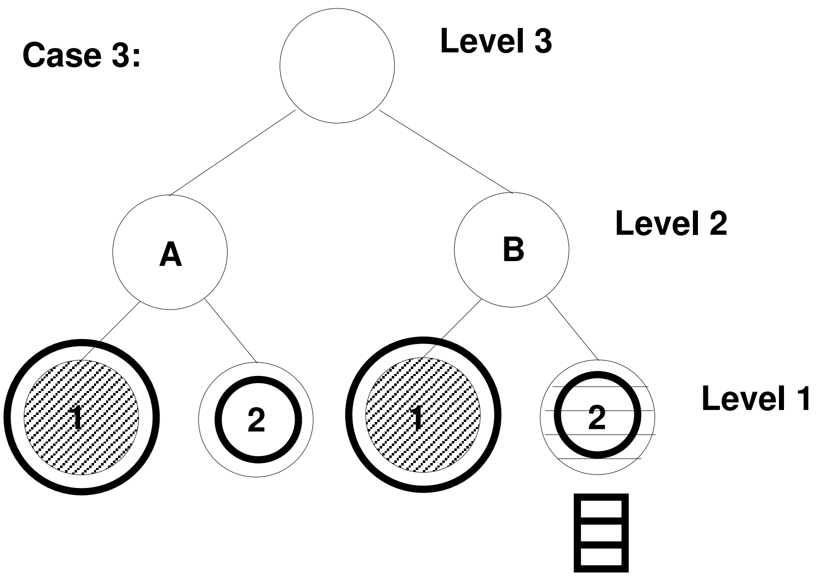
- Congestion-control/congestion avoidance

- Fair sharing (for some definition of “fair”)

“Non-goals” does not mean that these are non-problems. Other machinery solves these problems working within this framework.

Definition of terms:

- general and link-sharing scheduler
- regulated and unregulated classes
- overlimit, underlimit, and at-limit classes
- satisfied and unsatisfied classes
- levels in the class structure



Formal link-sharing guidelines:

A class can continue unregulated if one of the following conditions hold:

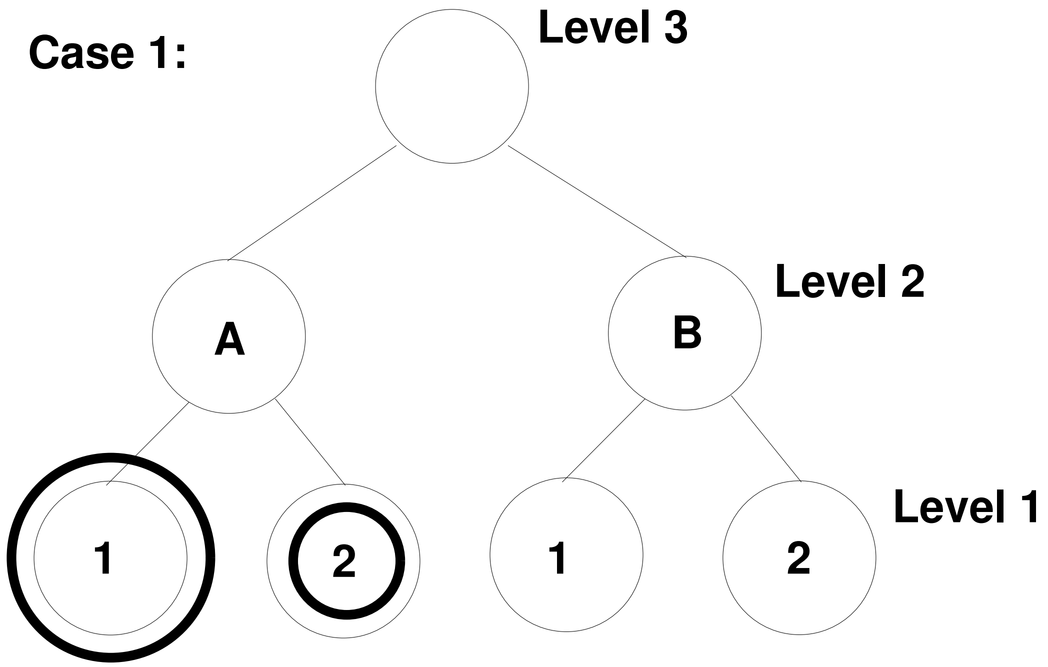
- **1:** The class is not overlimit, OR
- **2:** The class has a not-overlimit ancestor at level i, and there are no unsatisfied classes in the link-sharing structure at levels lower than i.

