Week 9: Quality of Service – 2nd Half

Tom asked the question: What would be the ratio of real to non-real time applications 10 or 20 years from now?

The guess is that almost all traffic will probably be real time then. The solution for that might be fixed circuits not carried over IP. The internet, however, still has a cost advantage to provide real-time applications because of equipment volume.

If data has nice statistical properties, we can measure utilization on links and do prediction. Reservation (weak guarantees) could then be made with proper capacity planning.

However, we have the self-similarity nature of internet data which might pose some problems to this approach (or at least not as clear as in telephony capacity planning). This is compounded by the multitude of real-time applications that need to be supported (just one for telephony, essentially).

Circuits: Sequence of state words through the network, effectively reserving bandwidth in each network element.

Packets: What does it mean to have a mechanism for reservation?
RSVP: Receiver driven.
Question: Is it OK to drop the connection? In a circuit you cannot recover (telephony switches are rated 5 9's, but for end-to-end one needs to do the convolution product of switches and links to figure out the availability, most AT&T connections involve 2 or at most 3 switches, though, in the long-distance portion).

Digression: Evolution of OS reliability. Rule of thumb: It takes about 10 years of OS life to get it reliable. (Unix over 25 years, Windows about 10 years becoming reliable now).

Question: If another route fails, would it affect my route? (routes are not exactly independent, the problem in one might affect reliability or degradation due to increase traffic in a second route).

The objective is to be able to re-route if a path fails (source routing with reservations). Packet switches should be capable of this.

RSVP: It is based on the notion of soft state (refresh before it times out). This maintains the robustness of the internet. Receiver rather than sender oriented (as it is in connection-oriented networks).

Question: When all is said and done and, assuming traffic is largely going real-time, would it still be cheaper to use the internet versus circuit switching (a la telephony today).
Answer: Yes, volume is the driver: A 2X increase in volume (of routers) translates into a

10% reduction in cost. (AT&T network only about 100 4ESS switches nationwide, worth millions of dollars each, versus 1 million Cisco routers today).

The same rule of thumb (2X volume = 10% off cost) can be used to explain x86 processors over any other, clusters of PCs over MPs, small cheap disk arrays (RAID) over large single disks.

Going back to rerouting, packet switching should have advantages for this over circuit switching: We should not need to setup a new connection (as in a circuit switched network).

We discussed a bit scheduling (manage of packets in queues) and classifying (associate the packet with the appropriate reservation priority).

We discussed weighted RED (RIO) which implements per-hop behavior (PHB) such that the drop probability curve favors the high priority traffic by setting in drop at a higher queue length, eventually reaching the best effort curve for long queues.

Weighted fair queueing was also discussed. This approach is essentially a round-robin over time slices that guarantee a percentage of the service time will go to priority packets. This approach is equivalent to a virtual clock for the priority traffic. The scalability of the approach is proportional to the number of queues (i.e. the number of ways the available serving time needs to be divided). The number of queues should be less or equal than the number of flows (we should be able to assign multiple flows to a queue). We save on the number of queues if we can implement it at the edge of the network.

No system currently reserves resources across ISPs.