Congestion Control and QoS

CS158a
Chris Pollett
Apr 11, 2007.

Outline

- Congestion Control Algorithms
- Quality of Service

Congestion Prevention Policies

- We briefly point out some policy issues in each layer which can affect congestion.
- The choice is like an open loop decision which can increase or reduce congestion.

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Congestion Control in Virtual Circuit Subnets

- One way to prevent congestion in Virtual Circuits is to use **admission control.**
- That is, one congestion has been signaled, no more virtual circuits are set up until the problem goes away.
- This is actually what was done in the original telephone system.
- Another technique is to use a reservation system when setting up a connection. Here a connection cannot be set up in the first place unless a certain quality of service can be guaranteed.

Congestion Control in Datagram Subnets

- In datagram subnets, each router can monitor the utilization of its output lines.
- This is usually represented as a float between 0 and 1.
- When this value exceeds a threshold, it enters a warning state. This was sometimes indicated (DECnet, frame relay) by putting a warning bit in acknowledgement packets.
- Another approach is to send a **choke packet** back to the source host directly to tell it to slow down
- When a source gets a choke packet it is supposed to slow down its transmission rate to a given destination by X percent, or alternatively the window size can be changed.
- Over long distances, choke packets do not work so well, so instead a variant called **hop-by-choke** is used. Here the choke packet has its effect on each router it passes through.

Load Shedding

- If the previous techniques don't work another, technique to reduce congestion is simply to discard packets.
- This is called **load shedding**.
- Which packets are discarded is usually based on their priority.
- In virtual circuits it the source sends more than it said it would all of its excess packets are marked as low priority.
- In the datagram setting since a given router might not be able to determine which source is causing the congestion, one technique is to pick random packets in the queue and drop them before congestion gets too bad.
- Usually, this will mean the source that was causing the congestion will get one of its packets dropped and so will throttle back.
- This technique is called **RED(Random Early Detection).**

Jitter Control

- For streaming applications (sound, video) what is important is the variation in packet arrival times, not the total transit time.
- This variation is called **jitter**.
- Jitter can sometimes be eliminated by just buffering at the destination.
- However, in some situations such as telephony, you need real-time not buffered signals.

Quality of Service

- The above techniques are rather ad-hoc.
- We are interested in techniques which guarantee a certain rate of traffic from the source to the destination.
- A stream of packets from the source to the destination is called a **flow**.
- Four parameters characterize the requirements of a flow: reliability, delay, jitter, bandwidth.
- Here are some example flow situations:

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Techniques

- **Overprovisioning** -- make sure each router has so much capacity, buffer space,etc that packets can easily get to their destinations (expensive)
- **Buffering** -- this can be used to smooth out jitter,
- **Traffic Shaping** -- here the source sends out packets with a uniform spacing between them rather than as fast as it can.
- **Leaky Bucket Algorithm** -- in this algorithm the host is allowed to put one packet per clock tick onto the network. A host maintains a queue of packets it wants to send, if this queue exceeds a certain length, the excess packets are discarded. For variable length packets typically have a fixed number of bytes that can be sent each clock tick instead. There are also token based variants of this algorithm.
- **Resource Reservation --** this is the admission control procedure discussed briefly earlier
- **Proportional Routing** -- rather than send packets only along the best path to the route, split the traffic over multiple routes depending in proportion to the bandwidth they can handle
- **Packet Scheduling --** queue packets on outputs lines according to which host they came from. (**fair queueing).**

Integrated services

- The IETF developed a generic framework for an streaming multimedia between 1995 and 1997 known as **flow-based algorithms** or integrated services.
- It was aimed at both unicast and multicast situations.
- The main protocol in this framework is **RSVP** (**resource reservation protocol**). This uses multicast routing using spanning trees.
- Besides RSVP, the framework also includes protocols for **differentiated service** based on classes., the simplest being **expedited forwarding** in which there are two types of packets regular and expedited. There is also a four level scheme called **assured forwarding.**

Label Switching

- Besides the IETF, several router companies worked on how to do forwarding better.
- One idea was to put a label in front of each packet and doing the routing based on the label rather than the destination address (similar to a virtual circuit).
- The technique goes under the name of **label switching** or **tag switching** and was standardized under the name **MPLS** (**multiprotocol label switching**).
- One difference from virtual circuits is that packets from several sources might end up with the same label.