Otherwise, the class will be regulated by the link-sharing scheduler.

A regulated class will be rate-limited to its link-sharing bandwidth.

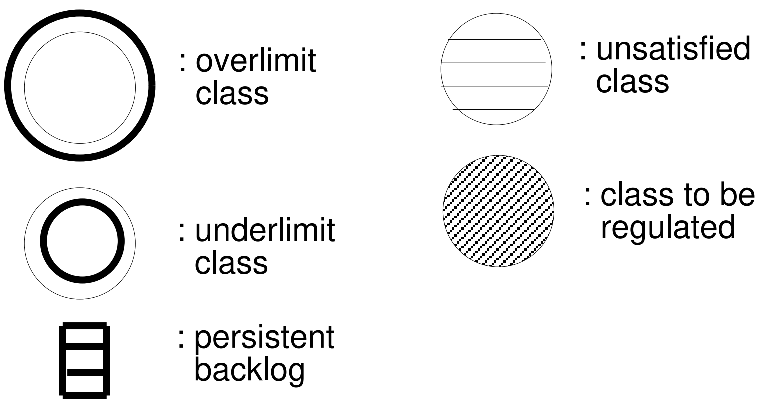
Examples of the link-sharing guidelines:

Case 1:

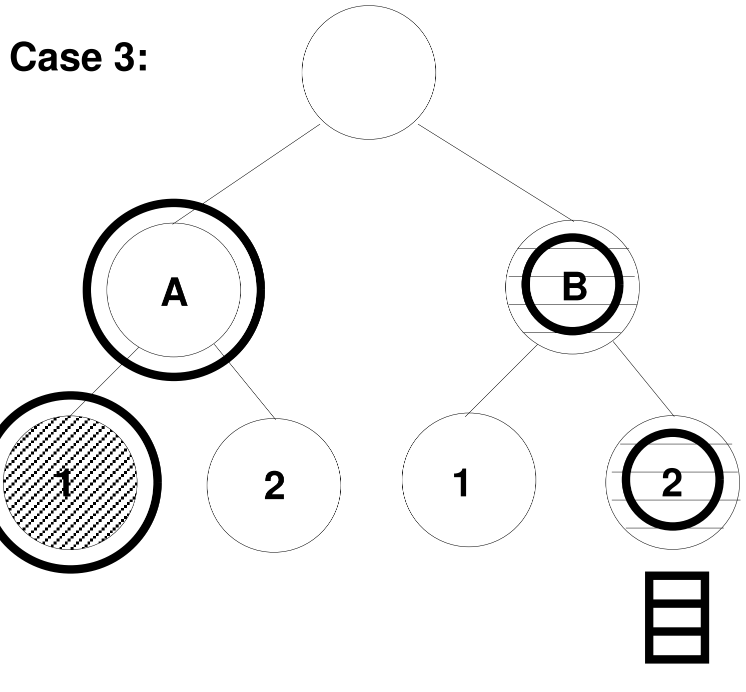


- No classes have to be regulated.

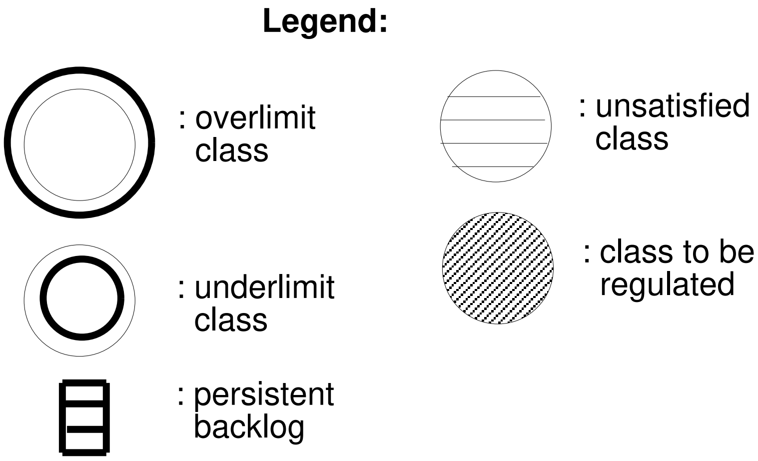
Legend:



More examples of the link-sharing guidelines:

