Chapter 13 – Congestion in Data Networks

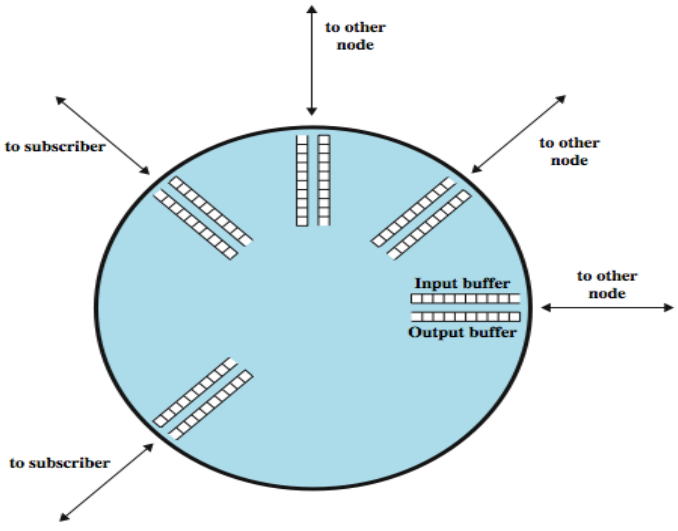
by William Stallings

What is Congestion?

Congestion occurs when the number of packets being transmitted through a network begins to approach the packet-handling capacity of the network. The objective of congestion control is to maintain the number of packets within the network below the level at which performance falls off dramatically.

A data network or internet is a network of queues. At each node (data network switch, internet router), there is a queue of packets for each outgoing channel. If the rate at which packets arrive and queue up exceeds the rate at which packets can be transmitted, the queue size grows without bound and the delay experienced by a packet goes to infinity. Even if the packet arrival rate is less than the packet transmission rate, queue length will grow dramatically as the arrival rate approaches the transmission rate. As a rule of thumb, when the line for which packets are queuing becomes more than 80% utilized, the queue length grows at an alarming rate. This growth in queue length means that the delay experienced by a packet at each node increases. Further, since the size of any queue is finite, as queue length grows, eventually the queue must overflow.

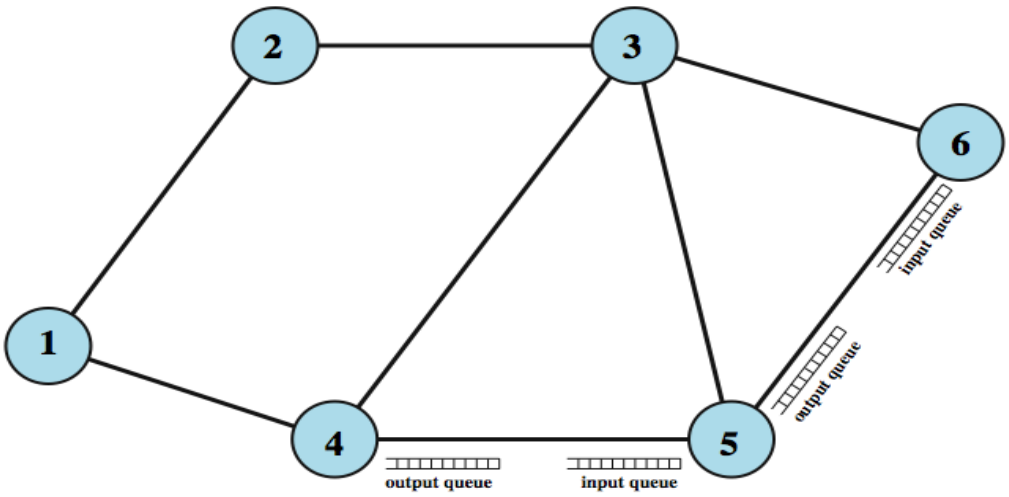
Queues at the Node



Consider the queuing situation at a single packet switch or router, such as is illustrated in Stallings DCC8e Figure 13.1. Any given node has a number of I/O ports attached to it: one or more to other nodes, and zero or more to end systems. On each port, packets arrive and depart. We can consider that there are two buffers, or queues, at each port, one to accept arriving packets, and one to hold packets that are waiting to depart. In practice, there might be two fixed-size buffers associated with each port, or there might be a pool of memory available for all buffering activities. In the latter case, we can think of each port having two variable-size buffers associated with it, subject to the constraint that the sum of all buffer sizes is a constant.

