UNIT 3 CONGESTION CONTROL IN PUBLIC SWITCHED NETWORK

Structure **Page Nos.**

3.0 INTRODUCTION

Congestion occurs when the number of packets being transmitted through the public switched networks approaches the packet handling capacity of the network. When the number of packets dumped into the subnet by the hosts, is within its carrying capacity, all packets are delivered (except for a few that are afflicted with transmission errors). The networks get overloaded and start dropping the packet, which leads to congestion in the network. Due to the dropping of packets, the destination machine may demand the sources to retransmit the packet, which will result in more packets in the network and this leads to further congestion. The net result is that the throughput of the network will be very low as, illustrated in the *Figure 1*.

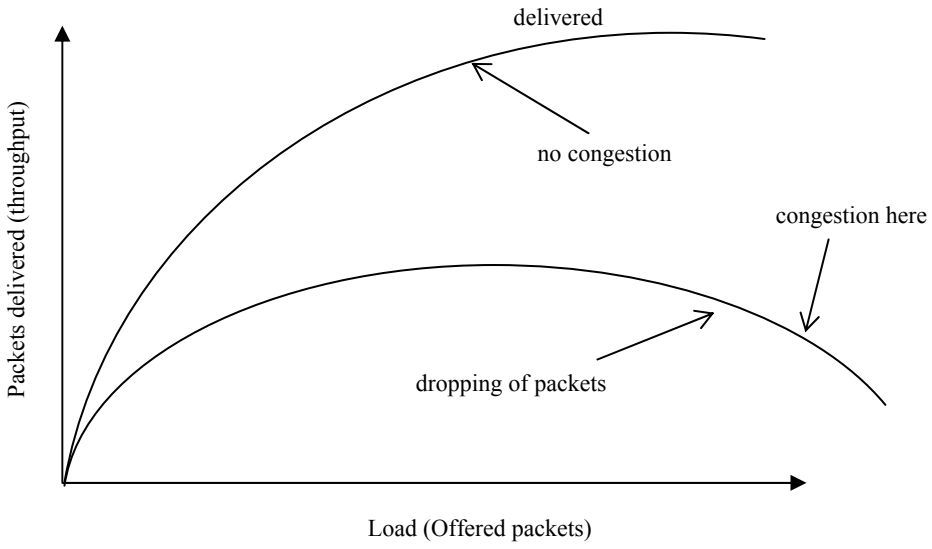


Figure 1: Congestion in network

3.1 OBJECTIVES

After going through this unit, you should be able to:

- define congestion;
- list the factors for the occurrence of congestion in the Network;
- differentiate between congestion control and flow control;
- outline the general principles of congestion Control, and
- discuss congestion prevention mechanism.

3.2 REASONS FOR CONGESTION IN THE NETWORK

Tanenbaum [*Ref 1*] has mentioned several reasons for the occurrence of congestion in the network:

1) Sudden arrival of a packet on a particular output line from multiple input source, leading to the formation of a queue and dropping of a packet in case of an insufficient memory in a router to hold these packets. **Nagle** has pointed out that, even if, there is a huge memory at the router level, there is no improvement in congestion, rather, it get worse because of time out of the packet.

2) Slow processors can also cause congestion.

3) Low bandwidth channels can also cause congestion. If, we increase the bandwidth without a fast processor or vice-versa, will be of little help.

3.3 CONGESTION CONTROL Vs FLOW CONTROL

The differences between congestion and flow control are whether mentioning as they are related terms. There is another term, related to these two, called **error control. Congestion Control** ensures that the subnet is able to carry the offered traffic and that it is not overloaded. **Flow control** has to do with ensuring that the fast sender does not overload packets from the noisy channels receiver. **Error control** deals with algorithms that recover or conceal the effects from packet losses. The goal of each of these control mechanisms are different, but, the implementation can be combined.