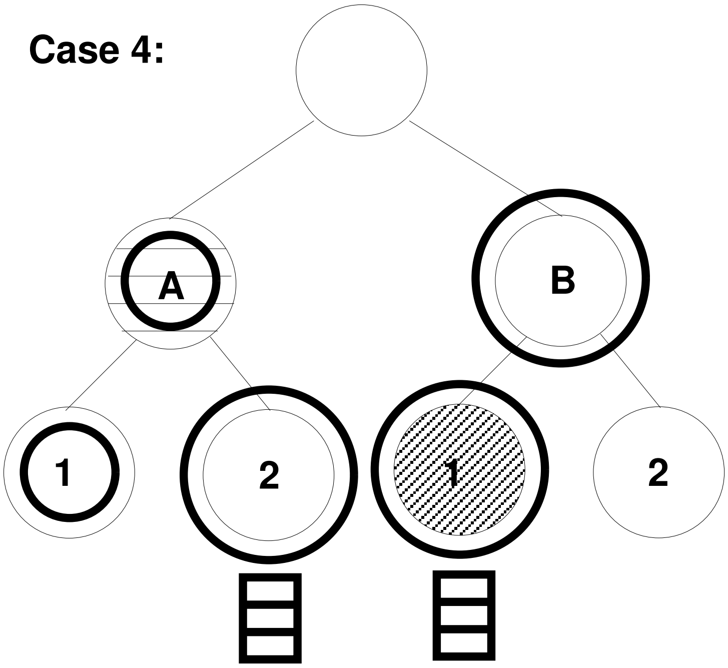


- Both the Agency B class and Agency B non-realtime class are unsatisfied. The Agency A realtime class needs to be regulated.


Legend:
: overlimit class
: underlimit class
: persistent backlog
: unsatisfied class
: class to be regulated

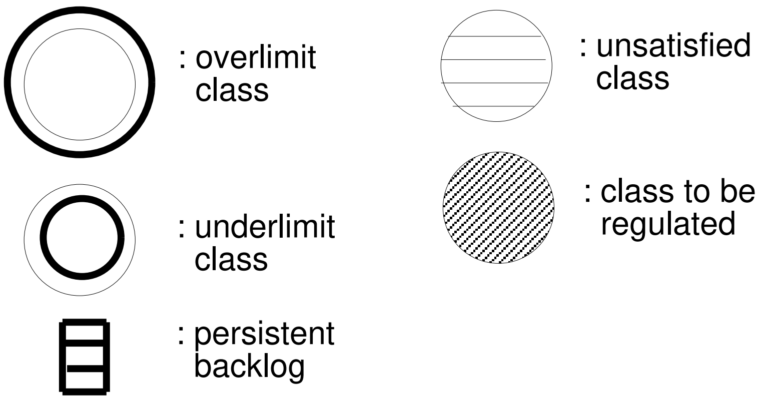
More examples of the link-sharing guidelines:

Case 4:

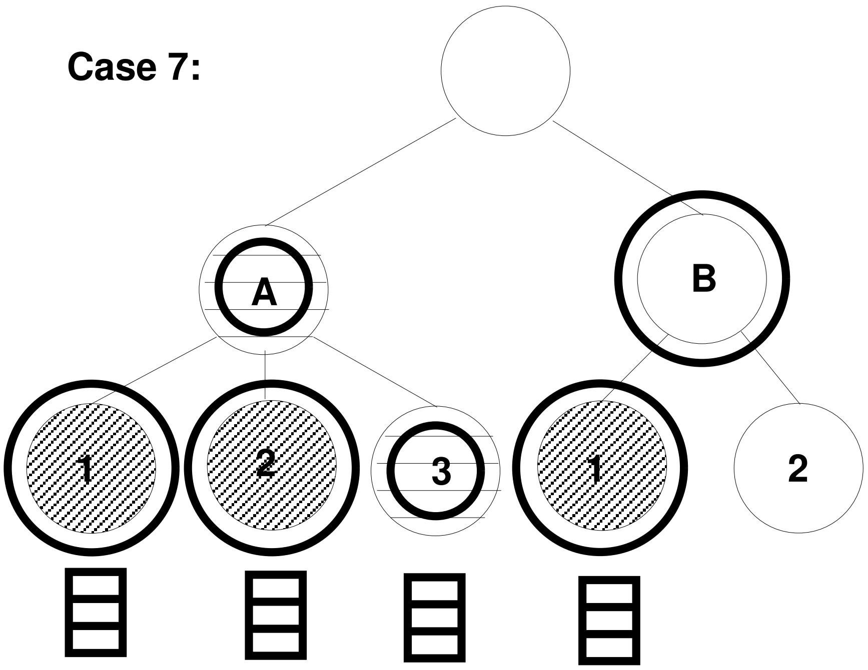


- The Agency A class itself is unsatisfied. The Agency B realtime class needs to be regulated.