In any case, as packets arrive, they are stored in the input buffer of the corresponding port. The node examines each incoming packet, makes a routing decision, and then moves the packet to the appropriate output buffer. Packets queued for output are transmitted as rapidly as possible; this is, in effect, statistical time division multiplexing. If packets arrive too fast for the node to process them (make routing decisions) or faster than packets can be cleared from the outgoing buffers, then eventually packets will arrive for which no memory is available.

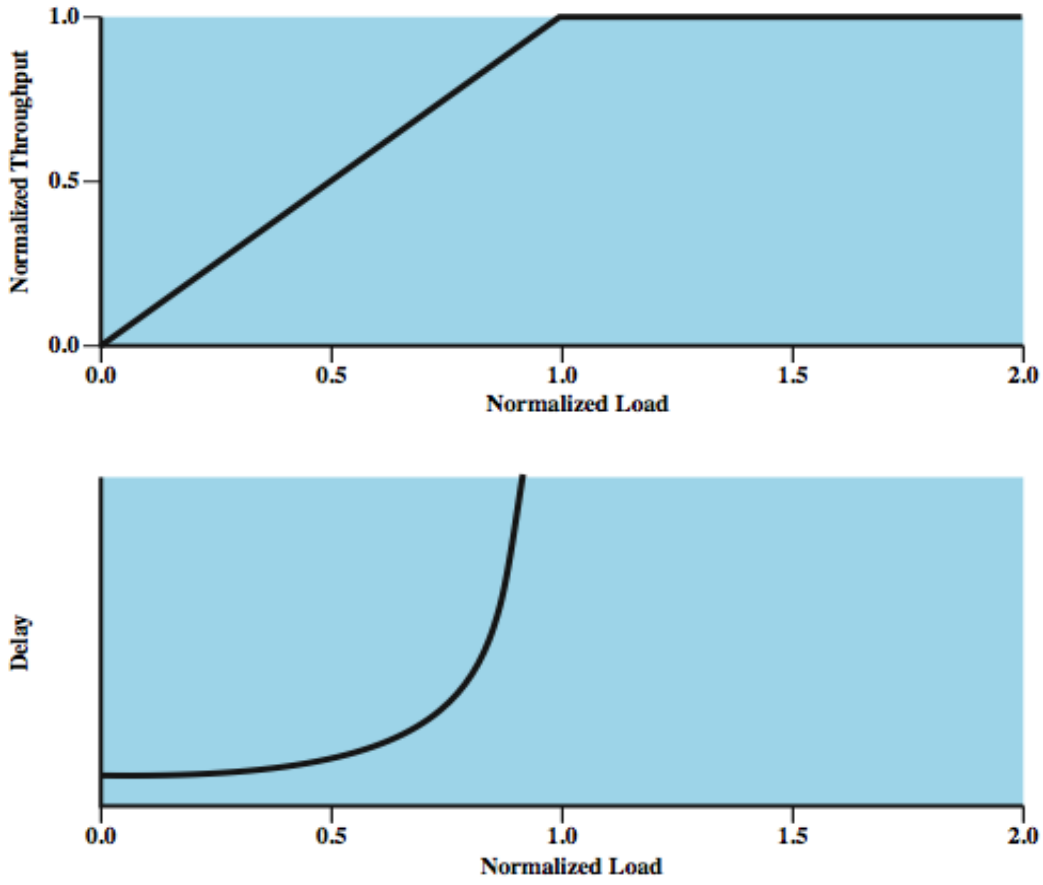
Interaction of queues



When such a saturation point is reached, one of two general strategies can be adopted. The first such strategy is to discard any incoming packet for which there is no available buffer space. The alternative is for the node that is experiencing these problems to exercise some sort of flow control over its neighbors so that the traffic flow remains manageable. But, as Stallings DCC8e Figure 13.2 illustrates, each of a node's neighbors is also managing a number of queues. If node 6 restrains the flow of packets from node 5, this causes the output buffer in node 5 for the port to node 6 to fill up. Thus, congestion at

one point in the network can quickly propagate throughout a region or the entire network. While flow control is indeed a powerful tool, we need to use it in such a way as to manage the traffic on the entire network.