The following *Table* differentiates between Congestion Control and Flow Control:

Table 1: Congestion Control vs. Flow Control

Congestion Control	Flow Control
Congestion Control is needed when buffers in packet switches overflow or cause congestion.	Flow Control is needed when, the buffers at the receiver are not depleted as fast as the data arrives.
Congestion is end-to-end, it includes all hosts, links and routers. It is a global issue.	Flow is between one data sender and one receiver. It can be done on link-to-link or end-to-end basis. It is a local issue.
If viewed as a "congested buffer", then the switch tells the source of the data stream to slow down using congestion control notifications. When output buffers at a switch fill up and packets are dropped this leads to congestion control actions.	If there is a queue on the input side of a switch, and link-by-link flow control is used, then as a "flow control" action the switch tells its immediate neighbour to slow down if the input queue fills up.

The reason for comparing congestion and flow control is that, some congestion control algorithms operate by sending messages back to the senders to slow down incase, the network is congested. In case, the receiver is overloaded, the host will get a similar message to slow down.

☞ Check Your Progress 1

1) Differentiate between Congestion Control and Flow Control.

……

……

……

2) What are the differences between open loop and closed loop solutions to congestion?

……

……

……

3.4 CONGESTION PREVENTION MECHANISM

The purpose of this section is to examine various mechanisms used at the different layers to achieve different goals. But, from the congestion control point of view, these mechanism are not very effective. We will start with data link layer first. **Go back N** and Selection Repeat are flow control mechanism available at the data link layer. Incase, there is any error with the packet, Go back N retransmits all the packets up to the packet where the error occurred. For example, there is an error in 5th packet, it means that it will retransmit 1, 2, 3, 4, and 5 which create extra load on the network, thereby, leading to congestion. Selective repeat retransmits only that packet. With respect to congestion control, selective repeat is clearly better than go back N.

The following table provides the detail:

Table 2: Mechanism Affecting Congestion

Layer	Mechanisms	Solution
Transport	• Sliding window with credit schemes • Piggybacking • Timeout determination	Similar to data link layer problem + Optimal time out determination
Network	• Bad Routing algorithm • Packet lifetime management	A good routing algorithm + Optimal lifetime manager
Data link layer	• Go back N, Selective Repeat • Piggybacking	Selective Repeat + Small window size.

Acknowledgement mechanism also affects congestion. If, each packet is acknowledged immediately then it will generate extra traffic. However, if acknowledgement is piggybacked onto reverse traffic along, extra timeouts and retransmissions may happen which will cause congestion. **A flow control scheme with a small window size reduces the data rate and thus, helps reduce congestion.**

At the network good routing algorithm can help avoid congestion by distributing the traffic over all the lines, whereas a bad one can send too much traffic over already congested lines. Finally, packet lifetime management deals with the duration a packet may live before being discarded. If, it is too long, lost packets may reside in the network a long time, but if, it is too short, packets may sometimes time out before reaching their destination, thus, inducing retransmissions. Therefore, we require good routing algorithm and optimal packet life time.

Between transport and data link layer there are common issues (Flow Control, acknowledge mechanism) with respect to Congestion Control mechanism but at the transport layer, the extra problem is related to determining time out internal across the network, which is a difficult problem.

3.5 GENERAL PRINCIPLES OF CONGESTION CONTROL

From our earlier discussions it appears that congestion problem is not very easy to solve. Typically, the solution depends on the type of requirements for the application (e.g., Q_o S, high bandwidth, fast processing). Like routing algorithm, congestion control algorithms have been classified into two types:

- Open Loop
- Closed Loop.

Open loop solutions attempt to solve the problem with a good design, that ensures that congestion does not occur in the network. Once having network is running midcourse corrections are not made. Open loop algorithm works on two basic mechanisms: i) **Admission Control** ii) **Resource Reservation**. Admission control is a function performed by a network to accept or reject traffic flow. Therefore, the purpose of open loop solution is to ensure that the traffic generated by the source will not lower the performance of network below the specified Q_o S. The network will accept traffic till Q_oS parameters are satisfied otherwise, it rejects the traffic.