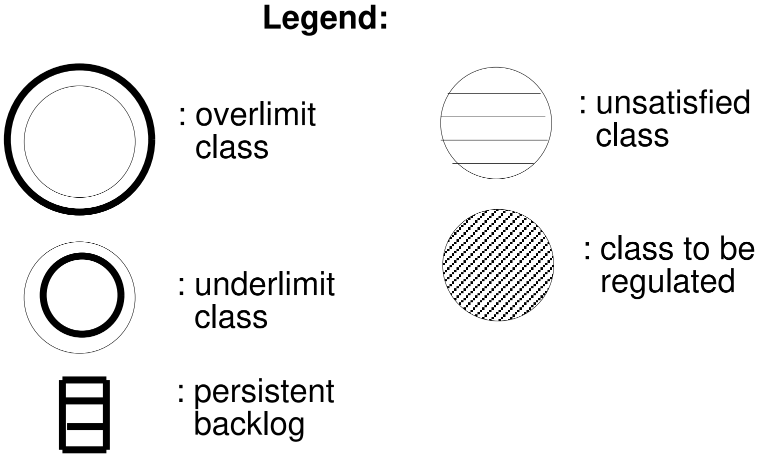
Legend:



More examples of the link-sharing guidelines:



- Both the Agency A ftp class and the Agency A class itself are unsatisfied. All three overlimit leaf classes have to be regulated.



Approximations to the Formal link-sharing guidelines:

Ancestors-Only link-sharing guidelines.
A class can continue unregulated if one of the following conditions hold:

- **1:** The class is not overlimit, OR
- **2:** The class has an underlimit ancestor.

Otherwise, the class will be regulated by the link-sharing scheduler.

- Advantages: ease and efficiency of implementation
- Disadvantages: performance is less robust

Approximations to the Formal link-sharing guidelines:

Top-Level link-sharing guidelines.
A class can continue unregulated if one of the following conditions hold:

- **1:** The class is not overlimit, OR
- **2:** The class has an underlimit ancestor whose level is at most *Top-Level*.

Otherwise, the class will be regulated by the link-sharing scheduler.

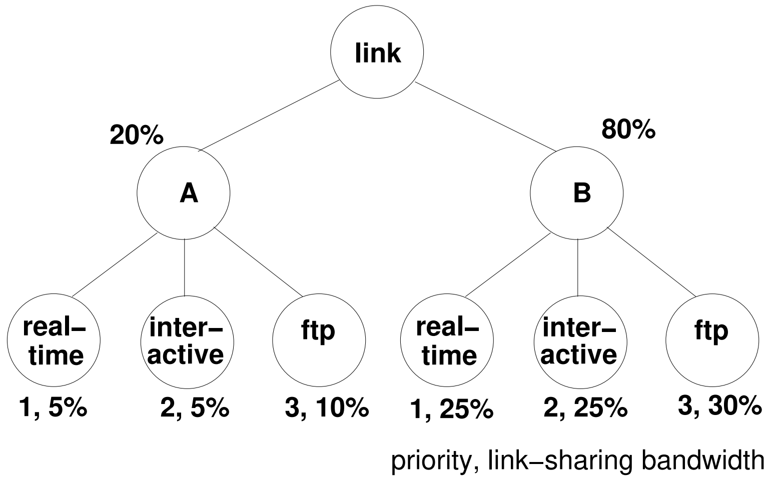
Heuristics for setting the Top-Level variable:

- *Top-Level* is set to the lowest level known to have an unsatisfied class.

Priority scheduling in a link-sharing framework:

- Delay-sensitive and throughput-sensitive traffic

- Investigate the advantage (or lack of advantage) of incorporating priorities.