The ideal goal for network utilization



The above figure suggests the ideal goal for network utilization. The top graph plots the steady-state total throughput (number of packets delivered to destination end systems) through the network as a function of the offered load (number of packets transmitted by source end systems), both normalized to the maximum theoretical throughput of the network. For example, if a network consists of a single node with two full-duplex 1-Mbps links, then the theoretical capacity of the network is 2 Mbps, consisting of a 1-Mbps flow in each direction. In the ideal case, the throughput of the network increases to accommodate load up to an offered load equal to the full capacity of the network; then normalized throughput remains at 1.0 at higher input loads. Note, however, what happens to the end-to-end delay experienced by the average packet even with this assumption of ideal performance. At negligible load,

there is some small constant amount of delay. As the load on the network increases, queuing delays at each node are added to this fixed amount of delay. When the load exceeds the network capacity, delays increase without bound. Suppose that each node in the network is equipped with buffers of infinite size and suppose that the input load exceeds network capacity. Under ideal conditions, the network will continue to sustain a normalized throughput of 1.0. Therefore, the rate of packets leaving the network is 1.0. Because the rate of packets entering the network is greater than 1.0, internal queue sizes grow. In the steady state, with input greater than output, these queue sizes grow without bound and therefore queuing delays grow without bound. This figure represents the ideal, but unattainable, goal. No scheme can exceed its performance.

Mechanisms for Congestion Control

In this book, we discuss various techniques for controlling congestion in packet-switching, frame relay, and ATM networks, and in IP-based internets. To give context to this discussion, Stallings DCC8e Figure 13.5 provides a general depiction of important congestion control techniques, which include:

- backpressure
- choke packets
- implicit congestion signaling
- explicit congestion signaling

Backpressure

Backpressure is a technique for congestion control, similar to backpressure in fluids flowing down a pipe. When the end of a pipe is closed (or restricted), the fluid pressure backs up the pipe to the point of origin, where the flow is stopped (or slowed). Backpressure can be exerted on the basis of links or logical connections (e.g., virtual circuits). Referring again to Stallings DCC8e Figure 13.2, if node 6 becomes congested (buffers fill up), then node 6 can slow down or halt the flow of all packets from node 5 (or node 3, or both nodes 5 and 3). If this restriction persists, node 5 will need to slow down or halt traffic on its incoming links. This flow restriction propagates backward (against the flow of data traffic) to sources, which are restricted in the flow of new packets into the network. Backpressure can be selectively applied to logical connections, so that the flow from one node to the next is only restricted or halted on some connections, generally the ones with the most traffic. In this case, the restriction propagates back along the connection to the source.

Backpressure is of limited utility. It can be used in a connection-oriented network that allows hop-by-hop (from one node to the next) flow control. X.25-based packet-switching networks typically provide this feature. However, neither frame relay nor ATM has any capability for restricting flow on a hop-by-hop basis. In the case of IP-based internets, there have traditionally been no built-in facilities for regulating the flow of data from one router to the next along a path through the internet.

Choke Packet

A choke packet is a control packet generated at a congested node and transmitted back to a source node to restrict traffic flow. An example of a choke packet is the ICMP (Internet Control Message Protocol) Source Quench packet. Either a router or a destination end system may send this message to a source end system, requesting that it reduce the rate at which it is sending traffic to the internet destination. On receipt of a source quench message, the source host should cut back the rate at which it is sending traffic to the specified destination until it no longer receives source quench messages. The source quench message can be used by a router or host that must discard IP datagrams because of a full buffer. In that case, the router or host will issue a source quench message for every datagram that it discards. In addition, a system may anticipate congestion and issue source quench messages when its buffers approach capacity. In that case, the datagram referred to in the source quench message may well be delivered. Thus, receipt of a source quench message does not imply delivery or nondelivery of the corresponding datagram. The choke package is a relatively crude technique for controlling congestion. More sophisticated forms of explicit congestion signaling are discussed subsequently.