In contrast, **closed loop** solutions are based on the concept of a feedback loop. These algorithms are called closed loop because the state of the network has to be fed up to the source that regulates traffic. Closed loop algorithms follow dynamic approach to the solution of congestion problems. It reacts during the congestion occurrence period or when the congestion is about to happen. A variety of metrices have been proposed to monitor the state of a subnet in order to observe congestion. These are:

- Queue length
- The number of retransmitted packets due to timeout
- Percentage of rejected packets due to shortage of the router's memory
- Average packet delay.

To prevent congestion from occurring, this mechanism monitors the system to detect when and where congestion occurs, pass is this information to places where action can be taken usually at the source and finally adjusts the systems operation to correct the problem.

The presence of congestion means that the offered load in the network is (temporarily) greater than the resources (routers) can handle. Two straight forward solutions are to **increase the resources** or **decrease the load.** To increase the resources the following mechanism may be used as suggested in ***Tanenbaum*** *[Ref 1]*.

i) Higher bandwidth may be achieved by increasing transmission power of a satellite.

ii) Splitting traffic on multiple routers instead of a single best rout.

iii) Use of temporary dial-up telephone line between certain points.

However, sometimes it is not possible to increase the capacity or it has already been increased to the limit. But, if the network has reached the maximum limit, the alternative is to reduce the load. The following mechanism may be used (i) Denying service to some users (ii) Degrading services to some or all users.

For subnets that use virtual circuits internally, these methods can be used at the network layer. In the next section, we will focus on their use in the network layer. We will also discuss the open loop control mechanism in detail. The closed loop mechanism will be discussed in Block 4 Unit 2 as a part of TCP Protocol.

3.6 OPEN LOOP CONTROL

As mentioned earlier Open Loop Control algorithm prevent the occurrence of from Congestion occurrence rather than dealing with it after it has occurred. It does not rely on feedback information to regulate the traffic. Thus, this technique is based on the assumption that once the packets are accepted from a particular source, the network will not get overloaded. In this section, we will examine several such techniques and its implementation. Learners are requested to **Leon Garcia's** book [*Ref. 2*].

3.6.1 Admission Control

This is a widely used technique in virtual circuit network. The technique is very simple and straight forward. Once congestion has occurred, no more virtual circuits are to be set up. This is similar to a telephone system in which there is no dial tone in case the source gets overloaded. When a source wants to establish a connection, admission control mechanism examine Q_0S parameters of the connections initiated by the source. If it is OK, the connection (VC) is established otherwise, it is rejected. In order to initiate a connection setup, a source specifies its traffic flow indicating a set of parameters called **traffic descriptor**, which includes **peak rate**, **average rate**, and **maximum traffic burst size**. (Maximum length of time the traffic is generated at the peak rate) and so on. Based on the characteristic of traffic flow, admission control mechanism reserves the bandwidth (which usually lies between peak rate and average rate).

3.6.2 Traffic Policing and its Implementation

Leon Garcia [*Ref 2*] has **defined traffic policing as the process of monitoring and enforcing the traffic flow of packets during the connection period**. The network may drop or mark the packet as noncompliant and give low priority to traffic in case, it does not obey the agreed upon parameters values during the initial connection setup phase. Most implementations of traffic policing is done through the **Leaky Bucket** algorithm. In this algorithm the bucket is considered as a network and traffic flow is considered as water being poured into a bucket. The following assumption are made:

- The bucket has certain depth to hold water just like a network can accept a certain number of packets.
- The bucket leaks at a certain rate (if there is water is the bucket) no matter at what rate water enters the bucket. In terms of computer networks, it should be interpreted as follows: No matter at what rate the packets arrive at the input lines of a routers, routers in a subnet passes to its outgoing link at a fixed rate.

- If the bucket does not overflow when the water is poured into the bucket, then the bucket of water is said to be **conforming.** In terms of network, if the traffic is within the agreed norms, all packets will be transferred.

- The bucket will spillover if, it is full and if, additional water is poured into it, if it gets more packets than it can handle, it will lead to congestion and then the network due to which the additional packets will be lost.