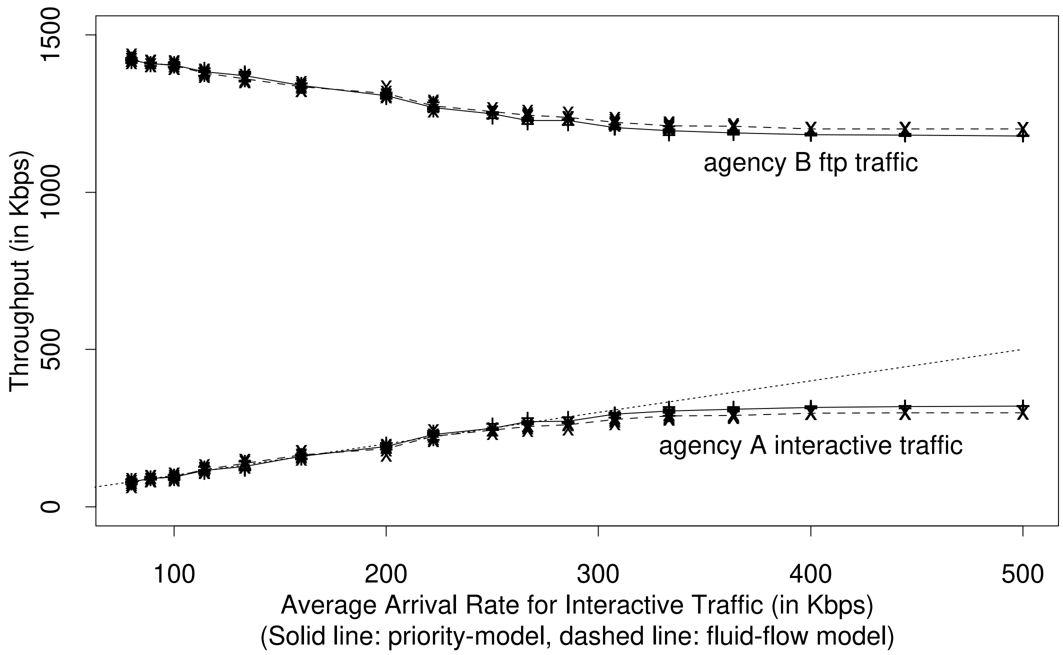
- Agency A interactive class: UDP, bursts of four 1000-byte packets are sent at exponential time intervals.

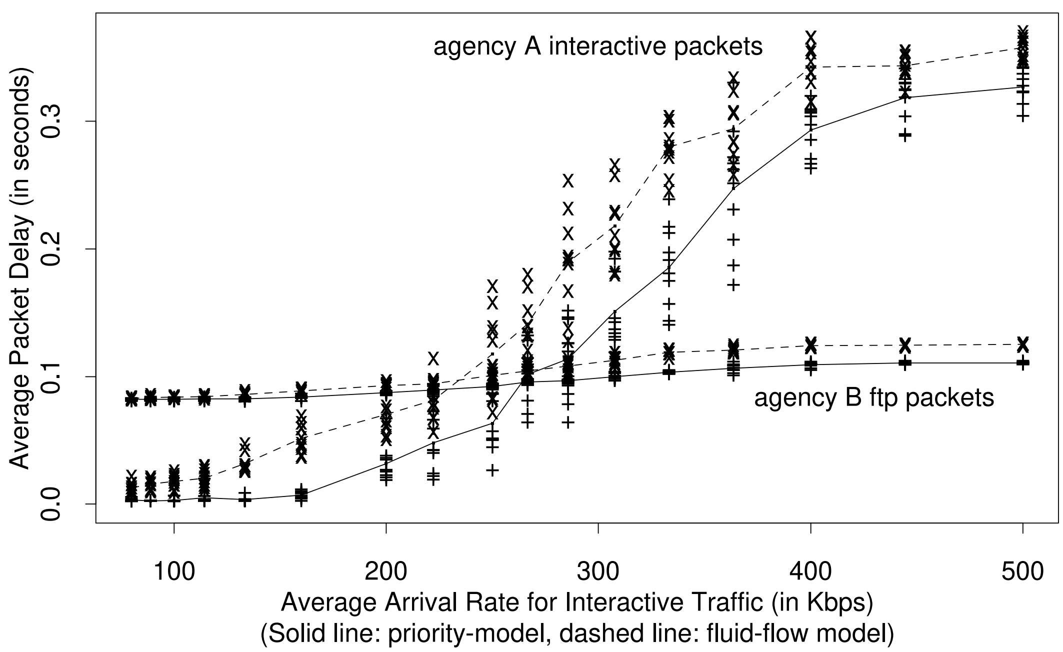
- Agency B interactive class: UDP, single 50-byte packets at exponential time intervals.

- FTP class: three TCP connections.
  Each class has its own queue with a 20-packet buffer.