Implicit Congestion Signaling

When network congestion occurs, two things may happen: (1) The transmission delay for an individual packet from source to destination increases, so that it is noticeably longer than the fixed propagation delay, and (2) packets are discarded. If a source is able to detect increased delays and packet discards, then it has implicit evidence of network congestion. If all sources can detect congestion and, in response, reduce flow on the basis of congestion, then the network congestion will be relieved. Thus, congestion control on the basis of implicit signaling is the responsibility of end systems and does not require action on the part of network nodes. Implicit signaling is an effective congestion control technique in connectionless, or datagram, configurations, such as datagram packet-switching networks and IP-based internets. In such cases, there are no logical connections through the internet on which flow can be regulated. However, between the two end systems, logical connections can be established at the TCP level. TCP includes mechanisms for acknowledging receipt of TCP segments and for regulating the flow of data between source and destination on a TCP connection. TCP congestion control techniques based on the ability to detect increased delay and segment loss are discussed in Stallings DCC8eChapter 20. Implicit signaling can also be used in connection-oriented networks. For example, in frame relay networks, the LAPF control protocol, which is end to end, includes facilities similar to those of TCP for flow and error control. LAPF control is capable of detecting lost frames and adjusting the flow of data accordingly.

Explicit Congestion Signaling

It is desirable to use as much of the available capacity in a network as possible but still react to congestion in a controlled and fair manner. This is the purpose of explicit congestion avoidance techniques, where the network alerts end systems to growing congestion within the network and the

end systems take steps to reduce the offered load to the network. Typically, explicit congestion control techniques operate over connection-oriented networks and control the flow of packets over individual connections. Explicit congestion signaling approaches can work in one of two directions:

• **Backward:** Notifies the source that congestion avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received notification. It indicates that the packets that the user transmits on this logical connection may encounter congested resources. Backward information is transmitted either by altering bits in a header of a data packet headed for the source to be controlled or by transmitting separate control packets to the source.

• **Forward:** Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the same direction as the received notification. It indicates that this packet, on this logical connection, has encountered congested resources. Again, this information may be transmitted either as altered bits in data packets or in separate control packets. In some schemes, when a forward signal is received by an end system, it echoes the signal back along the logical connection to the source. In other schemes, the end system is expected to exercise flow control upon the source end system at a higher layer (e.g., TCP).

Explicit Signaling Categories

We can divide explicit congestion signaling approaches into three general categories:

• **Binary:** A bit is set in a data packet as it is forwarded by the congested node. When a source receives a binary indication of congestion on a logical connection, it may reduce its traffic flow.

• **Credit based:** These schemes are based on providing an explicit credit to a source over a logical connection. The credit indicates how many octets or how many packets the source may transmit. When the credit is exhausted, the source must await additional credit before sending additional data. Credit-based schemes are common for end-to-end flow control, in which a destination system uses credit to prevent the source from overflowing the destination buffers, but credit-based schemes have also been considered for congestion control.

• **Rate based:** These schemes are based on providing an explicit data rate limit to the source over a logical connection. The source may transmit data at a rate up to the set limit. To control congestion, any node along the path of the connection can reduce the data rate limit in a control message to the source.

Traffic Management

There are a number of issues related to congestion control that might be included under the general category of traffic management. When a node is saturated and must discard packets, it can apply some simple rule, such as discard the most recent arrival. However, other considerations can be used to refine the application of congestion control techniques and discard policy.

Fairness ensurse that in the absence of other requirements, the various flows suffer from congestion equally. Simply to discard on a last-in-first-discarded basis may not be fair. For example, a node can maintain a separate queue for each logical connection or for each source-destination pair. If all of the queue buffers are of equal length, then the queues with the highest traffic load will suffer discards more often, allowing lower-traffic connections a fair share of the capacity.