If, we expect the traffic flow to be very smooth, then the bucket has to be of a shallow type. In case, the flow is bursty in nature, the bucket should be deeper. In summary, what we want to observe is whether the outflow of packets corresponds to the arrival rate of packets or not? Implementation of a leaky bucket is similar to queue data structure implementation. When a packet arrives, if there is space left in the queue, it gets appended to the queue otherwise, it gets rejected.

3.6.3 Traffic Shaping and its Implementation

Leon Garcia [*Ref 2*] has defined **traffic shaping** as the process of altering traffic flow to another flow. It is mainly used for smoothening the traffic. Consider an example, where a host is generating data at 30 kbps, which, it can transmit to the network in several ways (as shown in the following *Figure 2*).

- It can transmit at the rate of 100 kbps for 0.3 Second
- It can transmit at the rate of 75 kbps for 0.4 Second

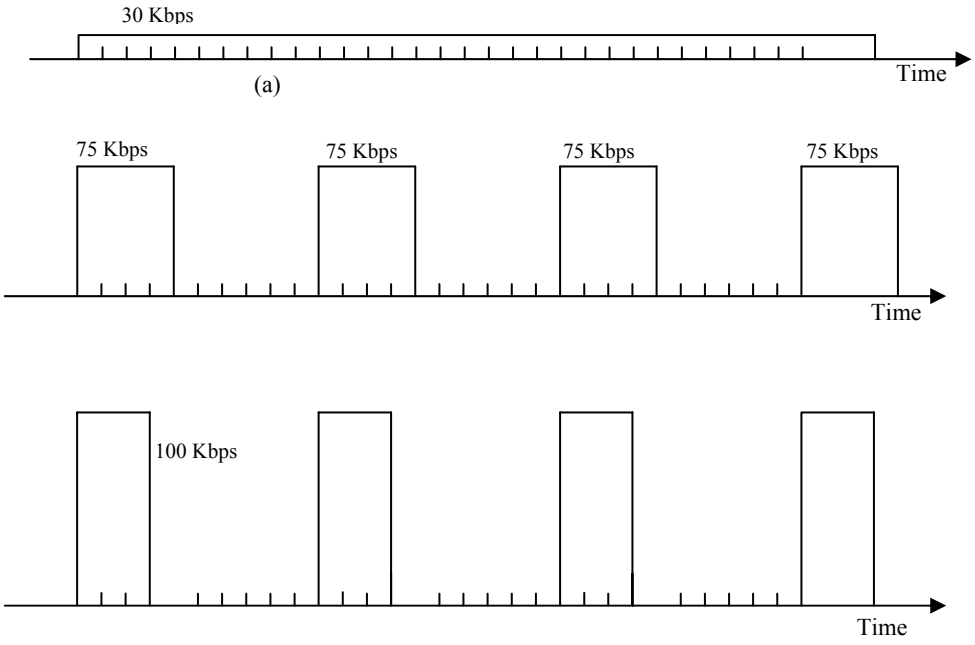


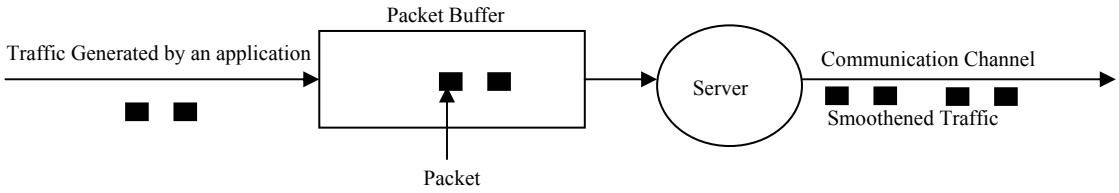
Figure 2: Possible traffic patterns at the average rate of 30 Kbps

You can make observation from the *Figure 2* that the *Figure 2 (a)* shows the smoothened pattern will create less stress on the network but the destination machine may not want to wait for 1 Second to retrieve 30 kbps data at each period. Now, we will look at its implementation. There are two mechanism:

- Leaky bucket traffic Shaper
- Token bucket traffic Shaper.

Leaky Bucket Traffic Shaper: It is a very simple mechanism in which data stored at the senders buffer, is passed on at a constant interval to smoothen traffic as shown

in *Figure 3*. The buffer is used to store bursty traffic. This size defines the maximum burst that can be accommodated.