Priority scheduling: simulation results

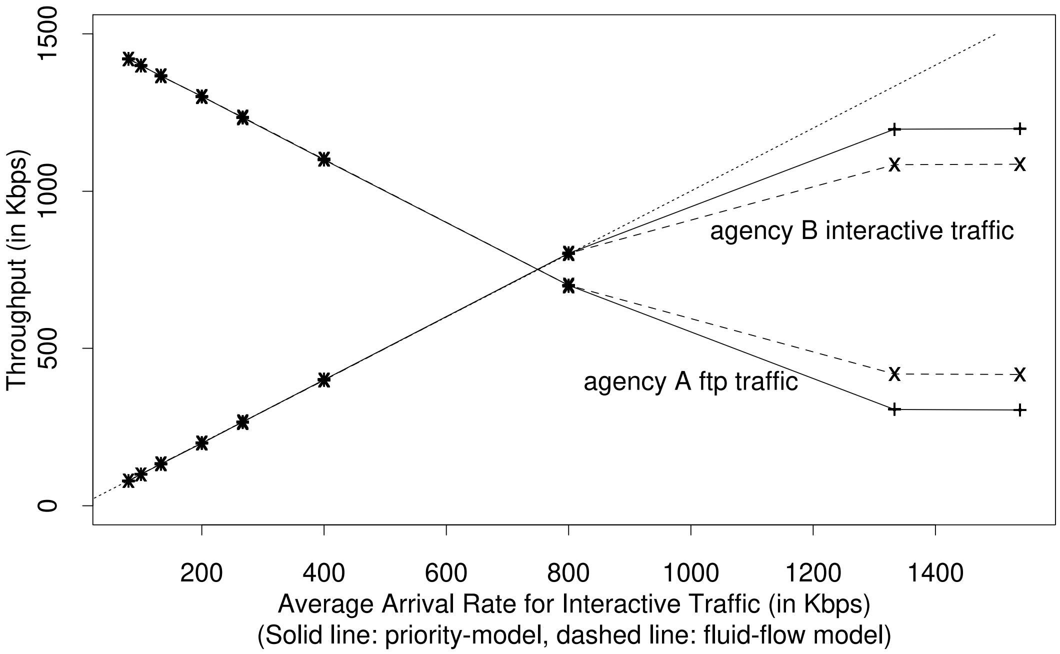
A higher-priority class with bursty arrivals, receiving a small fraction of the link bandwidth.

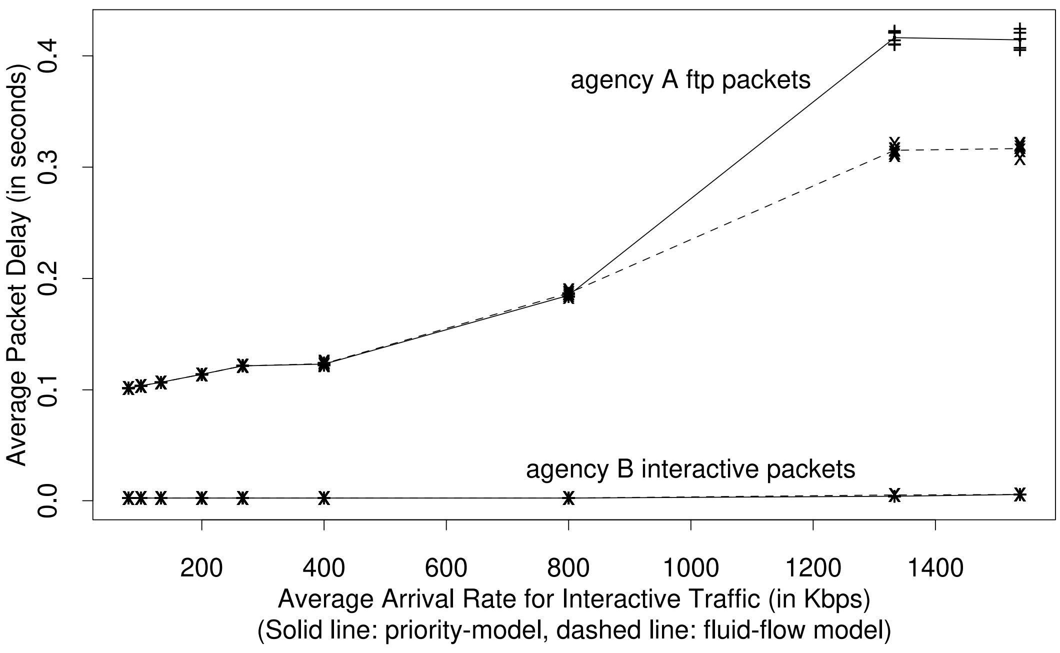




Priority scheduling: simulation results

A higher-priority class with smooth arrivals, receiving a large fraction of the link bandwidth.





