Congestion Control

When one part of the subnet (e.g. one or more routers in an area) becomes overloaded, *congestion* results. Because routers are receiving packets faster than they can forward them, one of two things must happen:

1. The subnet must prevent additional packets from entering the congested region until those already present can be processed.
2. The congested routers can discard queued packets to make room for those that are arriving.

See Fig. 5-26 for possible sources.

We now consider the problem of congestion and some possible solutions.

Three general approaches:

1. prevent it altogether
2. congestion avoidance
3. deal with it if it occurs

Preallocation of Resources

Preallocation schemes aim to *prevent* congestion from happening in the first place. For example, we can require that resources be preallocated *before* any packets can be sent, guaranteeing that resources will be available to process each packet.

In virtual circuit networks, for example, the sender opens a connection before sending data. The circuit setup operation selects a path through the subnet, and each router on the path dedicates buffer space and bandwidth to the new circuit.

What happens when a user attempts to open a virtual circuit and the subnet is congested? The subnet can refuse to open the connection, forcing the user to wait until sufficient resources become available.

Note: The ability of the subnet to reject requests to open connections is an important property of connection oriented networks.

Traffic Shaping

Control the *rate* at which packets are sent (not just how many).

A congestion avoidance method.

Relates to Quality of Service (QoS) as means to provide more reliable service despite variable transmission patterns.

At set up, the sender and carrier negotiate a traffic pattern (shape).

Leaky Bucket Algorithm used to control rate in a datagram network. See Fig. 5-24. A single-server queue with constant service time. If bucket (buffer) overflows then packets are discarded.

Enforces a constant output rate regardless of burstiness of input. Does nothing when input is idle. In contrast the *Token Bucket Algorithm* causes a token to be generated periodically, which during idle periods can be saved up.

Related to traffic shaping is *flow specification*, where a particular quality of service is agreed upon between sender, receiver and carrier.

Isarithmic Congestion Control

Another approach to congestion avoidance is to limit the total number flow of packets in the subnet at any one time. The idea is similar to the token ring:

1. When an router accepts a packet from a host, it must obtain a *permit* before sending the packet into the subnet.
2. Obtaining a permit is analogous to "seizing the token", but there can be many permits in the subnet. When an router obtains a permit, it destroys it.
3. The destination router regenerates the permit when it passes the packet to the destination host.

Issues:

1. Although we have limited the total number of packets in the subnet, we have no control over *where* in the subnet those packets will be. Thus, an router in one part of the subnet might be congested, while an router in another part remains idle, but unable to process packets for lack of permits.
2. Regenerating lost permits is difficult, because no single node knows how many permits are currently in the subnet.
3. How to distribute permits? Distribute among the routers or centralize for a known access point.

Virtual Circuits Admission Control

Refuse to set up new connections if congestion is present. Also helps with congestion avoidance.

Flow Control

Flow control is aimed at preventing a fast sender from overwhelming a slow receiver. Flow control can be helpful at reducing congestion, but it can't really solve the congestion problem.

For example, suppose we connect a fast sender and fast receiver using a 9.6 kbps line:

1. If the two machines use a sliding window protocol, and the window is large, the link will become congested in a hurry.
2. If the window size is small (e.g., 2 packets), the link won't become congested.

   Note how the window size limits the total number of packets that can be in transmission at one time.

Flow control can take place at many levels:

- User process to user process (end-to-end). Later, we'll see how TCP uses flow control at the end-to-end level.
- Host to host. For example, if multiple application connections share a single virtual circuit between two hosts.
- router to router. For example, in virtual circuits.

Load Shedding/Discarding Packets (No Preallocation)

At the other end of the spectrum, we could preallocate no resources in advance, and take our chances that resources will be available when we need them. When insufficient resources are present to process existing packets, discard queued packets to make room for newly arriving ones.

Who retransmits the discarded packets? Two cases: connection oriented and connectionless. In datagram (connectionless) networks, the sending host (transport layer) retransmits discarded packets (if appropriate). In virtual circuit networks, the previous-hop router retransmits the packet when it fails to receive an acknowledgment.

Failure to preallocate resources leads to two problems: potential *deadlock* and *unfairness*. First, let us consider deadlock.

Suppose that all of an router's buffers hold packets. Because the router has no free buffers, it cannot accept additional frames. Unfortunately, it also ignores frames containing ACKs that would free up some of those buffers!

Suppose further, that two adjacent routers, A and B, are sending packets to each other. Since both are waiting for the other to accept a packet, neither can proceed. This condition is known as a *deadlock*.

Solution: Reserve at least one buffer for each input line and use it to hold incoming packets. Note that we can extract the ACK field and still discard the packet, if we don't have buffers to hold it.

Advantage of discarding packets when congested: Easy to implement.

Disadvantages:

1. Wastes resources. The network may have expended considerable resources processing a packet that is eventually discarded.
2. Non-deterministic. There is less guarantee than with virtual circuits that packets will ever reach their destination.
3. Requires that sending hosts pay attention to congestion. If the network can't prevent a host from sending data, a host can overload the network. In particular, a "broken" host may cause the network to become overly congested.
4. In the extreme case, *congestion collapse* occurs. The network becomes so overloaded, that few packets reach their destination. Meanwhile, the sending hosts continue to generate more data (both retransmissions and new packets). This condition occurred several times back in 1987, and the Internet/Arpanet became unusable for a period of hours to days.

Random Early Detection (RED)

Start dropping packets before a router runs out of buffer space. Basic idea:

```
     ^
  1.0|         --------
     |         |
     |         |
 maxp|         |
     |        /
drop |       /
prob.|      /
      -----|---|-------------------------->
           min max
               queue size
```

Choke Packets

ECN (Explicit Congestion Notification) is an example where instead of dropping packets, routers mark packets indicating that router is congested.

routers can monitor the level of congestion around them, and when congestion is present, they can send *choke packets* to the sender that say “slow down”.

How can an router measure congestion? An router might estimate the level of congestion by measuring the percentage of buffers in use, line utilization, or average queue lengths.

Advantage: Dynamic. Host sends as much data as it wants, the network informs it when it is sending too much.

Disadvantage:

1. Difficult to tune. By how much should a host slow down? The answer depends on how much traffic the host is sending, how much of the congestion it is responsible for, and the total capacity of the congested region. Such information is not readily available in practice.
2. After receiving a choke packet, the sending host should ignore additional choke packets for a short while because packets currently in transmission may generate additional choke packets. How long? Depends on such dynamic network conditions as delay.

Variations exist.

Congestion Summary

Overall: Varying methods for congestion control with different levels of effectiveness.

More attention being paid to reserving resources so that chances of congestion are reduced and the quality of service is more reliable.

Will also look at congestion control again when we examine TCP as it has its own congestion control mechanism.