Source: Ref.[2]

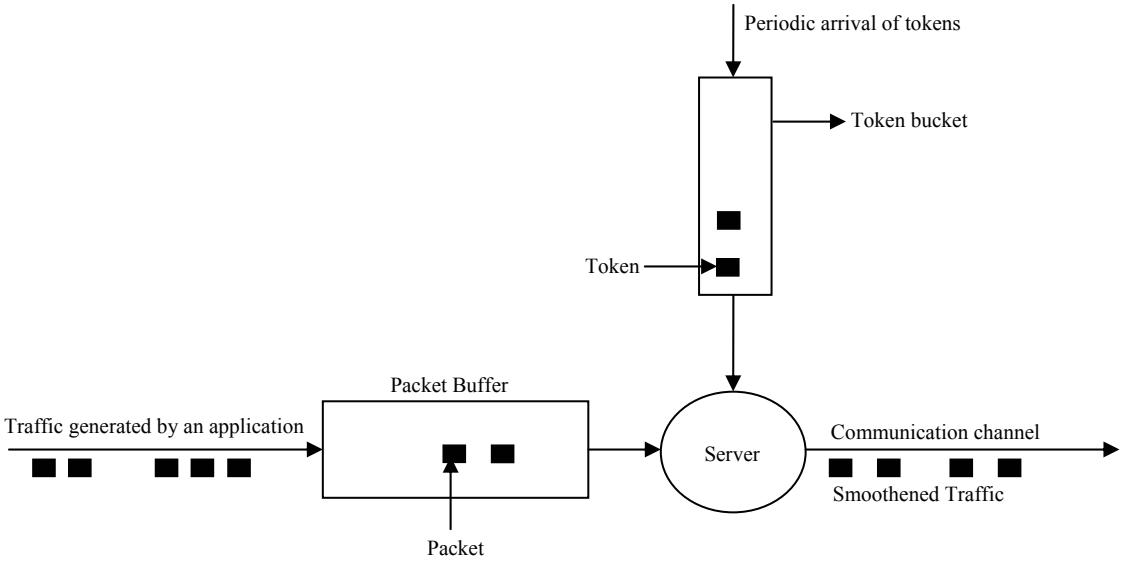
Figure 3: A leaky bucket traffic shaper

This mechanism is different from a leaky bucket algorithm which was used in traffic policing. The bucket in traffic policing is just a counter whereas, a bucket is traffic shaper is a buffer that stores the packets.

Token Bucket Traffic Shaper: The leaky bucket traffic shaper has a very restricted approach. Since, the output pattern is always constant no matter how bursty traffic is. Many applications produce variable rate traffic; sometimes bursty but sometimes normal. If, such traffic is allowed to pass through a leaky bucket traffic shaper, it may cause a very long delay. One such algorithms that deals with such situations is the **token bucket algorithm**.

The following are the features of token bucket traffic shaper:

- Token is used here, as a permit to send a packet. Unless there is a token, no packet can be transmitted.
- A Token bucket holds tokens which are generated periodically at a constant rate.
- New tokens are discarded, incase, the token buckets are full.
- A packet can be transmitted only if, there is a token in the token buffer. For example; in the *Figure 4* there are two tokens in the token buffer and five packets to be transmitted. Only two packets will be transmitted and the other three will wait for tokens.
- Traffic burstiness is proportional to the size of a token bucket.



Source: Ref.[2]

Figure 4: Token bucket traffic shaper

Now, let us discuss the operation of this algorithm;

Just assume that the token bucket is empty and the numbers of packets have arrived in the buffer. Since, there is no token in the token buffer, the packets have to wait until the new packet is generated. Since, tokens are generated periodically, the packet will be also transmitted periodically at the rate at which the tokens arrive. In, the next section, we will compare between the leaky bucket traffic shaper and token bucket traffic shaper. Students are also requested to see question 2 of Check Your Progress 2.