Priority scheduling: Conclusions

- A higher-priority class with bursty arrivals receiving a moderate fraction of the link bandwidth. Priority scheduling reduces delay for the higher-priority class, without reducing throughput for the lower-priority class.

- A higher-priority class with fairly smooth arrivals, receiving a large fraction of the link bandwidth. Priority scheduling is of little benefit in reducing delay.

Link-sharing and realtime traffic:

- For a class of controlled-load realtime traffic with a measurement-based admissions control procedure, if the admission control procedure's prediction of future traffic is incorrect, and the controlled load class becomes oversubscribed, the choice at the gateway is to limit the bandwidth of the controlled load class (e.g., using CBQ), or to risk starvation of lower-priority classes.

- For emerging realtime applications such as source- or receiver-based rate-adaptive video classes, one possibility would be for this traffic to be aggregated in a CBQ class with high-priority and an allocated bandwidth.

Related work:

- Bennett and Zhang, “Hierarchical Packet Fair Queueing Algorithms”, Sigcomm 96.

- Goyal, Vin, and Cheng, “Start-time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks”, Sigcomm 96. Both of these papers support hierarchical link-sharing building on Fair Queueing algorithms. Neither, however, incorporates priority-based scheduling.

- Shreedhar and Varghese, “Efficient Fair Queueing using Deficit Round Robin”, SIGCOMM 95. This is similar to the Weighted Round Robin within a priority level in CBQ.

- Stoica and Zhang, “A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-time and Priority Services”, to appear in SIGCOMM 97. This is based on service curves, and decouples the delay and bandwidth allocation for each class.

Related work on CBQ:

- Floyd, S., and Jacobson, V., Link-sharing and Resource Management Models for Packet Networks, IEEE/ACM Transactions of Networking, Vol. 3 No. 4, pp. 365-386, August 1995. (Most is this talk is from this paper.)

- CBQ-related papers, implementations, and commercial products: http://ftp.ee.lbl.gov/floyd/cbq.html.

- CBQ is implemented in the ns-1 simulator (http://www-nrg.ee.lbl.gov/ns/), and soon will be available in ns-2.

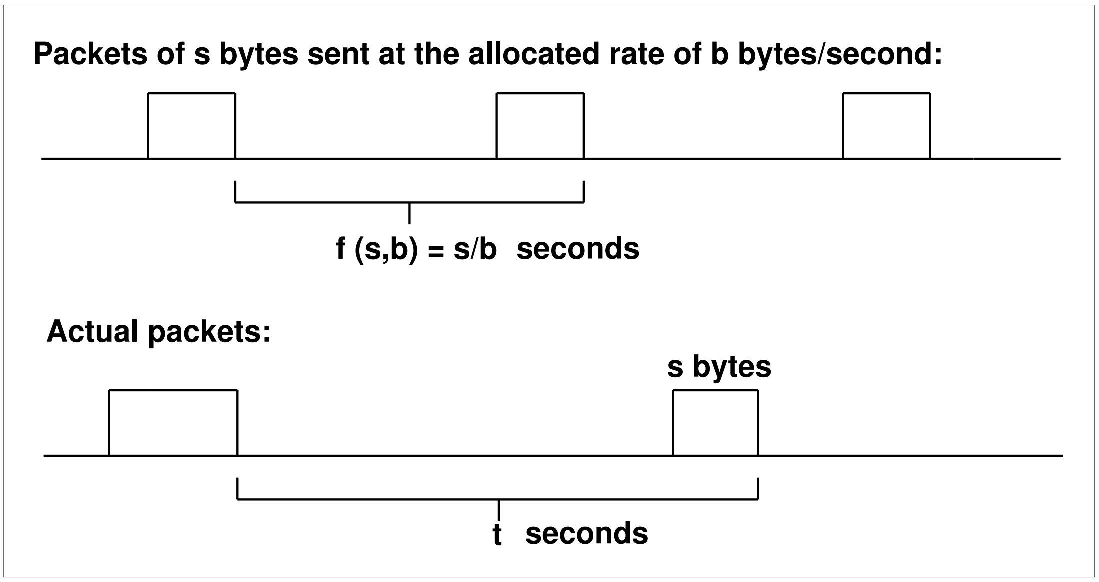
Open Questions:

- What mechanisms are needed for administrative control, and for communication both within and across administrative boundaries? Are the existing tools of RSVP and SNMP sufficient, or is something else needed?

- How can the aggregation of traffic be handled in the interior of the network?