It is important during periods of congestion that traffic flows with different requirements be treated differently and provided a different quality of service (QoS). For example, a node might transmit higher-priority packets ahead of lower-priority packets in the same queue. Or a node might maintain different queues for different QoS levels and give preferential treatment to the higher levels.

One way to avoid congestion and also to provide assured service to applications is to use a reservation scheme. Such a scheme is an integral part of ATM networks. The network agrees to give a defined QoS so long as the traffic flow is within contract parameters; excess traffic is either discarded or handled on a best-effort basis, subject to discard.

Congestion Control in Packet Switched Networks

A number of control mechanisms for congestion control in packet-switching networks have been suggested and tried. The following are examples:

1. Send a control packet from a congested node to some or all source nodes. This choke packet will have the effect of stopping or slowing the rate of transmission from sources and hence limit the total number of packets in the network. This approach requires additional traffic on the network during a period of congestion.

2. Rely on routing information. Routing algorithms, such as ARPANET's, provide link delay information to other nodes, which influences routing decisions. This information could also be used to influence the rate at which new packets are produced. Because these delays are being influenced by the routing decision, they may vary too rapidly to be used effectively for congestion control.

3. Make use of an end-to-end probe packet. Such a packet could be timestamped to measure the delay between two particular endpoints. This has the disadvantage of adding overhead to the network.

4. Allow packet-switching nodes to add congestion information to packets as they go by. There are two possible approaches here. A node could add such information to packets going in the direction opposite of the congestion. This information quickly reaches the source node, which can reduce the flow of packets into the network. Alternatively, a node could add such information to packets going in the same direction as the congestion. The destination either asks the source to adjust the load or returns the signal back to the source in the packets (or acknowledgments) going in the reverse direction.

Frame Relay Congestion Control

I.370 defines the objectives for frame relay congestion control to be the following:

• Minimize frame discard.

• Maintain, with high probability and minimum variance, an agreed quality of service.

• Minimize the possibility that one end user can monopolize network resources at the expense of other end users.

• Be simple to implement, and place little overhead on either end user or network.

• Create minimal additional network traffic.

• Distribute network resources fairly among end users.

• Limit spread of congestion to other networks and elements within the network.

• Operate effectively regardless of the traffic flow in either direction between end users.

• Have minimum interaction or impact on other systems in the frame relaying network.

• Minimize the variance in quality of service delivered to individual frame relay connections during congestion (e.g., individual logical connections should not experience sudden degradation when congestion approaches or has occurred).

FR Congestion Control

Table 13.1 Frame Relay Congestion Control Techniques

Technique	Type	Function	Key Elements
Discard control	Discard strategy	Provides guidance to network concerning which frames to discard	DE bit
Backward explicit Congestion Notification	Congestion avoidance	Provides guidance to end systems about congestion in network	BECN bit or CLLM message
Forward explicit Congestion Notification	Congestion avoidance	Provides guidance to end systems about congestion in network	FECN bit
Implicit congestion notification	Congestion recovery	End system infers congestion from frame loss	Sequence numbers in higher-layer PDU

Congestion control is difficult for a frame relay network because its protocol has been streamlined to maximize throughput and efficiency. Congestion control is the joint responsibility of the network and the end users. The network is in the best position to monitor the degree of congestion, while the end users are in the best position to control congestion by limiting the flow of traffic. Some congestion control techniques defined in the various ITU-T and ANSI documents are:

Discard strategy deals with the most fundamental response to congestion: When congestion becomes severe enough, the network is forced to discard frames.

Congestion avoidance procedures are used at the onset of congestion to minimize the effect on the network. These procedures would be initiated at or prior to point A in Figure 13.4 to prevent congestion from progressing to point B. Near point A there would be little evidence available to end users that congestion is increasing. This needs some **explicit signaling** mechanism from the network to trigger congestion avoidance.

Congestion recovery procedures are used to prevent network collapse in the face of severe congestion. These procedures are typically initiated when the network has begun to drop frames due to congestion. Such dropped frames will be reported by some higher layer of software (eg., LAPF or TCP) and serve as an **implicit signaling** mechanism. Congestion recovery procedures operate around point B and within the region of severe congestion, as shown in Stallings DCC8e Figure 13.4.(please refer the book)