3.6.4 Difference Between Leaky Bucket Traffic Shaper and Token Bucket Traffic Shaper

The following tables differentiate between two types of traffic shapers:

Table 3: Leaky Bucket Traffic Shaper and Token Bucket Traffic Shaper

Leaky Bucket	Token Bucket
• Leaky Bucket (LB) discards packets. • With LB, a packet can be transmitted if the bucket is not full. • LB sends the packets at an average rate. • LB does not allow saving, a constant rate is maintained.	• Token Bucket (TB) discards tokens. • With TB, a packet can only be transmitted if, there are enough tokens to cover its length in bytes. • TB allows for large bursts to be sent faster by speeding up the output. • TB allows saving up of tokens (permissions) to send large bursts.

3.7 CONGESTION CONTROL IN PACKET-SWITCHED NETWORKS

Several control mechanism for Congestion Control in packet switched network have been explored and published. **William Stalling** [*Ref 3*] has presented the following mechanism to handle congestion:

1) Send a control packet (choke packet) from a node where congestion has occurred to some or all source nodes. This choke packet will have the effect of stopping or slowing the rate of transmission from sources and therefore, it will reduce total load on the network. This approach requires additional traffic on the network during the period of congestion.

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

3) 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 moving in the direction opposite to the congestion. This information reaches the source node quickly, and this reduces the flow of packets into the network. Alternatively, a node could add such information to packets moving in the same direction as the congestion. The destination either asks the source to adjust the load or returns the signal to the source in the packets (or acknowledgements) moving in the reverse direction.

☞ Check Your Progress 2

1) What are the different approaches to open loop control?

……

……

……

2) What is the difference between leaky bucket traffic shaper and token bucket traffic shaper?

……

……

……

3.8 SUMMARY

In this unit, we examined several aspects of congestion i.e., what is Congestion? How does it occur? We also differentiated between congestion control and flow control. Then, we gave two broad classification of congestion control; open loop and closed loop. At the end, we touched upon issues related to congestion control in packet switched network.

3.9 SOLUTIONS/ANSWERS

Check Your Progress 1

1)

Congestion Control vs. Flow Control	
Congestion Control is needed when buffers in packet switches overflow or congest.	Flow Control is needed when the buffers at the receiver are not depleted as fast as the data arrives.
Congestion is end to end, it includes all hosts, links and routers. It is a global issue.	Flow in between one data sender and one receiver. It can be done on link-to-link or end-to-end basis. It is a local issue.

2) The purpose of open loop solution is to ensure that the traffic generated by the source will not lower the performance of the network below the specified Q_o S. The network will accept traffic till Q_oS parameters are satisfied, otherwise, it rejects the packets.

In contrast, closed loop solutions are based on the concept of a feedback loop. These algorithms are called closed loop because the state of the network has to be fed, up to the source that regulates traffic. Closed loop algorithms follow dynamic approach to solution of congestion problems. It reacts during the congestion occurrence period or when the congestion is about to happen.

Check Your Progress 2

1) The following are the different approaches:

- Admission Control Mechanism
- Traffic Policing
- Traffic Shaping

2) (i) Token bucket algorithm is more flexible than leaky bucket traffic shaper algorithm but both are used to regulate traffic.

(ii) The leaky bucket algorithm does not allow idle hosts to save up permission to send large bursty packets later, whereas, token bucket algorithm allows saving up to maximum size of the bucket N. This means that burst up to N packet can be sent at once, allowing some burstiness in the output stream and giving fast response to sudden bursts of output [*Ref 1*].

(iii) Token bucket algorithm throws away tokens (i.e., transmission capacity), when the bucket fills up but never throws packets. In contrast, the leaky bucket algorithm discards the packets when the bucket fills up.

3.10 FURTHER READINGS

1) *Computer Networks,* 4th Edition, A.S. Tanenbaum, Prentice Hall of India, New Delhi.

2) *Communication Networks fundamental concepts and key architecture,* Leon Garcia and Indra Widjaja, Tata McGraw Hill, New Delhi.

3) *Data and Computer Communication,* 6th edition, William Stallings, Pearson Edition, New Delhi.