Implementation issues: the estimator.

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



- InterPktTime: the actual time between packets.

- IDLE is InterPktTime - AllocTime[PacketSize].

- AvgIDLE is positive when the class is over its allocated rate, and negative otherwise.

Implementation issues: the classifier.

- Uses information in the packet to locate the appropriate class structure. Relevant fields:
  - source and destination IP addresses, protocol, TOS field (in IP header),
  - source and destination port numbers (in UDP or TCP header).
- A classifier implemented by Wakeman et al. is now being used with CBQ to control link-sharing in the FAT pipe.

Implementation issues: the selector.

- Inspects classes in priority order.

- Inspects classes at the same priority using weighted round-robin.
  (Each class at each round gets to send its weighted share in bytes, including finishing sending the current packet. That class's weighted share for the next round is decremented by the appropriate number of bytes.)

- Invokes the delayer for an overlimit class that is unable to borrow.

Implementation issues: the delayer.

Rate-limits overlimit classes to their allocated bandwidth.

Analysis: priority one classes.

- A class that is not overlimit will not be regulated.

- When all classes are satisfied, no classes will be regulated.

- Assume a priority-based scheduler that uses weighted round-robin among classes of the same priority. Further assume that at most half of the bandwidth is allocated to priority-one classes.

  Then each priority-one class is guaranteed to receive its allocated share of the bandwidth in each round-robin round, unless some of this allocation was used `in advance' in the most recently-sent packet from that connection.

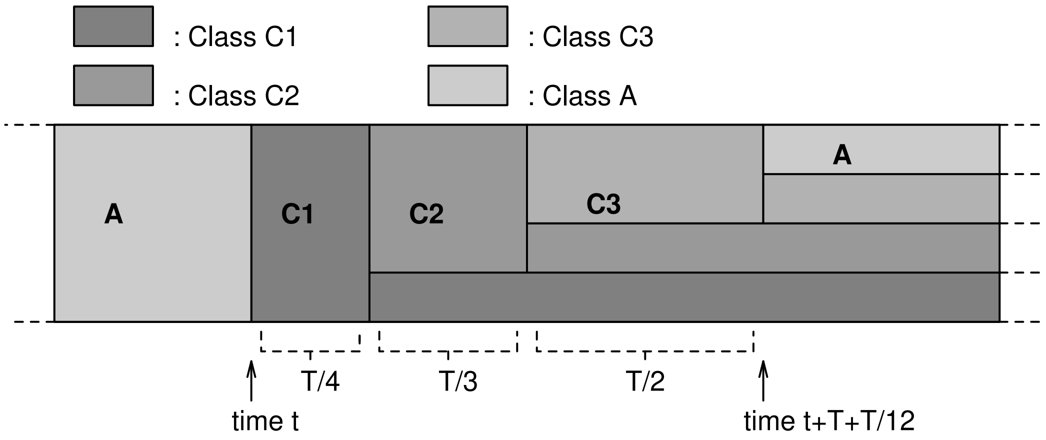
Scheduling for priority-one classes:

- With round-robin scheduling among priority-one classes, the worst-case packet delay at a router is a function of the time that it takes the scheduler to go through one round of the round robin. This worst-case delay is the same for all priority-one classes, regardless of their bandwidth allocation.

- With Fair Queueing scheduling, the worst-case delay at a router is a function of the bandwidth allocation for that class or flow. That is, classes with high bandwidth allocations have small worst-case delays, and classes with small bandwidth allocations have large worst-case delays.

- Weighted round-robin can be modified to explore the continuum between the two extremes above.

Analysis: Limits on starvation for lower priority classes.



- Four classes with equal link-sharing allocations, where class A has the lowest priority. Class A can be denied bandwidth for $T + T/12$ seconds.

- Given n classes in the link-sharing structure, with link-sharing allocations p_1 - p_n, the lowest-priority class, with allocation p_n, could be denied bandwidth for at most

$$\sum_{i=1}^{n-1} \frac{p_i}{1 - \sum_{j=1}^{i-1} p_j} T$$

seconds, where T seconds is the interval over which bandwidth usage is measured.

Conclusions:

- We believe that needs met by link-sharing are fundamental, given the presence of congestion. We do not believe that the need for controlled link-sharing is a transient stage that will disappear with the full commercialization of the Internet.

- It is not easy to fully anticipate the service requirements of emerging real-time applications on the Internet. We believe that mechanisms for controlled link-sharing add flexibility for satisfying requirements of emerging real-time applications.